Journal of Computational Information Systems 11: 20 (2015) 7387–7395
Available at http://www.Jofcis.com

Co-training for Detecting Hedges and Their Scope in Biomedical Texts ⋆

Huiwei ZHOU*, Huijie DENG, Huan YANG, Degen HUANG, Yao LI

School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

Abstract

To avoid extracting uncertain statements as factual information, the detection of hedges and their scope becomes an important step in biomedical text mining. The current approaches focus on learning the detection models only with the labeled data. However, such approaches cannot make further progress due to the limited amount of training data and the difference between the training and working data. We proposes a co-training approach to make use of the limited labeled data to leverage some amounts of unlabeled data for boosting the detection performances of hedge cues and their scope. Experiments are carried out on the biomedical corpus of the CoNLL 2010 Shared Task and on free data derived from biomedical literature. Both the test data of the corpus and the free data are used as the unlabeled data. Experiment results show that the test data helps more than the free data on both tasks. The best F-score achieved in hedge cue identification is 88.12% and for hedge scope detection it is 63.09%, which significantly outperform previous systems. Co-training system can transfer the distribution of the unlabeled data to the labeled training data to improve the performance on the unlabeled data effectively.

Keywords: Hedge Cues; Hedge Scope; Co-training

1 Background

The term hedging was originally introduced by Lakoff [1]. When authors are not sure about their ideas or cannot verify their statements, they usually use hedging language to express the semantic uncertainty. Hedging is widely used in science texts, especially in the biomedical domain. The detection of hedges and their linguistic scope is important for factual information extraction.

In recent years, the detection of hedges has attracted more and more interest in natural language processing (NLP) community. The CoNLL 2010 Shared Task [2] formulates speculative language detection as two subtasks. Task 1 is to detect hedges and Task 2 is to detect the linguistic hedge scope of the cue words. For example, considering the following two sentences taken from the corpus of the CoNLL 2010 Shared Task:

⋆Project supported by the National Nature Science Foundation of China (No. 61272375, and No. 61173100).
*Corresponding author.
Email address: zhouhuiwei@dlut.edu.cn (Huiwei ZHOU).

1553–9105 / Copyright © 2015 Binary Information Press
DOI: 10.12733/jcis15646
October 15, 2015

(a) *These results indicate the utility of this FISH technique for a better definition of the biological characteristics of ductal carcinomas.*

(b) *This <xcope id="X433.9.2"><cue type="speculation" ref="X433.9.2"> indicates that </cue> functional differences observed in IL-2- and IL-12-stimulated cells <xcope id="X433.9.1" ><cue type="speculation" ref= "X433.9.1"> may </cue> depend, at least in part, on differential gene regulation </xcope></xcope>.*

Sentence (a) is a factual statement while sentence (b) contains two hedge cues: "*indicates that*" and "*may*". The word "*indicate*" plays different roles in the two sentences, acting as a hedge cue only in the second sentence. Each cue in sentence (b) has its own linguistic scope. The cue "*indicates that*" expresses that the statement "*indicates that functional differences observed in IL-2- and IL-12-stimulated cells may depend, at least in part, on differential gene regulation*" is uncertain, while "*may*" implies the speculative statement "*may depend, at least in part, on differential gene regulation*". These examples highlight some of the difficulties in detecting hedge cues and their scope.

The problem of identifying speculative sentences in biomedical articles is introduced by Light et al. [3]. Morante and Daelemans [4] developed a cue detector following a supervised approach. In the CoNLL 2010 Shared Task 1, Tang et al. [5] built a cascade system to detect hedges, which achieved the best performance with F-score 86.36% on the biological corpus. Zhou et al. [6] present a voting-based approach and achieved a higher F-score of 87.49%. Szarvas et al. [7] proposed a cross-domain and cross-genre semantic uncertainty recognition method.

As for hedge scope detection, Morante and Daelemans [4] formulated hedge scope detection problem as a labeling problem. Morante et al. [8] employed rich syntactic dependency features to a memory-based system. Velldal et al. [9] explored the rule-based approach and the data-driven approach. Zhou et al. [6] achieved an F-score of 60.87% with voting-based ensemble classifiers.

Most of the previous statistical approaches to hedge detection are based on an annotated corpus. However, such corpora are very rare and there is always a difference between the training and testing data. So the problem before us is how to use limited labeled data to leverage some amounts of unlabeled data to improve hedge detection. Co-training [10] is a semi-supervised learning algorithm that takes a set of labeled data as seeds to predict unlabeled data in multiple views to bootstrap the original models.

