SEMI-SUPERVISED PATTERN LEARNING FOR EXTRACTING RELATIONS FROM BIOSCIENCE TEXTS

SHILIN DING[†], MINLIE HUANG[†], XIAOYAN ZHU[*]

State Key Laboratory of Intelligent Technology and Systems (LITS), Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China,

dingsl@gmail.com, minlie.huang@hotmail.com, zxy-dcs@tsinghua.edu.cn

A variety of pattern-based methods have been exploited to extract biological relations from literatures. Many of them require significant domain-specific knowledge to build the patterns by hand, or a large amount of labeled data to learn the patterns automatically. In this paper, a semi-supervised model is presented to combine both unlabeled and labeled data for the pattern learning procedure. First, a large amount of unlabeled data is used to generate a raw pattern set. Then it is refined in the evaluating phase by incorporating the domain knowledge provided by a relatively small labeled data. Comparative results show that labeled data, when used in conjunction with the inexpensive unlabeled data, can considerably improve the learning accuracy.

1. Introduction

Knowledge extraction from bioscience texts has become an emerging field for both Information Extraction and Natural Language Processing communities. These tasks include recognizing biological named entities [10, 21], extracting relations between entities [4, 12, 19], and identifying biological events and scenarios [20]. The major challenges come from the fact that biomedical literatures contain abundant domain-specific knowledge, inconsistent terminologies, and complicated syntactic structures or expressions.

In this paper, the work is focused on extracting relations between biological entities, such as protein-protein interactions (PPI). Various methods and systems have been proposed. The most prevalent methods are rule-based or pattern-based. Such methods adopt hand-coded rules or automated patterns and then use pattern matching techniques to capture relations. Hand-coded patterns are widely used in the early stage of this research. For example, Ono [11] manually constructed lexical patterns to match linguistic structures of sentences for extracting protein-protein interactions. Such methods contribute high accuracy but low coverage. Moreover, the construction of patterns is time-consuming and requires much domain expertise.

Systems which can learn patterns automatically for general relation extraction include AUTOSLOG [14], CRYSTAL [17], SRV [6], RAPIER [1], ONBIRES [7, 8], and so forth. Most of them take annotated texts as input, and then learn patterns semi-

[†]The two authors have equal contributions

[*]Corresponding author: zxy-dcs@tsinghua.edu.cn Tel: 86-10-62796831 Fax: 86-10-62771138

automatically or automatically. But effective evaluation of these patterns remains a major unsolved problem. Moreover, most pattern-based applications require a well-annotated corpus for training [2, 5].

Since data annotation is expensive and time-consuming, the major problem in pattern-based methods is how to automatically learn patterns efficiently and effectively, with limited annotated data available. Unsupervised principle is preferable with the ability to exploit huge amount of unlabeled texts in biomedical domain. The crucial problem here is that patterns generated from unlabeled data may be erroneous or redundant. Therefore, pattern evaluation algorithm is indispensable. A systematic methodology based on ranking functions is widely used by most methods [3, 15, 18]. Such algorithms assign a score to each pattern according to the ranking functions and then keep the top n best patterns (n is a pre-specified threshold). In such algorithms, each pattern is evaluated independently. Thus, the redundancy among patterns is difficult to reduce.

To solve these problems, a semi-supervised [16, 22] model is proposed, combining both unlabeled and labeled data. A pattern generation algorithm is first implemented to mine relevant pattern structures from unlabeled data, where sentences are pairwise aligned by dynamic programming to extract identical parts as the pattern candidates. Since the generation algorithm does not require any annotation in the corpus, pattern evaluation algorithms with labeled information are then integrated to complete the learning procedure. Two types of pattern evaluation algorithms are investigated. The first is a ranking function based algorithm, which evaluates the effectiveness of every single pattern independently and delete the ones that have no contribution to the performance. The second is heuristic evaluation algorithm (*HEA*), which aims to search the optimal pattern set in a heuristic manner. Compared to the first method, the deletion of a pattern is determined by the current pattern set, not only itself. Comparative results show that our ranking function outperforms other prevalent ones and *HEA* exhibits advantages over ranking function based algorithms.

