An Improved Hidden Vector State Model Approach and Its Adaptation in Extracting Protein Interaction Information from Biomedical Literature

Deyu Zhou, Yulan He and Chee Keong Kwoh
School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798

Abstract

Large quantity of knowledge, which is important for biological researchers to unveil the mechanism of life, often hides in the literature, such as journal articles, reports, books and so on. Many approaches focusing on extracting information from unstructured text, such as pattern matching, shallow and full parsing, have been proposed especially for biomedical applications. In this paper, we present an information extraction system employing a semantic parser using the Hidden Vector State (HVS) model for protein-protein interactions. We found that it performed better than other established statistical methods and achieved 58.3% and 76.8% in recall and precision respectively. Moreover, the pure data-driven HVS model can be easily adapted to other domains, which is rarely mentioned and possessed by other approaches. Experimental results prove that the model trained on one domain can still generate satisfactory results when shifting to another domain with a small amount of adaptation training data.

1 Introduction

It is essential for biology researchers to understand functions of proteins and how they interact with each other which unveil the mechanism of living cells and provide targets for effective drug designs. Many databases, such as BIND [1], IntAct [2] and STRING [3], have been built to store protein-protein interaction information. However, 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 the full-text journals. 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 this way, 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 generate good results because they ignore sentence structures which play an important role in expressing protein-protein interactions.

Later, more and more complicated approaches have been proposed. They can be roughly classified into two categories, based on simple pattern matching, or employing parsing techniques. Approaches using pattern matching [8, 9, 10] rely on a set of predefined patterns or templates to extract protein-protein interactions. For example, Ono [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 precision rate of 84% for *Saccharomyces cerevisiae (yeast)* and *Escherichia coli*. Blaschke [9] and his colleagues introduced the 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 processing to define patterns when shifting to another domain.

Parsing based methods employ either shallow or full 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 are estimated at 50-80% and 30-80%, respectively.

Systems based on full parsing [13, 14, 15] deal with the structure of an entire sentence and therefore are potentially more accurate. Yakushiji [13] defined a grammar for biomedical domain and used a general full parser to extract interaction events. Another full parsing based approach used 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]. The major drawback of the aforementioned

methods is that they may require complete redesign of the grammar in order to be tuned to different domains.

Existing approaches, either pattern matching based or parsing based, still require human efforts to define word patterns or grammars in order to extract protein interaction events. It is therefore extremely difficult to port them into another domain. Thomas [16] tried to transform an existing Information Extraction (IE) system, SRI's Highlight, to a new biomedical domain. The recall and precision rate were reported to about 30% and 70%.

In this paper, we present a protein-protein interaction extraction system based on the Hidden Vector State (HVS) model. Preliminary results have been reported in [17]. Here, we describe some modifications made on the system to improve its performance, such as amending the predefined extraction rules, adding preposition information to the semantic annotation and so on. Furthermore, we show that the HVS model can be easily adapted to another domain by employing some standard adaptation algorithms.

The rest of the paper is organized as follows. Section 2 briefly describes the HVS model and how it can be used to extract protein-protein interactions from un-structured texts. Section 3 presents the overall structure of the extraction system. Improved experimental results are described in section 4. Adaptation methods and results are discussed in section 5. Finally, section 6 concludes the paper.

2 The Hidden Vector State Model

The Hidden Vector State (HVS) model [18] 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.

Each vector state in the HVS model is in fact equivalent to a snapshot of the stack in a push-down automaton and state transitions may be factored into a stack shift by n positions followed by a push of one or more new preterminal semantic concepts relating to the next input word. Such stack operations are constrained in order to reduce the state space to a manageable size. Natural constraints to introduce are limiting the maximum stack depth and only allowing one new preterminal semantic concept to be pushed onto the stack for each new input word. Such constraints effectively limit the class of supported languages to be right branching. The joint probability $P(N, \mathbf{C}, W)$ of a series of stack shift operations N, concept vector sequence $\mathbf{C}$, and word sequence W can be decomposed as follows

$$P(N, \mathbf{C}, W) = \prod_{t=1}^{T} P(n_t|W_1^{t-1}, \mathbf{C}_1^{t-1}) \cdot P(c_t[1]|W_1^{t-1}, \mathbf{C}_1^{t-1}, n_t) \cdot P(w_t|W_1^{t-1}, \mathbf{C}_1^{t}) \quad (1)$$

where:

