Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Semi-supervised Learning of the Hidden Vector State Model for Protein-Protein Interactions Extraction

Deyu Zhou, Yulan He and Chee Keong Kwoh
School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798
{zhou0063, asylhe, asckkwoh}@ntu.edu.sg

***Abstract*— A major challenge in text mining for biology and biomedicine is automatically extracting protein-protein interactions from the vast amount of biological literature since most knowledge about them still hides in biological publications. Existing approaches can be broadly categorized as rule-based or statistical-based. Rule-based approaches require heavy manual efforts. On the other hand, statistical-based approaches require large-scale, richly annotated corpora in order to reliably estimate model parameters. This is normally difficult to obtain in practical applications. The hidden vector state (HVS) model, an extension of the basic discrete Markov model, has been successfully applied to extract protein-protein interactions. In this paper, we propose a novel approach to train the HVS model on both annotated and un-annotated corpus. Sentences selection algorithm is designed to utilize the semantic parsing results of the un-annotated corpus generated by the HVS model. Experimental results show that the performance of the initial HVS model trained on a small amount of the annotated data can be improved by employing this approach.**

***Keywords*—semi-supervised learning, information extraction, hidden vector state model, protein-protein interactions extraction.**

I. Introduction

Protein-protein interactions referring to the associations of protein molecules are crucial for many biological function. Understanding protein functions and how they interact with each other give biologists a deeper insight into the understanding of living cell, disease process and provide targets for effective drug designs. Although many database such as BIND [1], IntAct [2] and STRING [3], have been built to store protein-protein interactions, most knowledge about protein-protein interactions are so far still locked in the full-text journals. As a result, automatically extracting protein-protein interactions is crucial to meet the demand of the researchers.

Most existing approaches are either based on simple pattern matching, or by employing parsing methods. Approaches using pattern matching [4], [5] rely on a set of predefined patterns or rules to extract protein-protein interactions. Parsing based methods employ either shallow or deep parsing. Shallow parsers [6], [7] break sentences into none overlapping phases and extract local dependencies among phases without reconstructing the structure of an entire sentence. Systems based on deep parsing [8], [9] deal with the structure of an entire sentence and therefore are potentially more accurate. The major drawback of the aforementioned methods is that they may require complete manual redesign of grammars or rules in order to be tuned to different domains. On the contrary, statistical models can perform protein-protein interactions extraction without human intervention once they have been trained from annotated corpora. Many empiricist methods [10], [11] have been proposed to automatically generate the language model to mimic the features of un-structured sentences. For example, Seymore [12] used Hidden Markov Model (HMM) for the task of extracting important fields from the headers of computer science research papers. In [13], a statistical method based on the hidden vector state (HVS) model to extract protein-protein interactions has been proposed. However, methods of this categories do not perform well partially due to the lack of large-scale, richly annotated corpora.

How to learn from both annotated and un-annotated data, i.e. semi-supervised learning, have attracted much attention in recent years. The proposed methods include EM (expectation-maximization) with generative mixture models [14], self-training [15], co-training [16], transductive support vector machines [17], graph-based methods [18] and so on. Nigam *et al.* [14] applied the EM algorithm on the mixtures of polynomials for the task of text classification. They showed that the classifiers trained from both the labeled and unlabeled data perform better than those trained solely from the labeled data. Yarowsky [19] used self-training for word sense disambiguation. Rosenberg *et al.* [15] applied self-training to object detection from images. Jones [16] used co-training, co-EM and other related methods for information extraction from text. Blum *et al.* [18] proposed an algorithm based on finding minimum cuts in graphs to propagate labels from the labeled data to the unlabeled data. For a detailed survey on semi-supervised learning, please refer to [20].