The paper is organized as follows. The first part of semi-supervised learning, pattern generation method with unlabeled data, is presented in Section 2. Then pattern evaluation method which relies on labeled data to curate the learning result is explained in Section 3. The experiments and conclusions are discussed in Section 4 and 5.

2. Pattern Generation

First of all, several definitions are presented here:

- A Sentence is a sequence of word-tag pairs: $STN = WTP_{1,2,\ldots,N}$, where each WTP_i is a word-tag pair (w_i,t_i). Here w_i is a word and t_i is the part-of-speech (POS) tag of w_i.
- Sentence Structure is defined as $SS = \{prefix, NE_1, infix, NE_2, suffix\}$. NE_1 and NE_2 are semantic classes of the named entities. The *prefix, infix, and suffix* are sequences of *WTPs* before NE_1, between NE_1 and NE_2, *and* behind NE_2, respectively.
- A pattern is defined as $PTN=\{pre\text{-}filler, NE_1, mid\text{-}filler, NE_2, post\text{-}filler\}$. The *fillers* are sequences of *WTPs* before NE_1, between NE_1 and NE_2, and behind NE_2.

Examples for these definitions are shown in Table 1.

Table 1. Examples for sentences and patterns

	Examples
STN	Several/JJ recent/JJ studies/NNS have/VBP implicated/VBN P_00172/NN in/IN the/DT signaling/NN pathway/NN induced/VBN by/IN P_00006/NN ./.
WTP	induced/VB ; P_00172/NN
SS	{{*prefix*: {Several/JJ recent/JJ studies/NNS have/VBP implicated/VBN}} {NE_1: {PROTEIN/NN}} {*infix*: {in/IN the/DT signaling/NN pathway/NN induced/VBN by/IN}} {NE_2: {PROTEIN/NN}} {*suffix*: {NULL}}}
PTN	{{*pre-filler*: {NULL}} {NE_1: {PROTEIN/NN}} {*mid-filler*: {induced/VBN by/IN}} {NE_2: {PROTEIN/NN}} {*post-filler*: {NULL}}}

Sequence alignment algorithm is adopted to generate patterns by aligning pairwise sentences in training corpus. The identical parts in aligned sentences are extracted as pattern candidates. More formally, given two *SSs*: ($prefix^i$, NE_1^i, $infix^i$, NE_2^i, $suffix^i$) and ($prefix^j$, NE_1^j, $infix^j$, NE_2^j, $suffix^j$), the sequence alignment algorithm is carried out on three pairs – ($prefix^i$, $prefix^j$), ($infix^i$, $infix^j$), and ($suffix^i$, $suffix^j$) – to extract identical *WTPs* and form the three fillers of a *PTN*. The algorithm is shown in Figure 1.

Input: A sentence structure set $S = \{SS_1, SS_2, \cdots, SS_n\}$
Output: A set of patterns: P
1. **For** every pair in S: $(SS_i, SS_j) \in S$ $(i \neq j)$, do
2. **if** $SS_i.NE_1$!= $SS_j.NE_1$ **or** $SS_i.NE_2$!= $SS_j.NE_2$, **then** go to 1;
3. **else do**
4. $NE_1 = SS_i.NE_1$, $NE_2 = SSi_j.NE_2$
5. do alignment for $SS_i.prefix$ and $SS_j.prefix$;
6. extract the identical *WTPs* to form the *pre-filler* of a candidate pattern p.
7. do the same operations in step 5 and 6 to form *mid-filler* and *post- filler*,.
8. **if** p already exists in P, **then** increase the count of p with 1.
9. **else** add p to P with a count of 1;
10. Output P

Figure 1. Pattern learning algorithm

This algorithm automatically learns patterns from sentences whose named entities have been identified by a dictionary-based method and requires no further annotation. It is almost unsupervised which is able to make better use of the enormous data available online, and release domain experts from the heavy burden of creating annotated corpora.