- $\mathbf{C}_1^t$ denotes a sequence of vector states $\mathbf{c}_1..\mathbf{c}_t$. $\mathbf{c}_t$ at word position t is a vector of D_t semantic concept labels (tags), i.e. $\mathbf{c}_t = [c_t[1], c_t[2], .., c_t[D_t]]$ where $c_t[1]$ is the preterminal concept and $c_t[D_t]$ is the root concept (SS in Figure 1);
- $W_1^{t-1}\mathbf{C}_1^{t-1}$ denotes the previous word-parse up to position $t-1$;
- n_t is the vector stack shift operation and takes values in the range of $0, .., D_{t-1}$ where D_{t-1} is the stack size at word position $t-1$;
- $c_t[1] = c_{w_t}$ is the new preterminal semantic tag assigned to word w_t at word position t.

The result is a model which is complex enough to capture hierarchical structure but which can be trained automatically from only lightly annotated data.

In the HVS model used by our information extraction system, Equation 1 is approximated by

$$\begin{aligned} P(n_t|W_1^{t-1}, \mathbf{C}_1^{t-1}) &\approx P(n_t|\mathbf{c}_{t-1}) \\ P(c_t[1]|W_1^{t-1}, \mathbf{C}_1^{t-1}, n_t) &\approx P(c_t[1]|c_t[2..D_t]) \\ P(w_t|W_1^{t-1}, \mathbf{C}_1^{t}) &\approx P(w_t|\mathbf{c}_t) \end{aligned}$$

3 System Overview

The overall architecture of the extraction system is shown in Figure 2. Generally, the extracting process can be divided into three steps. At the beginning, abstracts are retrieved from MEDLINE and split into sentences. Protein names are identified based on a manually constructed dictionary of biological term. 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 shown in Figure 3(b). After that, each sentence is parsed by the HVS semantic parser. Before doing so, the HVS model needs to be trained using a lightly annotated training corpus. Figure 3(c) shows the two best parsing results. Finally, information about protein-protein interactions is extracted from the tagged sentences using a set of predefined simple rules. The extraction result is shown in Figure 3(d).

The details of each step were described in [17]. Here, we have made some modifications on the system as listed below:

1. Instead of using the best parsing result, that is with the highest probability, from the HVS model, the top 5 parsing results are considered. Based on the 5-best parsing results for each sentence, the extraction rules have been amended as follows:

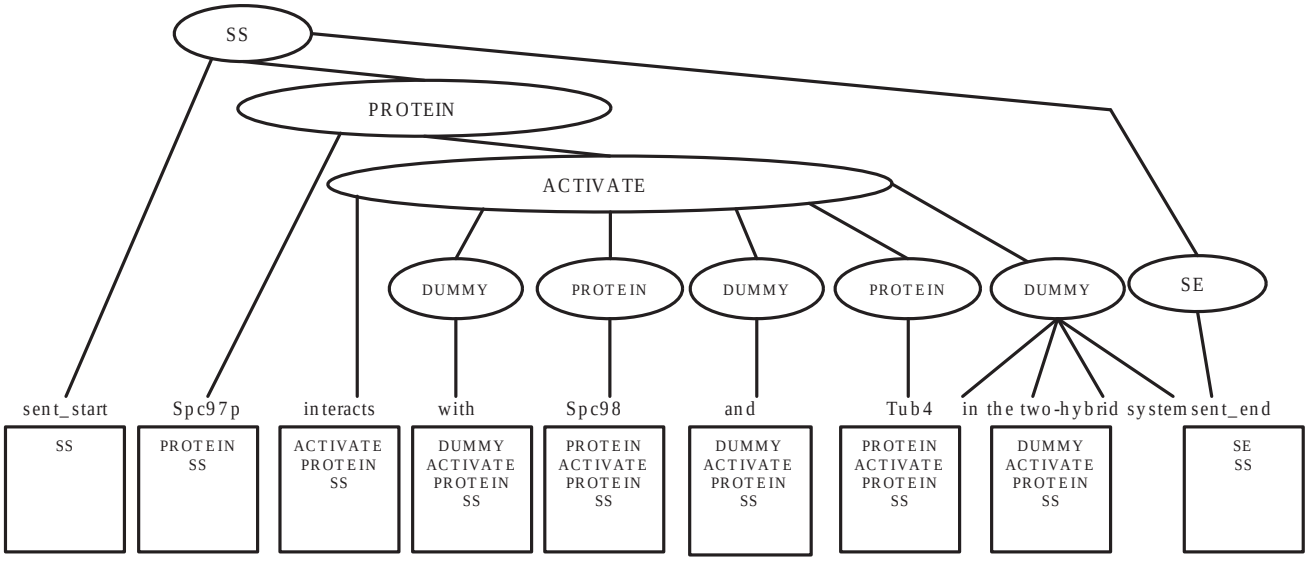

Figure 1: Example of a parse tree and its vector state equivalent.