In this paper, we present a novel method to train the HVS model on both un-annotated and annotated corpus. Utilizing the semantic parsing results of the un-annotated corpus generated by the HVS model trained on the annotated corpus, sentences with high confidence of being parsed correctly in un-annotated corpus are added into the annotated corpus for iteratively training the HVS model. The rest of the paper is organized as follows. Section II briefly describes the HVS model and how it can be applied to extract protein-protein

interactions. Section III presents the proposed approach of training the HVS model on both un-annotated and annotated corpus. Experimental results are discussed in section IV. Finally, section V concludes the paper.

II. The Hidden Vector State Model

The Hidden Vector State (HVS) model [21] 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 approximated as follows

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

where:

- $\mathbf{c}_t$ denotes the vector state at word position t, which consists of D_t semantic concept labels (tags), i.e. $\mathbf{c}_t = [c_t[1], c_t[2], .., c_t[D_t]]$ where $c_t[1]$ is the preterminal concept and $c_t[D_t]$ is the root concept (SS in Figure 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.

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)).
Such abstract annotations serve as constraints on limiting the forward-backward search during the model training to only include the states which are consistent with these constraints.

III. Methodologies

The HVS model uses a set of annotated sentences to learn class descriptions for protein-protein interactions. In practice, annotating the training sentences is a tedious, time consuming, error prone process. In order to reduce the effort of annotating sentences, a semi-supervised learning method is proposed, which is presented in this section.

As mentioned in section II, the HVS model does not require explicit semantic tag/word pairs to be given in the annotated corpus. All it needs are abstract semantic annotations for training. This means that many sentences might share the same semantic annotation and they therefore could possibly exhibit the similar syntactic structures which can be revealed through part-of-speech (POS) tagging. Figure 2 gives an example of several sentences sharing the same semantic annotation and their corresponding abbreviated POS tag sequences which were generated by removing unimportant POS tags from the original POS tag sequences. Here the symbol ACKEY denotes a protein-protein interaction keyword, PTN denotes a protein name, TO denotes the word "to", CC denotes a conjunction and IN denotes some prepositions such as "of", "between" etc.

A. Rationale

Suppose $E_L = \{\langle a_1, c(a_1)\rangle, \langle a_2, c(a_2)\rangle, \cdots, \langle a_n, c(a_n)\rangle\}$ is a set of labeled sentences with a_i being a sentence and $c(a_i)$ being its corresponding annotation and $E_U = \{\langle b_1, b_2, \cdots, b_m\rangle\}$ is a set of unlabeled sentences, we can use some classic clustering algorithm to group sentences in E_L and E_U into several clusters which is illustrated in Figure 3. The distance between two sentences is defined as the distance between their corresponding simplified POS tag sequences and its calculation will be described in detail in section III-B. The HVS model M is initially trained on the data set E_L. Since some sentences in E_U might be in the same cluster with those in E_L, their semantic structures are very likely to be identified correctly by M. Adding these sentences and their corresponding annotations which are automatically generated from semantic parsing results should improve the performance of the original model M. Based on this rationale, we can see that it is crucial to select the sentences from E_U to make ensure that their semantic parsing results are correct with high confidence. If adding examples with incorrect annotations, obviously the performance of M will be degraded.

The procedure of the proposed semi-supervised learning method is described in Figure 4, where it starts with a (small) set of annotated sentences E_L and a (large) set of un-annotated sentences E_U. A baseline HVS model M is constructed on the data set E_L. A hypothesis h (a semantic parsing result) is induced for each sentence in E_U using M. Then the un-annotated sentence $q \in E_U$ is selected to maximize the precision of its h, which is described in detail in section III-C. The annotation of q, $c(q)$ can then be deduced from its hypothesis h. After adding $\langle q, c(q)\rangle$ to E_L, the model M is refined with the enlarged annotated corpus and the whole procedure repeats until it converges.