3. Pattern Evaluation

The pattern generation algorithm discussed in Section 2 does not require supervised information. It may produce erroneous patterns such as (“”, PROTEIN, “shown to be”,

PROTEIN, “”), which will match many false positive instances. Previous works usually depended on rule-based methods or manual selection to screen out the best patterns.

To automate the relation extraction system, we developed a pattern evaluation algorithm to assess patterns by a small annotated corpus. Here we discuss two types of evaluation algorithms: the first one utilizes ranking functions and the second one is a heuristic evaluation algorithm.

3.1. Ranking Function Based Algorithm

Ranking function based evaluation algorithms assess each pattern independently. They assign a score to each pattern by ranking functions, and then filter out those patterns with lower scores than a threshold. Previous pattern-based systems have adopted various ranking functions, which take into consideration the number of instances that are correctly or incorrectly matched by a pattern. Two ranking functions are surveyed here:

The first one is proposed by Cohen in their system RIPPER [3]:

$$Ripper(p) = \frac{p.positive - p.negative}{p.positive + p.negative} \tag{1}$$

where *p.negative* indicates the number of false instances matched by the pattern *p* and *p.positive* denotes the number of correct instances. In essence, this function only takes into consideration the ratio of *p.positive* to *p.negative* (*p/n* for short). The second function is proposed by Riloff [15], with two factors – *p/n* and *p.positive*:

$$Riloff(p) = \frac{p.positive}{p.positive + p.negative} * \log_2(p.positive) \tag{2}$$

The critical issue about these ranking functions presented above is that only two factors, *p/n* and *p.positive*, are considered. However, other factors should be considered, such as *p*+*n*. *Ripper* can not distinguish a pattern with 50 true positives and 50 negatives (50/50 for short) from (1/1) pattern, while the former pattern apparently contributes more to precision and recall. Although *Riloff* function works well for the two patterns by introducing the log(*p.positive*) term, it does not work for such 4 patterns: (1/4), (2/8), (3/12) and (4/16). These patterns, whose *p.positive* is larger, will have a higher rank by *Riloff* function. However, since p/n is very low, it is reasonable to determine that patterns with larger (*p*+*n*) are worse. *Riloff* function fails to handle the case. To involve more factors for pattern evaluation, we propose a novel ranking function as follows:

$$HD(p) = (\beta + \log_2 \frac{p.positive + 0.5}{p.negative + 0.5}) * \ln(p.positive + p.negative + 1) \tag{3}$$

where the parameter β is a threshold that controls *p/n*. If $p/n > 2^{-\beta}$, *HD* is an increasing function of (*p*+*n*), which means if several patterns have the same *p/n* that exceeds $2^{-\beta}$, a pattern with larger (*p*+*n*) has a higher rank. If $p/n < 2^{-\beta}$, the first term is negative, which means a pattern with larger (*p*+*n*) will have a lower rank. Thus different ranking strategies are used when different p/n are met. Experiments in Section 4.3 will illustrate how *HD* outperforms other functions, where the parameter β is set to 0.5 empirically.

3.2. Heuristic Evaluation Algorithm (HEA)

Ranking function based algorithms assess each pattern independently. It is not difficult to delete erroneous patterns by these algorithms. However, redundancy among patterns, which heavily impose computational burden on relation extraction tasks, can not be reduced effectively. For example, there are two patterns (“”, PROTEIN, “bind to”, PROTEIN, “”) and (“”, PROTEIN, “to bind to”, PROTEIN, “”). Apparently, the second pattern is redundant since all instances it matches will also be captured by the first one. However, it cannot be filtered because its score is almost the same as the first ones. To remove erroneous and redundant patterns, we propose a heuristic evaluation algorithm (*HEA*), which aims to obtain the optimal pattern set in a heuristic manner.

