Training the Hidden Vector State Model from Un-annotated Corpus

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. Since most knowledge about protein-protein interactions still hides in biological publications, there is an increasing focus on automatically extracting information from the vast amount of biological literature. Existing approaches can be broadly categorized as rule-based or statistically-based. Rule-based approaches require heavy manual effort. On the other hand, statistically-based approaches require large-scale, richly annotated corpora in order to reliably estimate model parameters. This is normally difficult to obtain in practical applications. We have proposed a hidden vector state (HVS) model for protein-protein interactions extraction. The HVS model is an extension of the basic discrete Markov model in which context is encoded as a stack-oriented state vector. State transitions are factored into a stack shift operation similar to those of a push-down automaton followed by the push of a new preterminal category label. In this paper, we propose a novel approach based on the k-nearest-neighbors classifier to automatically train the HVS model from un-annotated data. Experimental results show the improved performance over the baseline system with the HVS model trained from a small amount of the annotated data.

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

1 Introduction

Understanding protein functions and how they interact with each other give biologists a deeper insight into the understanding of living cell as a complex machine, disease process and provide targets for effective drug designs. As of to date, vast knowledge of protein-protein interactions are still locked in the full-text journals. As a result, automatically extracting information about 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 [1] rely on a set of predefined patterns or rules to extract protein-protein interactions. Parsing based methods employ either deep or shallow parsing. Shallow parsers [2] 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 [3] 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 the protein-protein interactions extraction task without human intervention once they are trained from annotated corpora. Many empiricist methods [4] have been proposed to automatically generate the language model to mimic the features of un-structured sentences. For example, Seymore [5] used Hidden Markov Model (HMM) for task of extracting important fields from the headers of computer science research papers. In [6], a statistical method based on the hidden vector state model (HVS) to automatically extract protein-protein interactions from biomedical literature 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 been investigated. The proposed methods include EM (expectation-maximization) with generative mixture models [7], self-training [8], co-training [9], transductive support vector machines, graph-based methods [10] and so on. Nigam *et al.* [7] 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 [11] used self-training for word sense disambiguation. Rosenberg *et al.* [8] applied self-training to object detection from images. Jones [9] used co-training, co-EM and other related methods for information extraction from text. Blum *et al.* [10] 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 [12].

In this paper, we present a novel method to automatically train the HVS model from un-annotated data. Considering a semantic annotation as a class label for each sentence, we employ Part-Of-Speech (POS) tagging to convert the original sentence into the POS tag sequence, we then use the modified k-nearest-neighbors (KNN) classifiers to assign a semantic annotation to an unseen sentence based on its POS tag sequence. The rest of the paper is organized as follows. Section 2 briefly describes the HVS model and how it can be applied to extract protein-protein interactions from the biomedical literature. Section 3 presents the proposed approach on automatically training the HVS model from un-annotated data. Experimental results are discussed in section 4. Finally, section 5 concludes the paper.

2 The Hidden Vector State Model

The Hidden Vector State (HVS) model [13] 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. 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;
- $W_1^{t-1}\mathbf{C}_1^{t-1}$ 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 details of how this is done are given in [13]. 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)).

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 1 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 KEY denotes the protein-protein interaction keyword and PTN denotes the protein name.

SS(KEY(PROTEIN_NAME(PROTEIN_NAME)) SE)	
Sentence	Abbreviated POS tag sequence
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. 1. An example of multiple sentences sharing the same annotation.

3 Methodologies

In this section, the procedure of training the HVS model from un-annotated corpus is described which employs the k-nearest-neighbors algorithm with POS sequences alignment.

Considering the semantic annotation as the class label for each sentence, semantic annotation can be converted to a traditional classification problem. Given a small set of annotated data and a large set of un-annotated data, we would like to predict the annotations for the sentences from the un-annotated data based on their similarities to the sentences in the annotated corpus. At the beginning, full papers are retrieved from MedLine and split into sentences. Protein names and keywords describing protein-protein interaction are then identified based on a preconstructed dictionary. After that, each sentence is parsed by a POS tagger and the POS tag sequence is generated. Finally, based on the POS tag sequence, the sentence will be assigned an annotation based on the similarity measure to the existing sentences in the annotated corpus.

The details of each step are described below.

1. Identifying protein names and protein interaction keywords.
   Protein names need to be identified first in order to extract protein-protein interaction information. In our system, protein names are identified based on a manually constructed dictionary. Since protein interaction keywords play an important role in later step, a keyword dictionary describing interaction categories has been built based on [3].
2. Part-of-speech tagging.
   The part-of-speech (POS) tags for each sentence is generated by the Brill's tagger [14]. The Brill's tagger can only achieve 83% overall accuracy on biomedical text since there are many unknown biomedical domain-specific words. We plan to replace it with the POS tagger trained from the biomedical domain in our future work.
3. Automatically generating semantic annotations for the un-annotated sentences.
   The semantic annotations for the un-annotated sentences are assigned by the KNN classifier.

3.1 *k*-Nearest-Neighbor Classification with Constrains

Since automatically generating annotations for un-annotated sentences can be converted into a classification problem, we applied the k-nearest-neighbor (KNN) algorithm to handle this task. The training data consists of N pairs (x_1, y_1), (x_2, y_2), $\ldots, (x_N, y_N)$, with x_i denotes a POS tag sequence, and y_i denotes a semantic annotation. Given a query point x_q, the KNN algorithm finds the k training points $x_{(r)}$, $r = 1, ..., k$ closet in distance to x_q, and then classify using majority voting among the k neighbors.