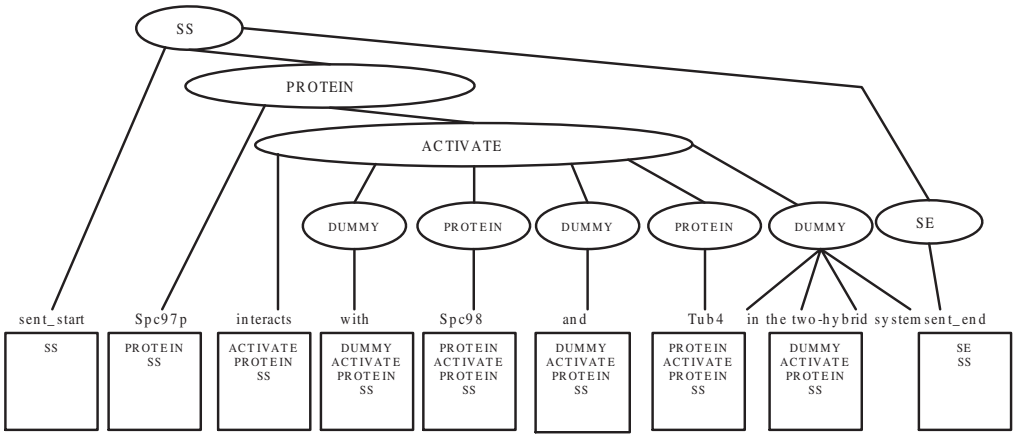

Fig. 1. Example of a parse tree and its vector state equivalent.

SS(KEY(PROTEIN_NAME(PROTEIN_NAME)) SE)	
Sentence	Abbreviated POS tag sequence
WW domain 3 (but not the other WW domains) was both necessary and sufficient for *the binding of hNedd4 to alphaENaC*.	ACKEY IN PTN TO PTN
The structural prediction was confirmed by site-directed mutagenesis of these electronegative residues , resulting in loss of *binding of Siah1 to SIP* in vitro and in cells.	ACKEY IN PTN TO PTN
The physical *interaction of cdc34 and ICP0* leads to its degradation.	ACKEY IN PTN CC PTN
Finally , an in vivo *interaction between pVHL and hnRNP A2* was demonstrated in both the nucleus and the cytoplasm.	ACKEY IN PTN CC NN PTN
The in vivo *interaction between DAP-1 and TNF-R1* was further confirmed in mammalian cells.	ACKEY IN PTN CC PTN

Fig. 2. An example of multiple sentences sharing the same annotation.

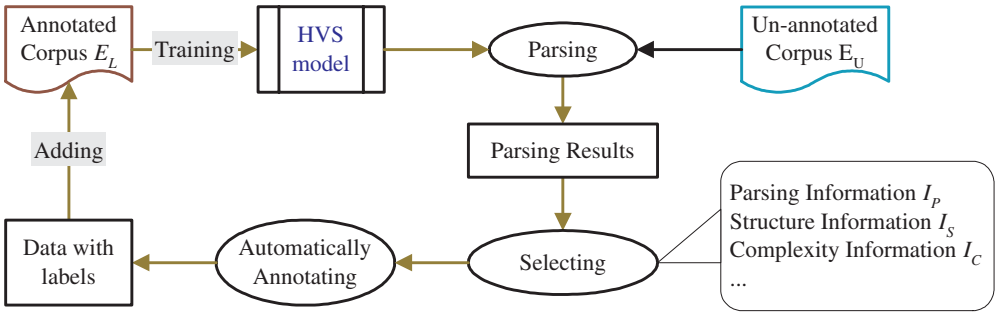

Fig. 4. Flowchart of training procedure employing semi-supervised learning.