Formally, given an evaluation corpus S and a pattern set P, we define an optimization function which maps the pattern set P to its performance on S:

$$\begin{aligned} M(S,\cdot): \Sigma &\to R \\ P &\mapsto M(S,P) \end{aligned} \tag{4}$$

where Σ denotes the space of all possible pattern sets and R is the real number space. Starting from the initial pattern set P_0, we aim to obtain the optimal set P^* by maximizing $M(S,P)$ in a heuristic manner. The iterative procedure follows formula (5):

$$P_{k+1} = P_k - \arg\limits_{p_i \in P_k} \nabla M(S, P_k) \tag{5}$$

where $\nabla M(S,P_k) = \dfrac{\partial M(S,P_k)}{\partial P_k} = \max\limits_{p_i \in P_k} \{M(S, P_k - \{p_i\}) - M(S,P_k)\}$ is the gradient of $M(S,P_k)$ in k-th step.

The algorithm is shown in Figure 2. In practice, we store and index all possible matching results produced by the whole pattern set P_0 by preprocessing. Thus, for each iteration, evaluating the pattern set P_k is carried out by finding the results in the index (excluding all the patterns that are not in P_k), without a whole re-running of the program. This method makes the iterative procedure computationally feasible.

Input: an initial pattern set $P_0 = \{p_1, p_2, \cdots, p_n\}$, the training set S, the testing set T, an optimization function $M(S,P)$
Output: the optimal pattern set P^*

1. k=0, $P_k = P_0$
2. Calculate the gradient:
   $\nabla M(S, P_k) = \max\limits_{p_i \in P_k} \{M(S, P_k - \{p_i\}) - M(S, P_k)\}$
3. Find the “worst” pattern to be deleted:
   $p_m = \arg\limits_{p_i \in P_k} \nabla M(S, P_k)$
4. **If** $\nabla M(S, P_k) \geq \Delta_{th}$ **then do**
5. $M(S, P_{k+1}) = M(S, P_k - \{p_m\})$
6. $P_{k+1} = P_k - p_m$
7. Evaluate the performance of the pattern set P_{k+1} on the testing corpus T
8. k=k+1, go to 2
9. **else** output $P^* = P_k$

Figure 2. Heuristic Evaluation Algorithm (*HEA*)

In this algorithm, an optimization function $M(S,P)$ has to be determined. Note that the goal of *HEA* is to achieve the optimal pattern set out of an initial set. Thus, the direct target of F_1 score can be taken as an optimization function.

4. EXPERIMENT

Corpora used in the experiments are introduced in Section 4.1. Experiments of pattern generation on unlabeled data and pattern evaluation on labeled data are discussed in detail in Section 4.2 and 4.3. These sections are aimed to investigate the effectiveness of the semi-supervised learning model.

4.1. Data Preparation

The first corpus used for protein-protein interaction extraction is downloaded from http://www.biostat.wisc.edu/~craven/ie/ [13]. This corpus consists of 2,430 sentences gathered from Munich Information Centre for Protein Sequences (MIPS). This corpus is used for pattern generation.

The second is collected from the GENIA corpus [9] which consists of 2000 abstracts from MEDLINE. We have manually annotated the protein-protein interactions, and finally obtained a corpus with 4,221 PPIs in 2,561 sentences.

4.2. Semi-supervised Learning Model

In this section, we discuss the effectiveness of semi-supervised learning model by comparing the performance of refined patterns with that of original patterns.