This paper proposes a co-training approach to boost the detection performance of hedge cues and their scope. In the initial form of co-training [10], the description of instances should be split into two independent views, each of which is sufficient to train a good classifier. Abney [11] showed that a co-training system can perform equally well under the condition of weak rule dependence. Co-training could improve the classification accuracy by extending a supervised learning to semi-supervised learning [12]. In our co-training system, two different supervise learning algorithms, Support Vector Machine (SVM) [13] and Conditional Random Field (CRF) [14] are employed to improve the performance of the uncertain information detection.

The CoNLL 2010 test data and the free data downloaded from the internet are taken as unlabeled data to train the co-training algorithm on hedge cue recognition and their scope detection tasks. Experiments show that the test data helps more than the free data on both tasks, because the predicted test data could help learning the test data distribution. Finally, competitive results are achieved with the combination of free data and test data, which not only enhances the performance of the system but also enlarges the hedging data set.

2 Methods

2.1 Co-training method

SVM and CRF are the two heterogeneous learning algorithms. The combination of the two algorithms can complement and facilitate each other. This paper investigates the co-training approach by using CRF and SVM algorithms.

The co-training approach is applied to the two sub tasks: hedge detection and hedge scope resolution. The applications of the co-training approach to the two tasks shares a general architecture as shown in Fig. 1.

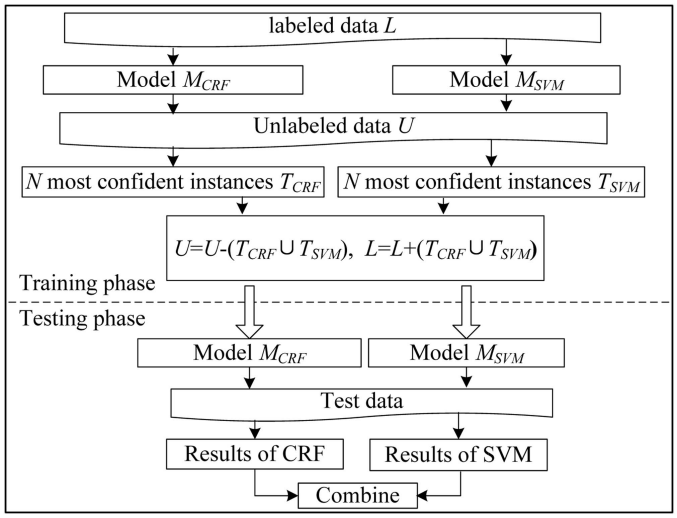

Fig. 1: System architecture

The architecture consists of a training phase and a testing phase. In the training phase, the SVM and CRF models learn from each other through the co-training process. Given a labeled training data set L and an unlabeled training data set U, the co-training process can be described as follows:

(1) Train the first model M_{CRF} on L;

(2) Predict U with M_{CRF}, and choose N most confidently predicted instances T_{CRF};

(3) Train the second model M_{SVM} on L;

(4) Predict U with M_{SVM}, and choose N most confidently predicted instances T_{SVM};

(5) Resolve the conflict between T_{CRF} and T_{SVM}, and acquire the union $T = T_{CRF} \bigcup T_{SVM}$ [1];

(6) Remove the instances in T from U, and add T to L with the predicted labels.

Such a process is repeated until T or U is empty and finally the two models M_{CRF} and M_{SVM} are obtained. In the testing phase, the two models are both applied to detect the test data, and then the confidence values predicted by the two models are combined and used to detect hedges.

[1]If the most confident factual sentences predicted by one model contains the most confident cue predicted by the other, the conflicting sentences will not be added to $T = T_{CRF} \bigcup T_{SVM}$.

2.2 Co-training in hedge detection

The hedge detection task is treated as a sequential labeling problem, in which each word in a sentence is labeled as B, I or O, respectively indicating the beginning of a cue, inside or outside.

The co-training algorithm used for hedge detection follows the process shown in Fig. 1. In the training phase, the labeling confidence values are given along with the predicting process on the unlabeled data set. The confidence value of a predicted cue (a positive instance) is calculated by the average predicted value of all the words in the cue, while a factual sentence (a negative instance) is calculated by that of all the words in the sentence.

