Extracting Protein-Protein Interactions from the Literature Using the Hidden Vector State Model

Deyu Zhou, Yulan He, and Chee Keong Kwoh

School of Computer Engineering, Nanyang Technological University,
Nanyang Avenue, 639798, Singapore
{zhou0063, asylhe, asckkwoh}@ntu.edu.sg

Abstract. In the field of bioinformatics in solving biological problems, the huge amount of knowledge is often locked in textual documents such as scientific publications. Hence there is an increasing focus on extracting information from this vast amount of scientific literature. In this paper, we present an information extraction system which employs a semantic parser using the Hidden Vector State (HVS) model for protein-protein interactions. Unlike other hierarchical parsing models which require fully annotated treebank data for training, the HVS model can be trained using only lightly annotated data whilst simultaneously retaining sufficient ability to capture the hierarchical structure needed to robustly extract task domain semantics. When applied in extracting protein-protein interactions information from medical literature, we found that it performed better than other established statistical methods and achieved 47.9% and 72.8% in recall and precision respectively.

1 Introduction

Understanding protein functions and how they interact gives researchers a deeper insight into understanding of living cell as a complex machine, disease process and provides target for effective drug designs. To date, many databases, such as PDB [1], Swiss-Prot [2] and BIND [3], have been built to store various types of information for protein. However, data in these databases were mainly hand-curated to ensure their correctness and thus limited speed in transferring textual information into searchable structure data. 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.

Existing approaches can be broadly categorized into two types, based on simple pattern matching, or employing parsing methods. Approaches using pattern matching [4, 5, 6] rely on a set of predefined patterns or rules to extract protein-protein interactions. For example, Ono's method [5] manually defines some rules and patterns which are augmented with additional restrictions based on syntactic categories and word forms to give better matching precision. It achieves high performance with a recall rate of 85% and precision rate of 84% for Saccharomyces cerevisiae (yeast) and Escherichia coli. Another method [6] tries to

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 718–725, 2006.

use dynamic programming to automatically discover patterns which describe protein-protein interactions. Their results give precision of 80.5% and recall of 80.0%. It is however not feasible in practical applications as it requires heavy manual processing to define patterns when shifting to another domain.

Parsing based methods employ either full or shallow parsing. Unlike word-based pattern matchers, shallow parsers [7, 8] break sentences into none overlapping phases. They extract local dependencies among phases without reconstructing the structure of an entire sentence. The precision and recall rates reported for shallow parsing approaches are estimated at 50-80% and 30-70%, respectively.

Systems based on full-sentence parsing [9, 10, 11] deal with the structure of an entire sentence and therefore are potentially more accurate. Yakushiji [9] defines grammars for biomedical domain and uses a general full parser to extract interaction events. However, no recall or precision value using this approach was reported. Another full parser-based approach uses the context-free grammar to extract protein interaction information with a recall rate of 63.9% and a precision rate of 70.2% [10]. 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.

In this paper, we describe a statistical method using the hidden vector state model (HVS) to automatically extract protein-protein interactions from biomedical literature. The HVS model has been successfully used to discover semantic information in spoken utterances [12]. 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. Thus, 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.

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 the biomedical literature. Section 3 presents the overall structure of the extraction system. 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 [12] 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 Fig. 1 which shows the sequence of HVS stack states corresponding to the given parse tree. State transitions are factored into separate stack pop and push operations constrained to give a tractable search space. 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 string W, and then extracts semantic concepts C from this tree. Each semantic concept

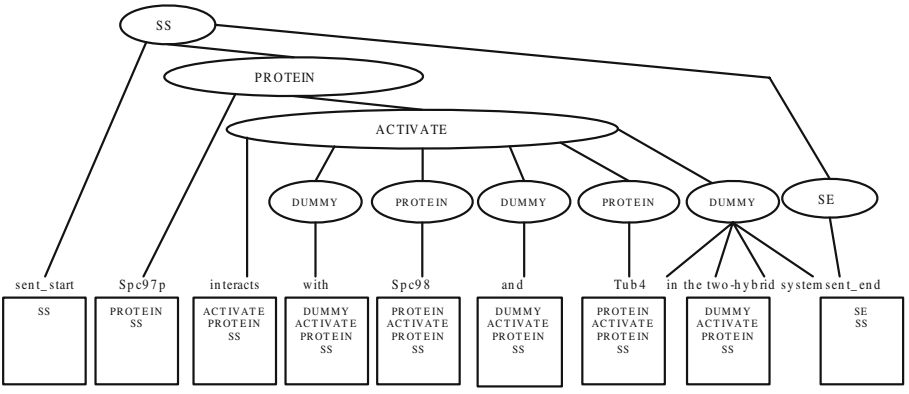

Fig. 1. Example of a parse tree and its vector state equivalent

consists of a name-value pair where the name is a dotted list of primitive semantic concept labels. For example, the top part of Fig. 1 shows a typical semantic parse tree and the semantic concepts extracted from this parse would be in equation 1