First, 2,480 patterns are initially obtained from MIPS by the generation algorithm. Their performance is set as the baseline. Then the GENIA corpus with annotated relations is randomly partitioned to five parts for 5-fold cross-validation, one of the five

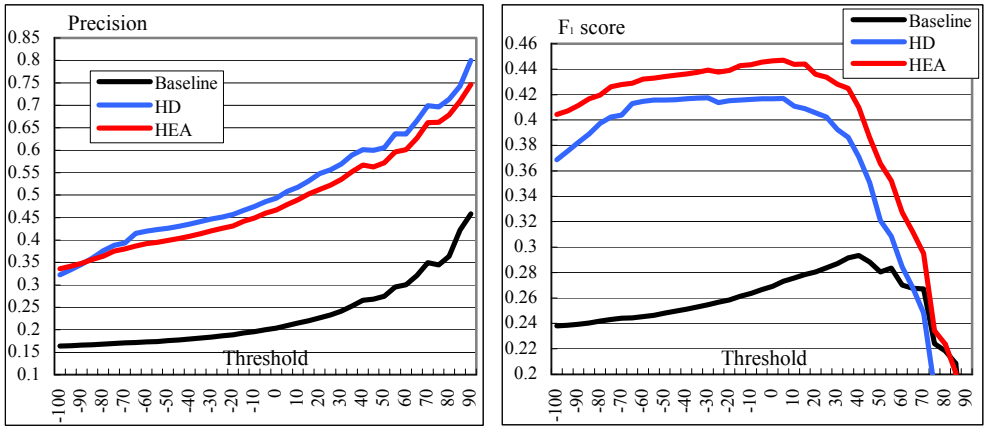

Figure 3. Performance of semi-supervised model

parts for testing and the remainder for pattern evaluation. For the two evaluation methods, the top 100 patterns are preserved to extract relations from the testing corpus.

Experiment results are shown in Figure 3 over different user-specified thresholds on the testing corpus. It shows that 1) the raw pattern set generated without labeled information is poor in accuracy, but has promising recall (about 45% to 50%). 2) Our proposed ranking function *HD* and *HEA* method both achieve significant improvements. The precision is improved by over 25% with little loss in recall, results in improvement of F_1 score by 16% to 19%. These results indicate that the pattern generation algorithm does extract useful patterns from unlabeled data, and pattern evaluation algorithm greatly improves the accuracy with labeled information.

4.3. Pattern Evaluation With Labeled Data

In this section, we discuss the difference among different evaluation algorithms, which is crucial in semi-supervised learning. The GENIA corpus is used in the same way as before for 5-fold cross-validation. The raw pattern set is also taken from the previous experiment. In this experiment, the pattern deletion order in ranking function based methods, including *Ripper*, *Riloff*, and *HD*, is determined by the corresponding functions. In other words, patterns with lower ranks (worse patterns) will be removed earlier. In *HEA*, which pattern to be deleted is determined dynamically as before. To provide a complete comparison, we delete all of the patterns in each algorithm, which means the parameter Δ_{th} in *HEA* could be set to a very small numerical value.

Table 2. Performance of optimal pattern sets determined by ranking function based algorithms and *HEA*

Method	Patterns	Precision	Recall	F_1 score	Impr. of F_1
Baseline (raw)	2480	19.0%	46.5%	27.0%	–
Ripper	1626	41.0%	40.8%	40.9%	+49.3%
Riloff	88	40.8%	44.5%	42.6%	+55.5%
HD	**92**	52.5%	38.8%	**44.5%**	**+62.4%**
HEA	**72**	43.5%	45.9%	**45.5%**	**+66.1%**

Table 2 shows the performance and cardinality of the optimal pattern sets achieved by different methods. The smallest pattern set and the best system performance are achieved by *HEA*, which means *HEA* can reduce redundancy maximally and guarantee the best system performance at the same time. Although the performance of *HD* and *HEA* is only slightly better than that of *Ripper* and *Riloff* function, further studies in Figure 4 will demonstrate why *HD* and *HEA* outperform the other two methods.