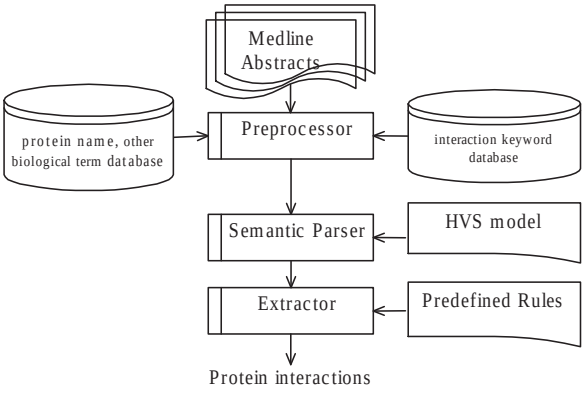

Figure 2: System architecture.

- If a sentence contains a keyword describing the protein interaction relationship, such as "activate", "attach" etc, and it is tagged with "DUMMY" in the best parsing result, 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 usage of the rule. 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+PROTEIN_NAME or SS+REL+PROTEIN_NAME+PROTEIN_NAME can be found in the parsing result, 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.

2. To train the HVS model, an abstract annotation needs to be provided for each sentence. For example, for the 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.
   The Annotation is:
   PROTEIN_NAME(ACTIVATE(PROTEIN_NAME)).
   We suspected that prepositions play an important role in expressing hierarchical relationships, therefore we provided another set of annotations which include the preposition information as shown below:
   PROTEIN_NAME(ACTIVATE(WITH(PROTEIN_NAME)))

4 Experiments

Some modifications have been discussed in section 3. In this section, experimental results are presented which show that these enhancements indeed greatly improve the system performance.