B. Distance Calculation

The distance between two sentences is defined as the distance between their corresponding simplified POS tag sequence, which is calculated based on sequence alignment. Suppose $\mathbf{a} = a_1 a_2 \cdots a_n$ and $\mathbf{b} = b_1 b_2 \cdots b_m$ be the two POS tag sequences of length of n and m, define $S(i, j)$ as the score of the optimal alignment between the initial segment from a_1 to a_i of $\mathbf{a}$ and the initial segment from b_1 to b_j of $\mathbf{b}$,

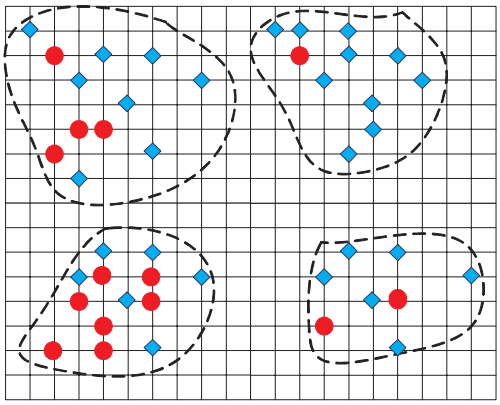

Fig. 3. Sketch map of clustering examples in E_L and E_U, where red circle denotes $a_i(i = 1 \cdots n)$ and blue diamond denotes $b_j(j = 1 \cdots m)$.

where $S(i, j)$ is recursively calculated as follows:

$$S(i,0) = 0, i = 1, 2, ...n \quad (2)$$

$$S(0,j) = 0, j = 1, 2, ...m \quad (3)$$

$$S(i,j) = \max \begin{cases} 0, \\ S(i-1, j-1) + s(a_i, b_j), \\ S(i-1, j) + s(a_i, '-'), \\ S(i, j-1) + s('-', b_j) \end{cases} \quad (4)$$

Here $s(a_i, b_j)$ is the score of aligning a_i with b_j and is defined as:

$$s(a_i, b_j) = \log \left[\frac{p(a_i, b_j)}{p(a_i) \times p(b_j)} \right] \quad (5)$$

where, $p(a_i)$ denotes the occurrence probability of tag a_i and $p(a_i, b_j)$ denotes the probability that a_i and b_j appear at the same position in two aligned sequences.

A score matrix can then be built and dynamic programming is used to find the largest score between the two sequences.

Based on the aforementioned method for distance measurement, sentences in E_L and E_U can be clustered into various groups. Such clustering results are then fed into the stage of sentence selection.

C. Sentences selection

To select the best semantic parsing results and their corresponding sentences from E_U, we need to define a variable DG_p to describe the degree of their fitness. First of all, we need to define some parameters which will be employed to express the variable DG_p.

Suppose sentence $S_i \in E_U$ has its correspondent parsing path P_i, parsing information I_P, structure information I_S, complexity information I_C are defined as follows:

- **Parsing Information** I_P, describing the information in the parsing result P_i, is defined as follows:

$$I_P = 1 - \frac{\sum_{j=1}^{N} KeyITD(S_{ij})}{\sum_{j=1}^{N} Key(S_{ij})}. \quad (6)$$

Here, N denotes the length of the sentence S_i, S_{ij} denotes the jth word of the sentence S_i and function $KeyITD$, Key are defined as:

$$Key(S_{ij}) = \begin{cases} 1, \text{if } S_{ij} \text{ is a protein name or a protein interaction keyword} \\ 0, \text{ otherwise} \end{cases} \quad (7)$$

$$KeyITD(S_{ij}) = \begin{cases} 1, \text{if } Key(S_{ij}) \text{ is 1 and the semantic tag of } S_{ij} \text{ is DUMMY} \\ 0, \text{otherwise} \end{cases} \quad (8)$$

- **Structure Information** I_S, describes the similarity between the structure information of the sentence S_i and those of all the sentences in E_L, which is defined as follows:

$$I_S = 1 - \frac{min(Dist(S_i, S_j)|S_j \in E_L)}{max(Dist(S_k, S_j)|S_k \in E_U, S_j \in E_L)} + \frac{Num(C(S_i))}{\|E_L\|}, \quad (9)$$

where $C(S_i)$ denotes the cluster where S_i locates and $Num(C(S_i))$ denotes the number of sentences of E_L in the cluster $C(S_i)$.

