Hedge Detection and Scope Finding by Sequence Labeling with Normalized Feature Selection*

Shaodian Zhang[12], **Hai Zhao**[123]†, **Guodong Zhou**[3] and **Bao-Liang Lu**[12]
[1]Center for Brain-Like Computing and Machine Intelligence
Dept of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University, 800 Dong Chuan Rd., Shanghai 200240, China
[3]School of Computer Science and Technology, Soochow University, Suzhou, China 215006
zhangsd.sjtu@gmail.com, {zhaohai,blu}@cs.sjtu.edu.cn,gdzhou@suda.edu.cn

Abstract

This paper presents a system which adopts a standard sequence labeling technique for hedge detection and scope finding. For hedge detection, we formulate it as a hedge labeling problem, while for hedge scope finding, we use a two-step labeling strategy, one for hedge labeling and the other for scope finding. In particular, various kinds of syntactic dependencies are systemically exploited and effectively integrated using a large-scale normalized feature selection method. Evaluation on the CoNLL-2010 shared task shows that our system achieves stable and competitive results for all the closed tasks. Furthermore, post-deadline experiments show that the performance can be much further improved using a sufficient feature selection.

1 Introduction

Hedges are linguistic devices representing speculative parts of articles. Previous works such as (Hyland, 1996; Marco and Mercer, 2004; Light et al., 2004; Thompson et al., 2008) present research on hedge mainly as a linguistic phenomenon. Meanwhile, detecting hedges and their scopes automatically are increasingly important tasks in natural language processing and information extraction, especially in biomedical community. The shared task of CoNLL-2010 described in (Farkas et al., 2010) aims at detecting hedges (task 1) and finding their scopes (task 2) for the literature from BioScope corpus(Szarvas et al., 2008) and Wikipedia. This paper describes a system adopting sequence labeling which performs competitive in the official evaluation, as well as further test. In addition, a large-scale feature selection procedure is applied in training and development. Considering that BioScope corpus is annotated by two independent linguists according to a formal guideline (Szarvas, 2008), while Wikipedia weasels are tagged by netizens who are diverse in background and various in evaluation criterion, it is needed to handle them separately. Our system selects features for Wikipedia and BioScope corpus independently and evaluate them respectively, leading to fine performances for all of them.

The rest of the paper is organized as follows. The next section presents the technical details of our system of hedge detection and scope finding. Section 3 gives information of features. Section 4 shows the evaluation results, including official results and further ones after official outputs collection. Section 5 concludes the paper.

2 Methods

Basically, the tasks are formulated as sequence labeling in our approach. The available label set differs between task 1 and 2. In addition, it is needed to introduce an indicator in order to find scopes for the multi-hedge sentences properly.

2.1 Hedge detection

The valid label set of task 1, hedge detection, contains only two labels: "Hedge" and "_", which represent that a word is in a hedge cue or not respectively. Since results of hedge detection in this shared task are evaluated at sentence level, a sentence will be classified as "uncertain" in the post-process if it has one or more words labeled "Hedge" in it and otherwise "certain".

* This work is partially supported by the National Natural Science Foundation of China (Grants 60903119, 60773090, 90820018 and 90920004), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No.2008AA02Z315).

† corresponding author

2.2 Scope finding

The second task is divided into two steps in our system. The first step is quite the same as what the system does in task 1: labeling the words as in hedge cues or not. Then the scope of each hedge will be labeled by taking advantage of the result of the first step. A scope can be denoted by a beginning word and an ending word to represent the first and the last element. In scope finding the available label set contains "Begin", "End", "Middle" and "_", representing the first and last word in the scope, in-scope and out-of-scope. As an example, a sentence with hedge cue and scope labeling is given in Table 1. Hedge cue "indicating" with its scope from "indicating" itself to "transcription" are labeled. While evaluating outputs, only "Begin"s and "End"s will be taken into consideration and be treated as the head and tail tokens of the scopes of specific hedge cues.