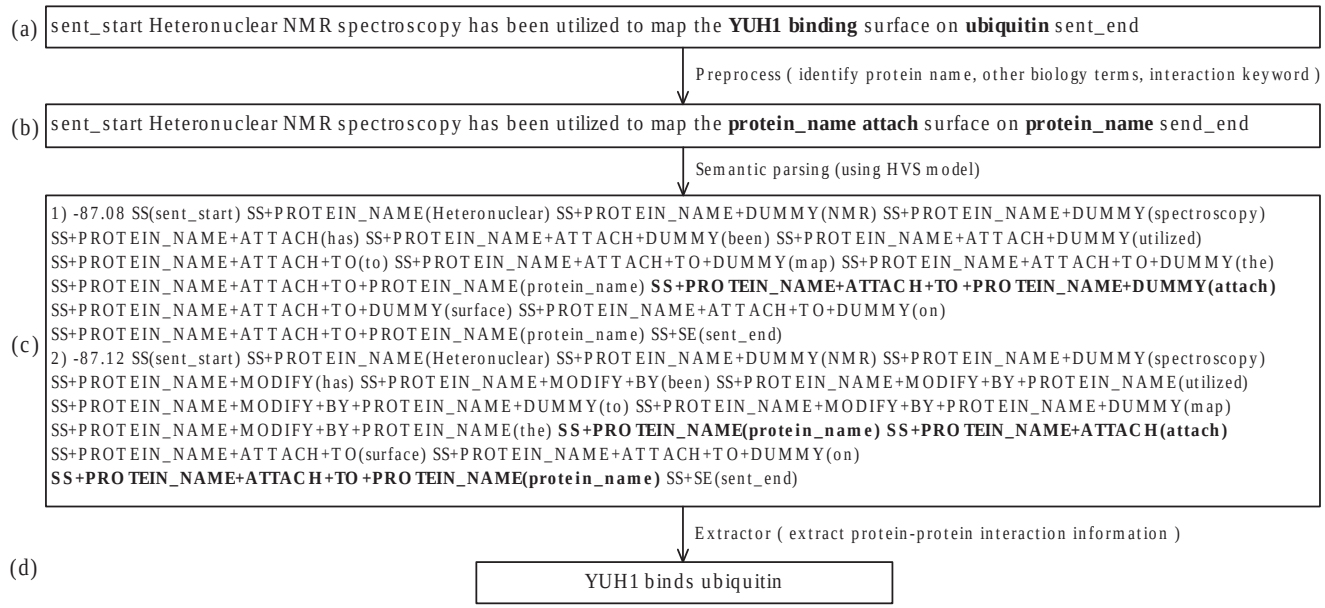

Figure 3: An example of a procedure for information extraction based on the HVS model.

4.1 Setup

The experimental data were obtained from [10]. The initial corpus consists of 1203 sentences. The protein interaction information for each sentence is also provided. All sentences were examined manually to ensure the correctness of the protein-protein interactions. After cleaning up the sentences which do not contain protein interaction information, 800 sentences were kept. We name it as Corpus I.

The corpus I data were split randomly into the training set and the test set at the ration of 9:1. The test set consists of 80 sentences and the remaining 720 sentences were used as the training set.

4.2 Results

Experiments were conducted three times (i.e Experiment 1, 2, 3 in Table 1) with different training and test data each round. 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 here are based on the values of TP, TN, and NP. TP is the number of correctly extracted interactions. (TP+TN) is the number of all interactions in the test set and (TP+NP) is the number of all extracted interactions. F-score is computed using the formula below:

$$\text{F-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \qquad (2)$$

where Recall is defined as $TP/(TP + TN)$ and Precision is defined as $TP/(TP + NP)$.

4.2.1 Including prepositions in the annotation of the training set

As mentioned in section 3, two types of annotations were provided for the training set. Table 1 lists the results generated by the HVS model trained without or with the preposition information. It can be seen that by including the preposition information, the relative improvement on F-measure is 5-17%. This gives positive support on our hypothesis that preposition information do play an important role on revealing the underlying semantic information of the sentence.

Experiment	Recall (%)	Precision (%)	F-Score (%)
No preposition information			
1	56.7	71.6	63.3
2	43.1	83.3	56.8
3	50.3	73.6	59.8
overall	50.1	75.2	60.2
Including preposition information			
1	61.7	71.8	66.4
2	52.6	91.0	66.7
3	60.2	72.7	65.8
overall	58.3	76.8	66.3

Table 1: Results with or without the preposition information.

4.2.2 Results based on the sentence complexity and the interaction category

To analyze the ability of the HVS model in extracting information from syntactically complex sentences, we measured the performance on the sentences containing only one protein-protein interaction and the sentences containing more than one interaction separately. The rationale behind this is that in general, sentences containing more than one protein-protein interaction would exhibit more complex syntactic structures. It can be observed from Table 2 that F-measure only dropped slightly by 3% for the Experiment 1 and Experiment 3 data when tested on the more complex sentences. It is also noted that the degradation of F-score is quite dramatic for Experiment 2. One possible reason is that this data set contains quite a lot extremely complex sentences. One example is:
The polo-box-dependent interactions between Cdc5 and septins (Cdc11 and Cdc12) and genetic interactions between the dominant-negative cdc5DeltaN and Cyk2/Hof1 or Myo1 suggest that direct interactions between cdc5DeltaN and septins resulted in inhibition of Cyk2/Hof1- and Myo1-mediated cytokinetic pathways.

Experiment	Recall (%)	Precision (%)	F-Score (%)
With one or no protein-protein interaction			
1	68.1	68.1	68.1
2	64.2	87.2	73.9
3	73.9	63.0	68.0
With more than one protein-protein interaction			
1	57.5	75.0	65.1
2	42.9	96.4	59.3
3	54.0	81.0	64.8

Table 2: Results based on the sentence complexity.

By analyzing the categories of protein-protein interactions in our data set, we found that two categories, *activate* and *attach* accounts for about 50% of all protein-protein interactions. Thus, the results based on these two categories are also shown here. It can be observed from Table 3 that there are slight changes in F-score when compared with the overall performance result in Table 1. It increases by about 1% for the *activate* category and drops 2% for the *attach* category.

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

Table 3: Results based on the interaction category.