At each iteration, the N most confident instances are selected. $N = 2 * (p + n)$, where p is the number of positive predicted cues and n is the number of negative (factual) sentence instances, for each of the two systems. In the hedge detection systems, the ratio of p over n instances added into the labeled data is fixed at 1/4 to balance the class distribution of the labeled data. Therefore at each iteration, if p hedge sentences (positive instances) are selected by one system, the maximum number of sentences added by the two systems is $10p$.

With the number of added instances increasing, more and more incorrectly predicted instances are added to the labeled data set. Therefore, the co-training performance will probably fall down unless a threshold parameter is set for instances selection. Only on condition that the labeling confidence value exceeds the threshold, can the positive instance be selected into the most confidently predicted result set. The threshold is determined, by a 4-fold crossover experiment on the training data set, as the smallest confidence value among the correctly predicted instances.

The two models learned in the training phase are used to predict the test data in the testing phase. The predicted values in the CRF model (the marginal probability) ranges from 0 to 1 while that in the SVM model (the distance to the separating hyper-plane) ranges from $-\infty$ to $+\infty$. Therefore the predicted values of one model should be mapped to the range of the other model when combining the two models. The predicted values of the two basic models are combined in two ways:

Tangent mapping: map the confidence values of the CRF model to $(-\infty, +\infty)$ by a monotonically increasing tangent function, and then add it to the confidence values of the SVM model as shown in Eq. (1), in which i denotes the ith token in the sentence; j denotes the jth label class (namely B, I, O); F_i^j is the final confidence value of the ith token to be predicted as the jth label; C_i^j and S_i^j are confidence values predicted by the CRF and the SVM models respectively. The final label of the ith token is determined by Eq. (2), i.e., the most confident label is the final result. The tangent function in Eq. (1) could expand the probability function nonlinearly to the range of $(-\infty, +\infty)$.

$$F_i^j = \tan((C_i^j - 0.5) \times \pi) + S_i^j \tag{1}$$

$$P_i = \arg\max_{j \in \{B,I,O\}} F_i^j \tag{2}$$

Exponential mapping: map the confident values of the SVM model to (0, 1) by a monotonically increasing exponential function, and then add it to the confidence values of the CRF model as shown in Eq. (3). The final label of the ith token is determined by Eq. (2), too. The exponential function in Eq. (3) could give a good approximation of probability function from the decision function of SVMs.

$$F_i^j = \frac{1}{1 + \exp(-S_i^j)} + C_i^j \tag{3}$$

2.3 Co-training in hedge scope detection

For the sentences that contain cues, their corresponding scopes should be detected further. In hedge scope detection, each token is labeled as F-scope, L-scope or NONE, respectively denoting the first token or the last token of hedge scope, or others.

The co-training algorithm used for hedge scope detection also follows the procedure shown in Fig. 1. In the training phase, the confidence value of a scope is calculated by the average value of the F-scope and L-scope token. At each iteration, N most confident instances include p sentences predicted by the CRF system and p sentences predicted by the SVM system. Therefore in the hedge scope detection systems, the maximum sentence amount added by the basic CRF and SVM systems at each iteration is $2p$. In the testing phase, the predicted values of the SVM and the CRF models are combined in the following two ways:

Tangent mapping: follow the same combining strategy as in Task 1. Normalize the CRF confidence values with the tangent function in (1), in which the ith token is denoted as the scope labels F-scope, L-scope or NONE. The scope labels of each token should then be determined by Eq. (2), with F/L/N instead of B/I/O.

Exponential mapping: follow the same combining strategy as in Task 1. Normalize the SVM confidence value with the exponential function in (3), and then get the scope tags.

3 Results and Discussion

3.1 Experimental settings

For unlabeled training data, two kinds of data sets are exploited to investigate the influence of data type on the co-training systems. Table 1 gives the detailed statistics of the experimental datasets. In Table 1, the third column shows the cue amount for "CoNLL" data or the keyword amount for the "Free" data. A keyword is a word that is used as a cue in "CoNLL" training dataset. The same features and the post-processing rules used in Zhou et al. [6] are applied in our experiments.

The performance for hedge cue and scope detection is evaluated by the official tool of the CoNLL 2010 shared task. In Task 1, the evaluation for hedge detection is carried out on the sentence-level and the cue-level. The sentence-level scores correspond to correctly identifying sentences as being certain or uncertain. A sentence is labeled "uncertain" if it contains at least one recognized cue. The cue-level scores are based on the exact match of cue phrases. The evaluation for Task 2 is based on exact match of both cues and scopes.