In our implementation here, the distance between two POS tag sequences are derived based on dynamic programming for sequence alignment instead of the commonly used Euclidean distance. In section 3.2, we discuss in detail on the distance measure which is more appropriate for our purpose. Also, instead of majority voting, some rules are defined to classify a sentence among its k neighbors as shown in Figure 2. The reason behind is that only a small amount of training data are available here and majority voting would require a large amount of training data in order to get reliable results.

Input: a POS tag sequence S, its k neighbors set $K_N = \{N_1, N_2, \cdots, N_k\}$ and their corresponding class label $C_i = Class(N_i), i = 1, ..., k$.
Output: S's class label C_s
Algorithm:

1. Sort the k neighbors in descending order based on their respective similarity score with S and denote the sorted list as $N_1^*, N_2^*, ..., N_k^*$
2. Count the number of protein names c_{pt} and protein interaction keywords c_{kw} of S
3. For i = 1 to k do
   a) Count the number of protein names c_{pt}^i and protein interaction keywords c_{kw}^i of N_i^*
   b) If c_{pt}^i equals to c_{pt} and c_{kw}^i equals to c_{kw}, assign C_s with Class label of N_i^* and exit the loop
4. If C_s is not assigned any value, assign C_s with NULL
5. Return C_s

Fig. 2. Procedure of classification using KNN.

3.2 POS Sequence Alignment

The similarity between two POS sequences 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 $\mathbf{b}$,

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

$$S(i,0) = 0, i = 1,2,...n \tag{2}$$

$$S(0,j) = 0, j = 1,2,...m \tag{3}$$

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

Here $s(a_i,b_j)$ is the score of aligning a_i with b_j and is defined as $\log\left[\frac{p(a_i,b_j)}{p(a_i)\times p(b_j)}\right]$, where $p(a_i)$ denotes the appearance 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.

4 Experiments

To evaluate the efficiency of the proposed methods, corpus I was construct based on the GENIA corpus [15]. 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 2600 sentences were left.

The corpus I was split into two parts; part I contains 1600 sentences which can be further split into two data sets: E_L consisting of 400 sentences with annotations and E_U consisting of the remaining 1200 sentences without annotations, part II consists of 1000 sentences which was used as the test data set.

4.1 Choosing Proper *k*

The E_L data in Part I of Corpus I were split randomly into the training set and the validation set at the ration of 9:1. The validation set consists of 40 sentences and the remaining 360 sentences were used as the training set. Experiments were conducted ten times (i.e Experiment 1, 2, 3,..,9 in Figure 3) with different training and validating set each round. At each round, a set of experiments were conducted with k set to 1, 3, 5, 7. Figure 3 shows the classification precision of KNN with different k values, where precision is defined as $Precision = TP/(TP + FP)$. Here, TP is the number of sentences that have been assigned with the correct annotations, FP is the number of sentences that do not get the correct annotations. It can be observed from Figure 3 that the overall best performance was obtained when k is set to 3.

4.2 Extraction Results

The baseline HVS model was trained on E_L from the part I of Corpus I which consists of 400 sentences. Sentences from data set E_U were then automatically assigned with semantic annotations using the KNN method described in section 3.1. The HVS model were incrementally trained with these newly added training data. Total 187 sentences from the un-annotated training data were successfully assigned with the semantic annotations.

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 as $\frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall}+\text{Precision}}$, where Recall is defined as TP/(TP+FN) and Precision is defined as TP/(TP+FP). All these values are calculated automatically.

Figure 4 shows the protein-protein interactions extraction performance versus the number of un-annotated sentences added. It can be observed that in general the F-score value increases when increasingly adding more un-annotated data from E_U. The best performance was obtained when adding in 187 un-annotated sentences where F-score reaches 58.9%.

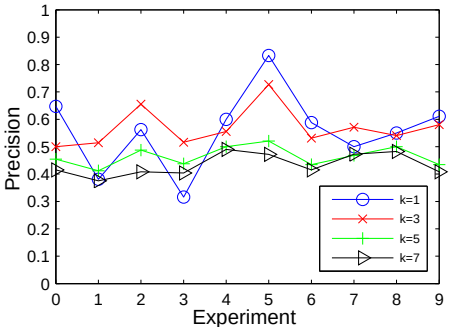

Fig. 3. Classification precision vs. different k value.

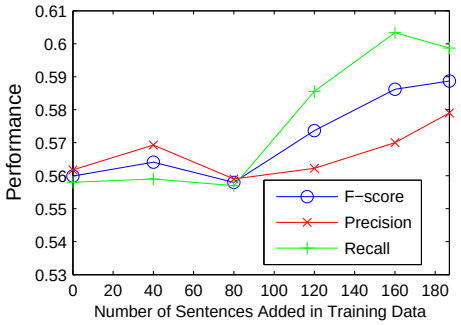

Fig. 4. Performance vs the amount of added un-annotated sentences.

5 Conclusions and Future work

In this paper, we presented a novel method to automatically train the HVS model from un-annotated data. Using the modified KNN algorithm, semantic annotations can be automatically generated for 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 other semi-

supervised learning methods to improve the classification performance. In addition, the current approach can only assign the existing semantic annotations to those un-annotated sentences. It would be interesting to incorporate bootstrapping style approach to derive the semantic annotations not just limited to the annotated corpus.

References

1. Minlie Huang, Xiaoyan Zhu, and Yu Hao. Discovering patterns to extract protein-protein interactions from full text. *Bioinformatics*, 20(18):3604–3612, 2004.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2005.
9. Rosie Jones. *Learning to extract entities from labeled and unlabeled text.* PhD thesis, Carnegie Mellon University, 2005.
10. 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.
11. David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.
12. Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
13. Yulan He and Steve Young. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106, 2005.
14. Eric Brill. Some Advances in Transformation-Based Part of Speech Tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.
15. JD. Kim, T. Ohta, Y. Tateisi, and J Tsujii. GENIA corpus–semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–2, 2003.