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

In the HVS-based semantic parser, conventional grammar rules are replaced by three probability tables. Let each state at time t be denoted by a vector of D_t semantic concept labels (tags) $c_t = [c_t[1], c_t[2], ..c_t[D_t]]$ where $c_t[1]$ is the preterminal concept label and $c_t[D_t]$ is the root concept label (SS in Fig. 1). Given a word sequence W, 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

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

where n_t is the vector stack shift operation and takes values in the range $0, \cdots, D_{t-1}$, and $c_t[1] = c_{w_t}$ is the new pre-terminal semantic label assigned to word w_t at word position t.

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

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

Each of these 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 string W is marked with the set of semantic concepts C that it contains. For example, if the sentence shown in Fig. 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 utterance 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 [12].

3 System Overview

The extraction system works as follows. At the beginning, abstracts 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, information about protein-protein interactions is extracted from the tagged sentences using a set of manually-defined simple rules. An example of the procedure is illustrated in Fig. 2.

The details of each step are described below.

1. *Identification of protein names, other biological terms and interaction keywords.*
   To extract information on protein-protein interactions from literature, protein names need to be first identified, which still remains as a challenging problem. In our system, protein names are identified based on a dictionary of manually constructed biological term. In addition, a category/keyword dictionary for identifying terms describing interactions has also been built

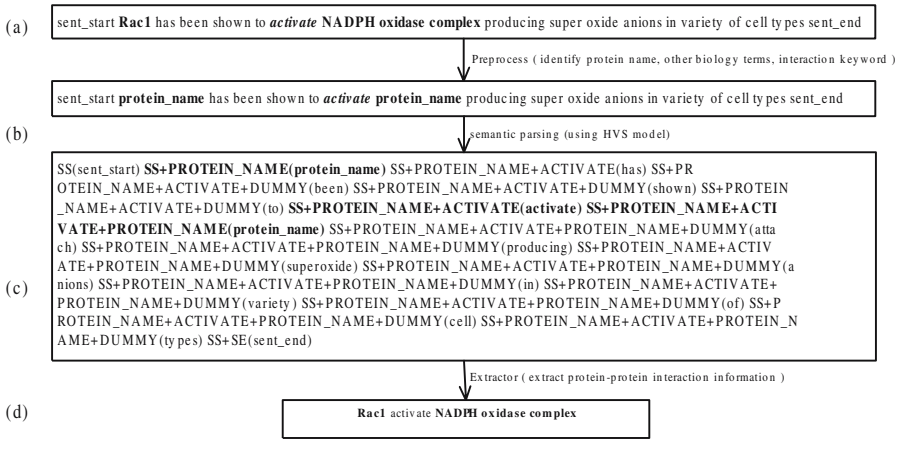

Fig. 2. An example of a procedure for information extraction using the HVS model

based on [10]. All identified biological terms and interaction keywords are then replaced with their respective category labels as can be seen in Fig. 2(b). By doing so, the vocabulary size of the training corpus can be reduced and the data sparseness problem would be alleviated.

2. *Parsing sentences using the HVS model.*
   A sentence which contains at least two proteins identified by Step 1 is then parsed with the HVS model. Before doing so, the HVS model needs to be trained using a lightly annotated training corpus. An annotation example is shown below.

   Sentence: CUL-1 was found to interact with SKR-1, SKR-2, SKR-3, SKR-7, SKR-8 and SKR-10 in yeast two-hybrid system
   Annotation: PROTEIN_NAME (ACTIVATE (PROTEIN_NAME))

   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 [12].
3. *Extraction of protein-protein interactions.*
   Given the HVS parsing result as shown in Fig. 2(c), the protein-protein interactions can be easily extracted follows the rules below:
   - ignore the semantic tag if its preterminal tag is DUMMY;
   - if the semantic tag is of the form SS+PROTEIN_NAME+REL+PROTEIN_NAME, SS+REL+PROTEIN_NAME+PROTEIN_NAME, and so on, REL can be any of the category names describing the interactions such as "activate", "inhibit" etc, extract the corresponding protein name, then search backwards or forward for the interaction keyword and the other protein name.

   Based on the rules described above, the protein-protein interactions can be easily extracted as shown in Fig. 2(d).

4 Results and Discussion

Experiments have been conducted on the two corpora. The corpus I was obtained from [6]. 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 interactions. After cleaning up the sentences which do not provide protein interaction information, 800 sentences were kept.

The corpus II comprises of 300 abstracts randomly retrieved from MEDLINE. These abstracts were then split into sentences and those containing more than two protein names were kept. Altogether 722 sentences were obtained. Note that these two corpora are disjoint sets.