- **Complexity Information** I_C, describing the complexity of the sentence S_i, is defined as follows:

$$I_C = 1 - \frac{Length(S_i)}{Max(Length(S_j)|S_j \in E_U \bigcup E_L)} \quad (10)$$

Overall, it can be observed that the higher the value of I_P, I_S, and I_C, the higher confidence of the correctness of the semantic parsing path P_i.

After defining the above parameters, DG_p is defined as

$$DG_p = \beta_p I_P + \beta_s I_S + \beta_c I_C + \beta_0, \quad (11)$$

which is a combination of the above defined three parameters. To estimate the coefficients $\boldsymbol{\beta} = (\beta_p, \beta_s, \beta_c, \beta_0)$, the method of least squares is applied and the coefficients $\boldsymbol{\beta}$ are selected to minimize the residual sum of squares,

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{N} (DG_{pi} - \beta_p I_{Pi} - \beta_s I_{Si} - \beta_c I_{Ci} - \beta_0)^2 \quad (12)$$

where N is the number of training data.

The sentence selection algorithm is described in Figure 5. An example is given in Figure 6 to illustrate how to automatically generate annotations from the semantic parsing results.

IV. Experiments

A. Setup

To evaluate the efficiency of the proposed method, a corpus is constructed, which consists of sentences retrieved from the GENIA corpus [22]. GENIA is a collection of research abstracts selected from the search results of Medline database with keywords (MESH terms) *human, blood cells and transcription factors* . These abstracts were then split into