1	Furthermore	...	_	_
2	,	...	_	_
3	inhibition	...	_	_
4	can	...	_	_
5	be	...	_	_
6	blocked	...	_	_
7	by	...	_	_
8	actinomycin	...	_	_
9	D	...	_	_
10	,	...	_	_
11	indicating	...	Hedge	Begin
12	a	...	_	Middle
13	requirement	...	_	Middle
14	for	...	_	Middle
15	de	...	_	Middle
16	novo	...	_	Middle
17	transcription	...	_	End
18	.	...	_	_

Table 1: A sentence with hedge cue and scope labeling

It seems that the best labeling result of task 1 can be used directly to be the proper intermediate representation of task 2. However, the complexity of scope finding for multi-hedge sentences forces us to modify the intermediate result of task 2 for the sake of handling the sentences with more than one hedge cue correctly. Besides, since task 1 is a sentence classification task essentially, while the goal of the first step of task 2 is to label the words as accurately as possible, it is easy to find that the optimal labeling results of task 1 may not be optimal to be the intermediate representations for task 2. This problem can be solved if sentence-level hedge detection and intermediate representation finding are treated as two separate tasks with independent feature selection procedures. The details of feature selection will be given in section 3.

2.3 Scope finding for multi-hedge cases

Sentences with more than one hedge cue is quite common in both datasets of BioScope corpus and Wikipedia. By counting hedges in every sentences, we find that about one fourth of the sentences with hedges have more than one hedge cue in all three data sources (Table 2). In (Morante and Daelemans, 2009), three classifiers predict whether each token is Begin, End or None and a postprocessing is needed to associate Begins and Ends with their corresponding hedge cues. In our approach, in order to decrease ambiguous or illegal outputs e.g. inequivalent numbers of Begins and Ends, a pair of Begin and End without their corresponding hedge cue between them, etc, sentences with more than one hedge cue will be preprocessed by making copies as many as the number of hedges and be handled separately.

The sentence which is selected as a sample has two hedge cues: "suggesting" and "may", so our system preprocesses the sentence into two single-hedge ones, which is illustrated in table 3. Now it comes to the problem of finding scope for single-hedge sentence. The two copies are labeled separately, getting one scope from "suggesting" to "mitogenesis" for the hedge cue "suggesting" and the other from "IFN-alpha" to "mitogenesis" for "may". Merging the two results will give the final scope resolution of the sentence.

1	IFN-alpha	_
2	also	_
3	sensitized	_
4	T	_
5	cells	_
6	to	_
7	IL-2-induced	_
8	proliferation	_
9	,	_
10	further	_
11	suggesting	Hedge
12	that	_
13	IFN-alpha	_
14	may	_
15	be	_
16	involved	_
17	in	_
18	the	_
19	regulation	_
20	of	_
21	T-cell	_
22	mitogenesis	_
23	.	_

1	IFN-alpha	_
2	also	_
3	sensitized	_
4	T	_
5	cells	_
6	to	_
7	IL-2-induced	_
8	proliferation	_
9	,	_
10	further	_
11	suggesting	_
12	that	_
13	IFN-alpha	_
14	may	Hedge
15	be	_
16	involved	_
17	in	_
18	the	_
19	regulation	_
20	of	_
21	T-cell	_
22	mitogenesis	_
23	.	_

Table 3: An example of 2-hedge sentence before scope finding

Dataset	# Sentence	# No-hedge	ratio	# One-hedge	ratio	# Multi-hedge	ratio
Biomedical Abstracts	11871	9770	82.3%	1603	13.5%	498	4.2%
Biomedical Fulltexts	2670	2151	80.6%	385	14.4%	134	5.0%
Wikipedia	11111	8627	77.6%	1936	17.4%	548	4.9%