Two tests were performed. In the first test, the corpus I was 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. The experiments were conducted three times with different training and test

Table 1. Results evaluated in sentence level on Corpus I

Experiment	TP	TP+TN	Recall(%)
1	59	80	73.8
2	65	80	81.3
3	63	80	78.8

Table 2. Results evaluated in interaction level on corpus I

Experiment	TP	TP+TN	FP	Recall(%)	Precision(%)	F-Score(%)
1	55	138	18	39.9	75.3	52.1
2	64	130	29	49.2	68.8	57.4
3	67	140	25	47.9	72.8	57.8

Table 3. Comparsion of Results on corpus I and corpus II

Experiment	F-Score(%)
corpus I	56.0
corpus II	50.4

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.

Table 1 shows the recall values evaluated at the sentence level. True Positive (TP) is the number of sentences which contain at least one correctly extracted protein interaction. (TP+TN) is the total number of sentences which contain protein-protein interactions. As corpus I does not have negative examples, this value is always 80. It can be seen from Table 1 that the best possible recall value that can be achieved is 81.3%.

Table 2 shows the evaluation results measured in the interaction level. 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:

$$\text{F-score} = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \tag{3}$$

In the second test, the HVS model is trained on corpus I and tested on corpus II. Table 3 gives comparison of the results between on corpus I and on corpus II. The value on corpus I is average of results based on Table 2. It was observed that a F-score of 50.4% was achieved when tested on a general corpus randomly extracted from MEDLINE, which is impossible to those systems based on predefined semantic grammar rules. For example, MedScan [13] can only successfully parse and generate semantic structures for about 34% sentences randomly picked from MEDLINE. The recall rate of MedScan was found to be 21% [13]. This demonstrated the robustness of the HVS model.

Generally, it is difficult to compare our method with other existing systems fairly, because there is neither an accurate task definition on processing the

MEDLINE abstracts nor a standard benchmark dataset. Since the corpus I data used in our experiments came from [6], it would be interested to see how our system performed compared to the method based on pattern matching proposed in [6]. If simply comparing the respective recall and precision rates, our method is less efficient. However, by examining the experimental results more carefully, we have the following findings:

1. The method proposed in [6] employed a part-of-speech (POS) tagger to preprocess the data. Some tags such as adjective, determiner and so on were removed. Since some interactions can be defined by adjectives, it therefore inevitably affected the system performance.
   For example, The sentence "The class II proteins are expressed constitutively on B-cells and EBV-transformed B-cells, and are inducible by IFN-gamma on a wide variety of cell types." provides an protein-protein interaction as shown by the underlying text. However, in [6], adjectives such as "inducible" were excluded and the system thus failed to extract the above interaction. On the contrary, our system was able to give the correct result.
2. Our system is able to generate reasonable results on a general domain as illustrated in the experiments on the corpus II, whilst the method proposed in [6] did not provide any results in this aspect.

5 Conclusions

In this paper, we have presented a system using the HVS model to automatically extract information on protein-protein interactions from text sources. The system is able to give reasonable performance measured in recall and precision. We have also shown the robustness of the system as it can be used in any general biomedical domain. Our results may provide a useful supplement to manually created resources in established public databases.

In future work we will work on the enhancement of the HVS model in order to improve the extraction accuracy. We will also study the adaptation issue of the HVS model and see how the model could give better performance by providing a small amount of adaptation data when the HVS model trained on one particular protein domain is used in another protein domain.

Acknowledgements

The authors would like to thank Minlie Huang of Tsinghua University, China, for providing the corpus I data.

References

1. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, pages 235–242, 2000.

2. B. Boeckmann, A. Bairoch, R. Apweiler, M. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. ODonovan, and I. Phan. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Research*, pages 365–370, 2003.
3. GD. Bader, D. Betel, and CW. Hogue. Bind: the biomolecular interaction network database. *Nucleic Acids Research*, 31(1):248–250, 2003.
4. 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.
5. 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.
6. Minlie Huang, Xiaoyan Zhu, and Yu Hao. Discovering patterns to extract protein-protein interactions from full text. *Bioinformatics*, 20(18):3604–3612, 2004.
7. Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86, Heidelberg, Germany, 1999. AAAI Press.
8. 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.
9. 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, 2001.
10. 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.
11. Shengyang Tang and Chee Keong Kwoh. Cytokine information system and pathway visualization. *International Joint Conference of InCoB, AASBi and KSBI (BIOINFO2005).*, 2005.
12. Yulan He and Steve Young. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106, 2005.
13. S. Novichkova, S. Egorov, and N. Daraselia. Medscan, a natural language processing engine for medline abstracts. *Bioinformatics*, 19(13):1699–1706, 2003.