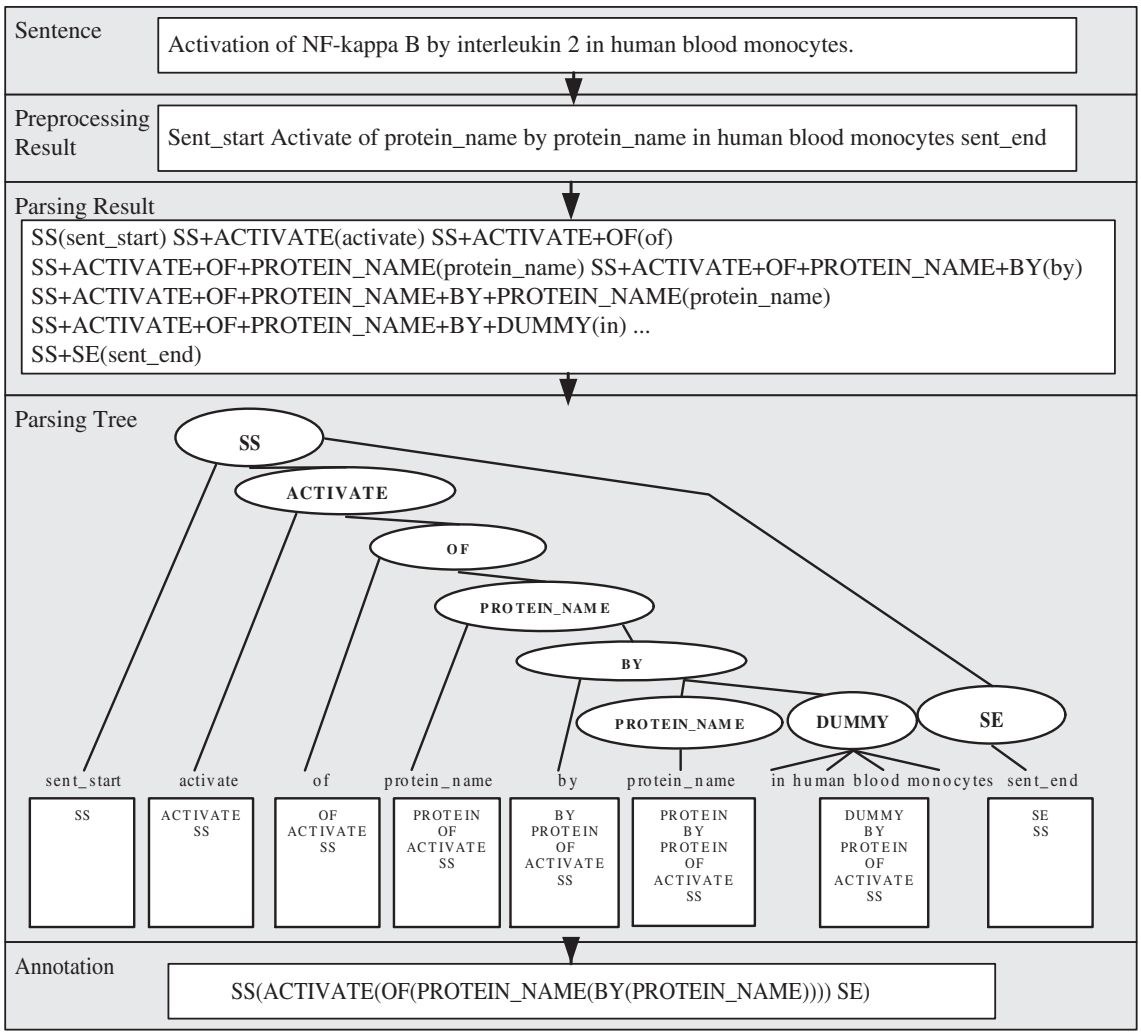

Fig. 6. An example illustrated the process of automatically generating annotations from semantic parsing results.

sentences and those containing more than two protein names were kept. Altogether 2500 sentences were left. The corpus was split into two parts; part I contains 1500 sentences which can be further split into two data sets: E_L consisting of 500 sentences and E_U consisting of the remaining 1000 sentences, part II consists of 1000 sentences which was used as the test data set.

We have preformed 10-fold cross-validation on the model performance. 250 sentences were randomly sampled from Part II in the corpus , which is done ten times. To ensure the justness of sampling, sentences in the test data set are grouped into four subsets based on their complexity I_C. Sentences were then drawn fairly from each of subsets so that the coverage over the whole test set (1000 sentences) in term of sentence complexity was ensured for each of the sampled test data (250 sentences). Figure 7 illustrates the distribution of sentence length in the whole test data set.

The results reported here are based on the values of TP (true positive), FN (false negative), and FP (false positive). 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:

$$\text{F-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{13}$$

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

B. Results

The baseline HVS model was trained on the data set E_L which consists of 500 sentences. Sentences from the data set E_U were then selected and automatically assigned with semantic annotations based on the method described in section III. The HVS model were incrementally trained with

Input: The sets E_L and E_U of labeled and unlabeled sentences
$\hat{DG}_p$, the threshold of DG_p
Clusters constructed on E_L and E_U
Procedure:
1: Train the HVS model from E_L
2: Generate parsing result for each sentence $S_i \in E_U$
3: Set Q to null
4: Loop for each sentence S_i
Calculate the DG_p for S_i
If ($DG_p > \hat{DG}_p$)
Generate annotation $c(S_i)$ for S_i
Add $\langle S_i, c(S_i)\rangle$ into Q
EndIf
EndLoop
5: If Q equals to null, procedure stops
Else remove sentences in Q from E_U, add Q to E_L and goto 1
EndIf

Fig. 5. Procedure of sentence selection.

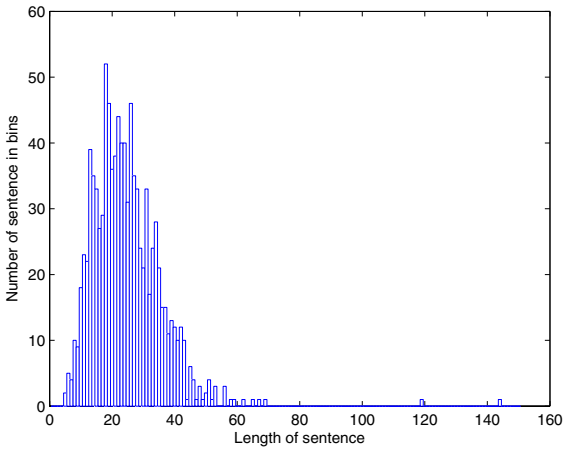

Fig. 7. Histogram of sentence length in test set.