Table 2: Statistics of hedge amount

However, compared with matching Begins and Ends in postprocessing given by (Morante and Daelemans, 2009), the above method gives rise to out of control of projections of the scopes, i.e. scopes of hedges may partially overlap after copies are merged. Since scopes should be intact constituents of sentences, namely, subtrees in syntax tree which never partly overlap with each other, results like this are linguistically illegal and should be discarded. We solve this problem by introducing an instructional feature called "Indicator". For sentences with more than one hedge cue, namely more than one copy while finding scopes, words inside the union of existing (labeled) scopes will be tagged as "Indicator" in unhandled copies before every labeling. For example, after finding scope for the first copy in Table 3 and words from "suggesting" to "mitogenesis" are put in the scope of cue "suggesting", these words should be tagged "Indicator" in the second copy, whose result is illustrated in Table 4.If not in a scope, any word is tagged "_" as the indicator. The "Indicator"s tagging from "suggesting" to "mitogenesis" in Table 4 mean that no other than the situations of a) "Begin" is after or at "suggesting" and "End" is before or at "mitogenesis" b) Both "Begin" and "End" are before "suggesting" c) Both next "Begin" and next "End" are after "mitogenesis" can be accepted. In other words, new labeling should keep the projections of scopes in the result. Although it is only an instructional indicator and doesn't have any coerciveness, the evaluation result of experiment shows it effective.

3 Feature selection

Since hedge and scope finding are quite novel tasks and it is not easy to determine the effective features by experience, an greedy feature selection is conducted. As it mentioned in section 2, our system divides scope finding into two sub-tasks:

a) Hedge cue labeling

b) Scope labeling

The first one is the same as hedge detection task in strategy, but quite distinct in target of feature

1	IFN-alpha	...	_	_	_
2	also	...	_	_	_
3	sensitized	...	_	_	_
4	T	...	_	_	_
5	cells	...	_	_	_
6	to	...	_	_	_
7	IL-2-induced	...	_	_	_
8	proliferation	...	_	_	_
9	,	...	_	_	_
10	further	...	_	_	_
11	suggesting	...	Indicator	_	_
12	that	...	Indicator	_	_
13	IFN-alpha	...	Indicator	_	Begin
14	may	...	Indicator	Hedge	Middle
15	be	...	Indicator	_	Middle
16	involved	...	Indicator	_	Middle
17	in	...	Indicator	_	Middle
18	the	...	Indicator	_	Middle
19	regulation	...	Indicator	_	Middle
20	of	...	Indicator	_	Middle
21	T-cell	...	Indicator	_	Middle
22	mitogenesis	...	Indicator	_	End
23	.	...	_	_	_

Table 4: Scope resolution with instructional feature: "Indicator"

set, because hedge detection is a task of sentence classification while the first step of scope finding aims at high accuracy of labeling hedge cues. Therefore, three independent procedures of feature selection are conducted for BioScope corpus dataset. As Wikipedia is not involved in the task of scope finding, it only needs one final feature set.

About 200 feature templates are initially considered for each task. We mainly borrow ideas and are enlightened by following sources while initializing feature template sets:

a) Previous papers on hedge detection and scope finding (Light et al., 2004; Medlock, 2008; Medlock and Briscoe, 2008; Kilicoglu and Bergler, 2008; Szarvas, 2008; Ganter and Strube, 2009; Morante and Daelemans, 2009);

b) Related works such as named entity recognition (Collins, 1999) and text chunking (Zhang et al., 2001);

c) Some literature on dependency parsing (Nivre and Scholz, 2004; McDonald et al., 2005; Nivre, 2009; Zhao et al., 2009c; Zhao et al., 2009a);

3.1 Notations of Feature Template

A large amount of advanced syntactic features including syntactic connections, paths, families and their concatenations are introduced. Many of these features come from dependency parsing, which aims at building syntactic tree expressed by dependencies between words. More details about dependency parsing are given in (Nivre and Scholz, 2004; McDonald et al., 2005). The parser in (Zhao et al., 2009a) is used to construct dependency structures in our system, and some of the notations in this paper adopt those presented in (Zhao et al., 2009c). Feature templates are from various combinations or integrations of the following basic elements.

Word Property. This part of features include word form (*form*), lemma (*lemma*), part-of-speech tag (*pos*), syntactic dependency (*dp*) , syntactic dependency label (*dprel*).