When the $2{,}480^{th}$~$1{,}600^{th}$ patterns determined by *Ripper* function are deleted, the performance is enhanced dramatically. However, when it starts to delete 50 more patterns, the performance degrades extremely. These patterns include (“”, NE1, “inhibit”, NE2), (“”, NE1, “induce”, NE2, “”) with both large *p.positive* and large *p.negative* (79/142 and 308/220 in our experiment), but the p/n is not large enough (compared to the 3/1 or 7/3

patterns), which directly leads to low ranks. Therefore, *Ripper* function that involves only the p/n factor can not assess patterns properly.

Riloff function is also unable to evaluate patterns adequately. Firstly, the "worst" 900 patterns ranked by *Riloff* function are not the worst in fact, because deleting these patterns does not lead to remarkable improvements. However, deleting the 900^{th}~100^{th} patterns results in significant improvements. Hence these patterns should have much lower ranks. Secondly, although the best result (42.6% with 88 patterns) is very promising, the curve keeps rising until it reached the optimal point at a very narrow peak. Thus it is very difficult to determine the number of patterns to hold in practice (The system performance is very sensitive to the threshold).

In comparison, *HD* function exhibits advantages over traditional ranking functions. The *HD* curve shows that it removes the most undesirable patterns at position $2{,}460^{th}$ ~$1{,}700^{th}$, with 16 percentages improvement of F_1 score. And then the curve rises slowly, when deleting "medium" patterns, until it reaches the optimal point (44.5% with 92 patterns). After that point, deleting any pattern will cause a remarkable decline in the performance. This curve shows that *HD* ranks the patterns more precisely. And it also has a much broader "safe" area (from the 500^{th} to 100^{th} patterns). Thus the number of patterns to hold is much easier to set, compared to the narrow peak in *Riloff* curve.

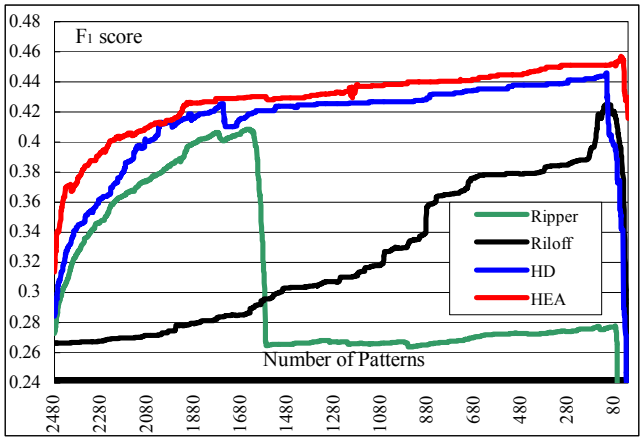

Figure 4. Comparison between ranking function based algorithm and *HEA*

In addition, the curve of *HEA* has almost the same trend as that of *HD*, which means *HEA* is also effective to remove incorrect and redundant patterns. The cardinality of the optimal pattern set obtained is smaller than that by *HD*, which means that *HEA* is more capable to reduce redundancy among patterns. From this figure, we can see that *HEA* outperforms other methods significantly.

The comparative experiments show that 1) algorithms based on traditional ranking functions fail to evaluate the contribution of the entire pattern set effectively because they take into account each pattern independently, while *HD* function is more reasonable by involving more factors; 2) *HEA* can remove erroneous and redundant patterns

effectively and consequently achieves a highest F_1 score (45.5%) with the smallest pattern set (72 patterns). Thus, the semi-supervised learning procedure can be carried out effectively and achieves a state-of-art performance.

5. CONCLUSION

Pattern-based methods have been widely used for the task of relation extraction from bioscience texts. However, most of these methods either construct patterns manually, or require a well-annotated training corpus to learn patterns. In this paper, we have proposed a semi-supervised model to automatically learn patterns with unlabeled and labeled data. Little domain expertise is required and vast texts available in biomedical domain can be fully exploited. Moreover, two types of pattern evaluation algorithms based on labeled information are proposed to remove erroneous and redundant patterns. The first one is based on a novel ranking function *HD* which takes into account more factors than prevalent ranking functions. Experimental results show that *HD* function exhibits advantages over other ranking functions. The second one is a heuristic evaluation algorithm, which aims to obtain the optimal pattern set in iterative steps. This algorithm contributes improvement over ranking function based algorithms.