those newly added training data. The process is repeated until no more sentences can be selected. Total 200 sentences from the un-annotated data set E_U were selected and assigned with the semantic annotations after ten iterations. Table I lists the evaluation results. The "baseline" results were obtained using the initial HVS model trained on E_L (500 sentences). The "improved" results were obtained using the final HVS model trained on the combined data which include the initial 500 sentences and the later added 200 sentences. The "Best" row shows the performance of the HVS model trained on the whole data Part I.

Overall, we found that by adding the sentences selected from unlabeled data, the relative improvement on F-measure is around 2.4%. It gives positive support on the efficiency of our method.

Experiment	Recall (%)	Precision (%)	F-Score (%)
Baseline	55.8	55.6	55.7
Improved	57.5	58.7	58.1
Best	64.2	59.5	61.7

TABLE I
TEXT MINING RESULTS USING THE HVS MODEL.

Figure 8 shows the protein-protein interactions extraction performance versus the training iterations where the number of sentences added in each iteration is listed in Table II. The best performance was obtained in the second iteration where F-score reaches 58.4%.

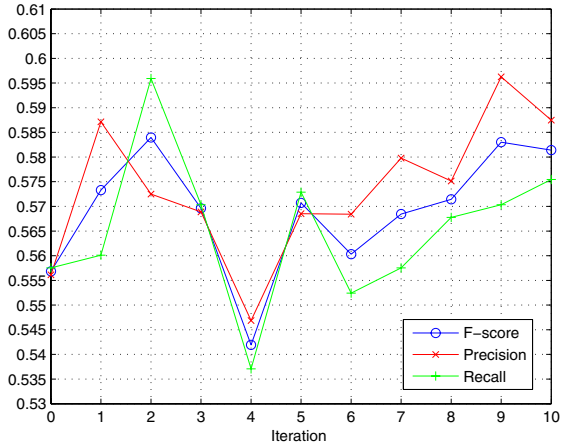

Fig. 8. F-measure vs training iterations.

Iteraction	Number of sentences
1	19
2	21
3	31
4	30
5	20
6	19
7	18
8	13
9	10
10	19

TABLE II
NUMBER OF SENTENCES ADDED IN EACH ITERATION.

It can be observed that F-score value in general as the semi-supervised learning procedure repeats. The performance however degrades in iteration 3 and 4, possibly due to un-correctness of annotations of added sentences.

V. Conclusions and Future work

In this paper, we presented a novel semi-supervised learning method to train the HVS model on both un-annotated and annotated data. Based on sentence selection algorithm, semantic annotations can be automatically generated from the semantic parsing results for the un-annotated sentences. The HVS model can then be refined with the increasingly added un-annotated data and this eventually leads to the increase on the F-measure when used for protein-protein interactions extraction. In future work, we will investigate the combination of semi-supervised learning with active learning to further improve the performance of the HVS model.

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 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] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] S. Novichkova, S. Egorov, and N. Daraselia. MedScan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, 19(13):1699–1706, 2003.

[11] 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.

[12] Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. Learning Hidden Markov Model Structure for Information Extraction. In *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

[13] 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.

[14] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[15] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised selftraining of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2005.

[16] Rosie Jones. *Learning to extract entities from labeled and unlabeled text*. PhD thesis, Carnegie Mellon University, 2005.

[17] Linli Xu and Dale Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI-05, The Twentieth National Conference on Artificial Intelligence*, 2005.

[18] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of 18th International Conference on Machine Learning*, pages 19–26. Morgan Kaufmann, San Francisco, CA, 2001.

[19] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[20] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

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

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