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 reduce the gap between training and test or 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 of models are the two MAP estimation methods investigated in speech recognition experiments [19]. In recent years, MAP adaptation has been successfully applied to lexicalized probabilistic context-free grammar (PCFG) models [20]. 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 [21].

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 [1].

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 [22].

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

$$\hat{P}_j(k) = \frac{1}{Z_\lambda} P_j(k)^{\lambda_1} \tilde{P}_j(k)^{\lambda_2} \tag{3}$$

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

$$\tilde{P}_j(k) = \frac{\sigma_j(k)}{\sum_{i=1}^{K} \sigma_j(i)} \tag{4}$$

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

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

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 [23].

5.2 Experimental Results

To justify the robustness of the HVS parser, another corpus named as corpus II was used. It comprises 300 abstracts which are randomly retrieved from the GENIA corpus [24]. GENIA is a collection of research abstracts selected from the search results of MEDLINE database with keyword (MeSH terms) *human, blood cells and transcription factors*. These abstracts were then split into sentences and those containing more than two protein names were kept. Altogether 1279 sentences were left.

Note that Corpus I obtained from [10] is constructed from the first 50 biomedical papers downloaded from the Internet with the keyword "*protein-protein interaction*". Thus Corpus I and Corpus II are disjoint sets and they might comprise different writing styles.

The baseline HVS model was trained on Corpus I and was later adapted using a small amount of adaptation data from Corpus II. Table 4 listed the recall, precision, and F-score obtained when tested on the 800 sentences from Corpus II. The "Baseline" results were obtained using the HVS model trained on Corpus I and tested on corpus II 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 160 randomly selected adaptation sentences from Corpus II.

System	Recall (%)	Precision (%)	F-score (%)
Baseline	50.7	52.7	51.7
In domain	65.7	68.8	67.2
Log-Linear	60.4	66.3	63.2

Table 4: Performance Comparison of adaptation to Corpus II.

Overall, we found that moving a system trained on Corpus I to Corpus II resulted in a 15% 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 51.7% was restored to 63.2%, which is not far from the in-domain F-score of 67.2% .

Figure 4 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 110. The performance however degrades when the number of adaptation utterances exceeds 160, possibly due to model overtraining. For this particular application, we conclude that just 110 adaptation utterances would be sufficient to adapt the baseline model to give comparable results to the in-domain model.

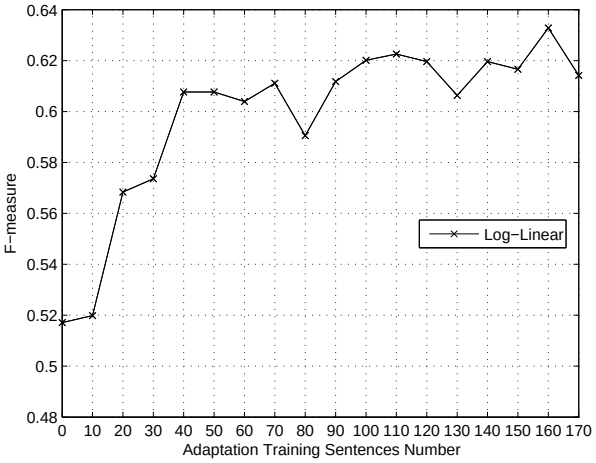

Figure 4: F-measure vs amount of adaptation training data.

6 Conclusions and Future work

In this paper, we have presented a improved HVS model-based system to automatically extract protein-protein interactions from unstructured text sources. The system 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 results may provide a useful supplement to manually created resources in established public databases.

In future work we will work on the modification of the HVS model, such as enlarging its ability of expressing more complex sentence structures, and dealing with negative sentences which constitutes a well-known problem etc.

References

[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 interac-

tion 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] Nikolai Daraselia, Anton Yuryev, Sergei Egorov, Svetalana Novichkova, Alexander Nikitin, and llya Mazo. Extracting human protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics*, 20(5):604–611, 2004.

[16] J. Thomas, D. Milward, C. Ouzounis, and S. Pulman. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 541–552, Hawaii, U.S.A, 2000.

[17] 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*, Reading, UK, 2006.

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

[19] 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.

[20] 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 (HLT-NAACL 2003)*, Edmonton, Canada, May 2003.

[21] B. Roark, M. Saraclar, and M. Collins. Corrective language modeling for large vocabulary asr with the perceptron algorithm. In *Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 749–752, 2004.

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

[23] 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.

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