Table 1: Detailed statistics of the experimental data

Date Set	Number of Sentences	Number of Cues
CoNLL train	14541	3376
CoNLL test	5003	1047
Free	44569	15143

3.2 Performances of hedge detection

Fig. 2 describes the learning curves of the co-training systems in hedge detection. Fig. 2(A) and Fig. 2(C) show the performance of the co-training system taking the test data as unlabeled data set (we call it the "test" system), in which we set the parameter $p = 80$. While Fig. 2(B) and Fig. 2(D) show the performance of the co-training system taking the free data as unlabeled data set (we call it the "free" system), in which we set the parameter $p = 160$ since the free unlabeled data is much larger than the test data.

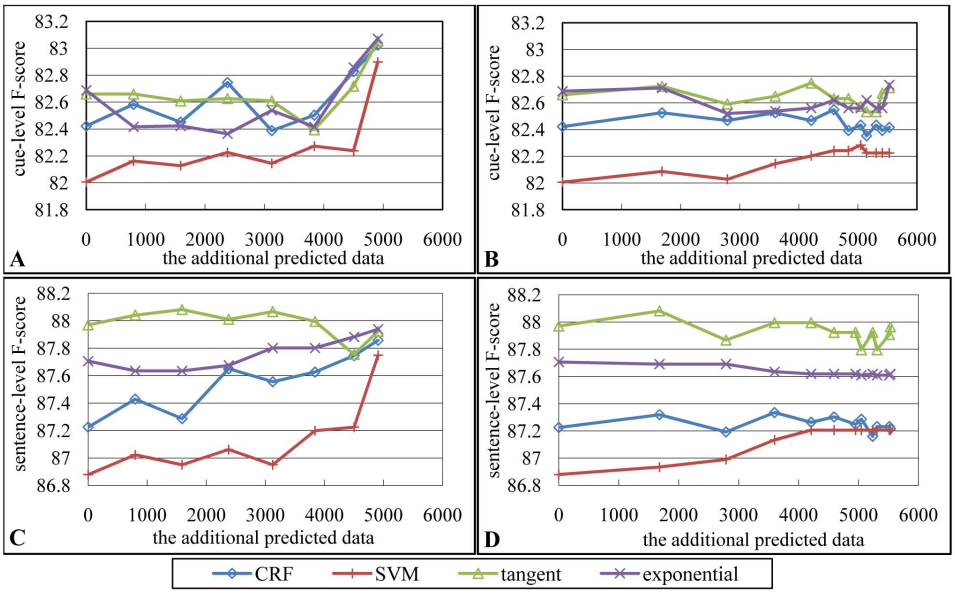

Fig. 2: Learning curves in hedge detection

In each co-training system, additional predicted data range from "0" (where "CRF" and "SVM" represents the baseline systems) to the maximum adding amount. At "0" additional training data, the sentence-level F-score of the baseline CRF and SVM systems are 87.22% and 86.88%. Without the co-training approach, combining the results of the two baseline systems alone can perform better than the baseline systems. During the bootstrapping process of co-training, the CRF and SVM systems could improve each other and become similar at the iteration. Furthermore, the performance of the two combined methods is obviously higher than the two component systems owing to their deep integration of the confidence values.

When comparing the performance of the "test" system with the "free" system, we can see that the co-training iteration helps more in the test co-training system than in the "free" system. In the "test" system, the CRF and the SVM systems are steadily improved with each iteration and reach an agreement on the test data finally. In Fig. 2(A), the performance of the system is increasing greatly towards the end of the test set. This suggests that the instances which are picked very late in the co-training process are the most useful ones, though they are most difficult to be predicted. In the "free" system, the co-training method seems to improve the performance a little. Though the sentence scale of the free data is up to 44569, the sentences that exceed the threshold and can be used as the training data is only 5532. This is perhaps because that the test data is quite different from the free data and difficult for the systems trained with the additional free data to predict.