We also note that the major bottleneck of pattern-based IE systems is whether they have an effective NLP module to handle complex syntactic structures in the bioscience texts. Currently we have a shallow parsing module to enhance the results; however, the future work will still focus on developing more competitive NLP techniques.

ACKNOWLEDGMENTS

The work was supported by Chinese Natural Science Foundation under grant No. 60572084 and 60321002.

REFERENCES

1. M. Califf and R. Mooney. Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction.*Journal of Machine Learning Research*, 4, 2003, pp. 177–210.
2. J. Chiang and C. Yu. Literature extraction of protein functions using sentence pattern mining. *IEEE. Trans. On Knowledge and Data Engineering*, 17(8), 2005, pp. 1088-1098.
3. W.W. Cohen. Fast effective rule induction. In *Proceedings of International Conference on Machine Learning*, Lake Tahoe, CA, 1995.
4. M. Craven. Learning to extract relations from Medline. *AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.
5. N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo. Extracting Human Protein Interactions from MEDLINE Using a Full-Sentence Parser. *Bioinformatics*, 20 (5), 2004, pp. 604- 611.
6. D. Freitag. Machine learning for information extraction in informal domains. *Machine Learning*, 39, 2000, pp. 169–202.

7. M.L. Huang, X.Y. Zhu, Y. Hao, D.G. Payan, K. Qu, and M. Li. Discovering patterns to extract protein-protein interactions from full texts. *Bioinformatics*, vol. 20, 2004, pp. 3604-3612.
8. M.L. Huang, X.Y. Zhu, S.L. Ding, H. Yu, and M Li. ONBIRES: ONtology-based BIological Relation Extraction System. In *Proceedings of the Fourth Asia Pacific Bioinformatics Conference*, Taiwan, China, 2006.
9. J.D. Kim, T. Ohta, Y. Teteisi, and J. Tsujii. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*. 19(suppl. 1), 2003, i180-i182.
10. K.J. Lee, Y.S. Hwang, and H.C. Rim. Two-Phase Biomedical NE Recognition based on SVMs. In *Proceeding of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, Sappora, Japan, 2003, pp. 33-40.
11. T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2), 2001, pp. 155-161.
12. C. Plake, J. Hakenberg, and U. Leser. Optimizing Syntax Patterns for Discovering Protein-Protein Interactions. *ACM Symposium on Applied Computing*,195-201, 2005
13. S. Ray, and M. Craven. Representing Sentence Structure in Hidden Markov Models for Information Extraction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence,* 2001, pp. 1273-1279.
14. E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., 1993, pp. 811–816.
15. E. Riloff. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996, pp. 1044-1049.
16. M. Seeger. *Learning with labeled and unlabeled data*, Technical Report, University of Edinburgh, 2001.
17. S. Soderland, D. Fisher, J. Aseltine, W. Lehnert. CRYSTAL: Inducing a Conceptual Dictionary. In *Proceedings of the Fourteenth International Joint Conference on Articial Intelligence* (IJCAI '95), 1995.
18. K. Sudo. *Unsupervised discovery of extraction patterns for information extraction*. Ph.D. Thesis, New York University, New York, NY. 2004.
19. J.M. Temkin, and M.R. Gilder. Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, 19, 2003, pp. 2046 - 2053.
20. A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii. Event extraction from biomedical papers using a full parser. In *Proceedings of Pacific Symposium on Bio-computing*, 2001, pp. 408-419.
21. G.D. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20 (7), 2004, pp.1178-1190.
22. X. Zhu. *Semi-Supervised Learning Literature Survey*, Technical Report No. 1530, Computer Sciences Department, University of Wisconsin, Madison, 2006