In order to examine the effects of parameter p, we conduct experiments on the different systems

with different p values. In the "test" system, the sentence-level and cue-level achieve the best F-score with $p = 60$. In the "free" system, the sentence-level achieves the best F-score with $p = 80$ and $p = 120$, while the cue-level achieves the best F-score with $p = 240$. The best final results of the co-training systems in hedge detection are shown in Table 2, where "free+test" stands for the co-training system using the free and test data together as unlabeled data. From Table 2, we can see that the best final result of the "test" is better than that of the "free". For the sentence-level evaluation, the "free+test" gives the highest performance with the F-score of 88.12%. As for the cue-level evaluation, the maximum F-score of 83.11% is achieved by the "test".

Our co-training system is compared with three state-of-the-art systems on the sentence-level in Table 3. Tang et al. [5] is the top system of the CoNLL 2010 Shared Task in the hedge detection task; Zhou et al. [6] propose a voting-based ensemble system based on three machine learning algorithms; Velldal et al. [9] build a filter model on the hedge cue patterns. It is obvious that our system outperforms Tang's, Zhou's and Velldal's systems respectively by increases of 1.76%, 0.63% and 1.54% in F-score.

Table 2: The best final results of the different co-training systems in hedge detection

Systems	**Sentence-level**			**Cue-level**		
	Rec.(%)	**Prec.**(%)	**F-score**(%)	**Rec.**(%)	**Prec.**(%)	**F-score**(%)
free	87.22	88.79	88	79.56	86.23	82.76
test	**87.47**	88.59	88.03	**81.09**	85.24	**83.11**
free+test	86.84	**89.44**	**88.12**	79.47	**86.49**	82.83

Table 3: Performance comparison on sentence-level hedge detection

Systems	**Rec.**(%)	**Prec.**(%)	**F-score**(%)
Tang et al. [5]	85.03	87.72	86.36
Zhou et al. [6]	**88.69**	86.33	87.49
Velldal et al. [9]	85.32	87.87	86.58
OURS	86.84	**89.44**	**88.12**

3.3 Performances of hedge scope detection

We evaluate our hedge scope detection systems with cues predicted by the "test" system, which achieves the best F-score 83.11% as shown in Table 2. As a result of the co-training systems in hedge detection, a total number of 800 speculative sentences from the test data set and 1071 uncertain sentences from the free data set are extracted for co-training in hedge scope detection.

Fig. 3 presents the learning curves of the co-training approach with the annotated data increasing. Fig. 3(A) gives the performance of "test" co-training system while Fig. 3(B) gives the performance of "free" co-training system. We set the parameter $p = 25$ for the "test" and "free" systems. For both data sets the proposed co-training approach performs better than the baseline systems. Comparing the two figures, we can see that the co-training approach is more

helpful in the "test" system, which complies with Task 1. The essential reason we think is that the distribution of the test data is transferred to the labeled training data by the co-training process.

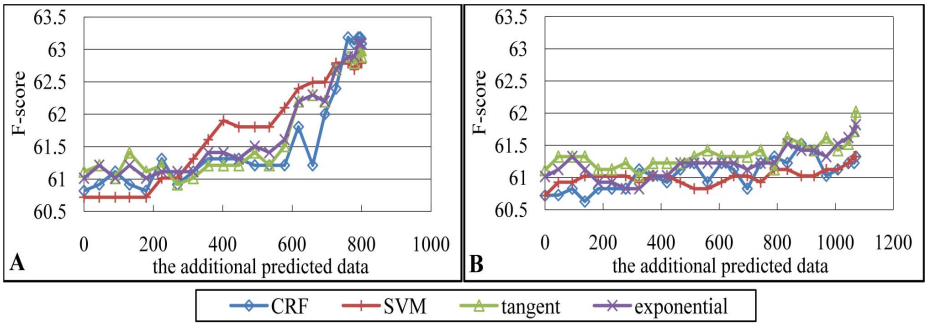

Fig. 3: Learning curves in hedge scope detection

In order to examine the effects of parameter p, we conduct experiments on the different systems with different p values. The hedge scope detection achieves best results with $p = 25$ in the "test" and "free" systems. Table 4 gives the best final iterative results of different p with the three kinds of unlabeled data in hedge scope detection, which surpass the baseline CRF and SVM systems. The best final F-score adding the two data sets (free+test) is unexpectedly lower than that of the "test" system, which is probably because the distribution of the free data is different to that of the test data and dominates the co-training performance owing to its large scale.

Table 4: The best final results of the different co-training systems in hedge scope detection

Systems	**Rec.**(%)	**Prec.**(%)	**F-score**(%)
free	60.31	63.83	62.02
test	**61.96**	**64.26**	**63.09**
free+test	61.28	63.87	62.55

In Table 5, the co-training system is compared with three state-of-the-art systems: Morante et al. [8] obtain the best result in hedge scope detection of CoNLL 2010 Shared Task; Zhou et al. [6] set up a voting-based ensemble system for hedge scope detection; Velldal et al. [9] combine a discriminative ranking function with manually crafted rules over the dependency and phrase parse tree of the sentences. Our system surpasses Morante's, Zhou's and Velldal's in F-score respectively by 5.77%, 2.22% and 3.68%. Besides, we also obtain 1071 high-quality cues in uncertain sentences from unlabeled free data.

Table 5: Performance comparison in hedge scope detection

Systems	**Rec.**(%)	**Prec.**(%)	**F-score**(%)
Morante et al. [8]	59.62	55.18	57.32
Zhou et al. [6]	**62.91**	58.95	60.87
Velldal et al. [9]	57.02	62.00	59.41
OURS	61.96	**64.26**	**63.09**

4 Conclusion

In this paper, we apply the co-training idea to the detection of hedge cues and their scope. Our research focuses on boosting the performance of hedge detection by leveraging two sets of unlabeled data-the free data set and the test data set. Experiments show that both the free data set and the test data set boost the performance of the basic systems. And adding predicted test data to the co-training system works better than adding free data. With the optimal setting, the co-training system achieves the best F-score of 88.12% in the sentence-level hedge detection and 63.09% in the hedge scope detection.

The semantic information of words plays an important role in detecting hedge cues and their scope. In future work, we will further extract the semantic knowledge of words from large-scale free data to improve the detection performance of hedge cues and their scope.

References

[1] G. Lakoff, Hedges: A study in meaning criteria and the logic of fuzzy concepts [J], Journal of Philosophical Logic, 2(4), 1973, pp. 458-508.

[2] R. Farkas, V. Vincze, G. Móra, et al., The CoNLL 2010 shared task: learning to detect hedges and their scope in natural language text [C], Proc. CoNLL Conf. 10, 2010, pp. 1-12.

[3] M. Light, X. Y. Qiu, and P. Srinivasan, The language of bioscience: facts, speculations, and statements in between [C], Proc. BioLink Wkshp. 04 at HLT/NAACL, 2004, pp. 17-24.

[4] R. Morante and W. Daelemans, Learning the scope of hedge cues in biomedical texts [C], Proc. Wkshp. on BioNLP, 2009, pp. 28-36.

[5] B. Z. Tang , X. L. Wang, X. Wang, et al., A cascade method for detecting hedges and their scope in natural language text [C], Proc. CoNLL Conf. 10, 2010, pp. 13-17.

[6] H. W. Zhou, X. Y. Li, D. G. Huang, et al., Voting-based ensemble clas-sifiers to detect hedges and their scopes in biomedical texts [J], IEICE Trans. Inf. Syst, E94-D(10), 2011, pp. 1989-1997.

[7] G. Szarvas, V. Vincze, R. Farkas, et al., Cross-genre and cross-domain detection of semantic uncertainty [J], ACL, 38(2), 2012, pp. 335-367.

[8] R. Morante, V. V. Asch, and W. Daelemans, Memory-based resolution of in-sentence scopes of hedge cues [C], Proc. CoNLL Conf. 10, 2010, pp. 40-47.

[9] E. Velldal, L. Vrelid, J. Read, et al., Speculation and negation: rules, rankers, and the role of syntax [J], ACL, 38(2), 2012, pp. 369-410.

[10] A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training [C], Proc. COLT 11th Annu. Conf., 1998, pp. 92-100.

[11] S. Abney, Bootstrapping [C], Proc. ACL 40th Annu. Meeting, 2002, pp. 360-367.

[12] K. L. Li, J. Zhang, H. Y. Xu, et al., A semi-supervised extreme learning machine method based on co-training [J], Journal of Computational Information Systems, 9(1), 2013, pp. 207-214.

[13] T. Joachims, Learning to classify text using Support Vector Machines: methods, theory, and algorithms [J], Computational Linguistics, 29(4), 2002, pp. 655-661.

[14] J. Lafferty, A. McCallum, and F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data [C], Proc. ICML Conf. 01, 2001, pp. 282-289.