Syntactic Connection. This includes syntactic head (*h*), left(right) farthest(nearest) child (*lm, ln, rm* and *rn*) and high(low) support verb, noun or preposition. Here we specify the last one as an example, support verb(noun/preposition). From a given word to the syntactic root along the syntactic tree, the first verb/noun/preposition that is met is called its low support verb/noun/preposition, and the nearest one to the root(farthest to the given word) is called as its high support verb/noun/preposition. The concept of support verb was broadly used (Toutanova et al., 2005; Xue, 2006; Jiang and Ng, 2006), and it is extended to nouns and prepositions in (Zhao et al., 2009b). In addition, a slightly modified syntactic head, *pphead*, is introduced, it returns the left most sibling of a given word if the word is headed by a preposition, otherwise it returns the original head.

Path. There are two basic types of path. One is the linear path (*linePath*) in the sequence, the other is the path in the syntactic parsing tree (*dpPath*). For example, $m{:}n|$*dpPath* represents the dependency path from word *m* to *n*. Assuming that the two paths from *m* and *n* to the root are p_m and p_n, $m{:}n|$*dpPathShare*, $m{:}n|$*dpPathPred* and $m{:}n|$*dpPathArgu* represent the common part of p_m and p_n, part of p_m which doesn't belong to p_n and part of p_n which doesn't belong to p_m, respectively.

Family. A children set includes all syntactic children(*children*) are used in the template notations.

Concatenation of Elements. For all collected elements according to *dpPath*, *children* and so on, we use three strategies to concatenate all those strings to produce the feature value. The first is *seq*, which concatenates all collected strings without doing anything. The second is *bag*, which removes all duplicated strings and sort the rest. The third is *noDup*, which removes all duplicated neighbored strings.

Hedge Cue Dictionary and Scope Indicator. Hedge cues in the training set are collected and put in a dictionary. Whether a word in the training or testing set is in the dictionary (*dic*) is introduced into feature templates. As the evaluation is non-open, we don't put in any additional hedge cues from other resources. An indicator (*indicator*) is given for multi-hedge scope finding, as specified in section 2.At last, in feature set for scope labeling, *hedge* represents that the word is in a hedge cue.

At last, we take x as current token to be labeled, and x_m to denote neighbor words. $m > 0$ represents that it is a word goes m_{th} after current word and $m < 0$ for word $-m_{th}$ before current word.

3.2 Feature template sets for each task

As optimal feature template subsets cannot be expected to be extracted from so large sets by hand, greedy feature selections according to (Zhao et al., 2009b) are applied. The normalized feature selection has been proved to be effective in quite a lot of NLP tasks and can often successfully select an optimal or very close to optimal feature set from a large-scale superset. Although usually it needs 3 to 4 loops denoted by "While" in the Algorithm 1 of (Zhao et al., 2009b) to get the best template set, we only complete one before official outputs collection because of time limitation, which to a large extent hinders the performance of the system.

Three template sets are selected for BioScope corpus. One with the highest accuracy for sentence-level hedge detection (Set B), one with the best performance for word-level hedge cue labeling (Set H) and another one with the maximal F-score for scope finding (Set S). In addition, one set is discovered for sentence-level hedge detection of Wikipedia (Set W)[1] . Table 5[2] lists some selected feature templates which are basic word or

[1] *num* in the set of Wikipedia represents the sequential number of word in the sentence

[2] Contact the authors to get the full feature lists, as well as entire optimized sets in post-deadline experiment

hedging properties for the three sets of BioScope corpus and Wikipedia. From the table we can see it is clear that the combinations of lemma, POS and word form of words in context, which are usually basic and common elements in NLP, are also effective for hedge detection. And as we expected, the feature that represents whether the word is in the hedge list or not is very useful especially in hedge cue finding, indicating that methods based on a hedge cue lists (Light et al., 2004) or keyword selection (Szarvas, 2008) are quite significant way to accomplish such tasks.

-	$x.lemma + x_1.lemma + x_{-1}.lemma + x.dic + x_1.dic + x_{-1}.dic$
-	$x.lemma + x_1.pos + x_{-1}.pos + x.pos + x_1.lemma + x_{-1}.lemma$
-	$x.form$
Set B	$x.pos + x_1.pos + x_{-1}.pos + x_2.pos + x_{-2}.pos$
-	$x.dic + x_1.dic + x_{-1}.dic$
-	$x_1.pos$
-	$x.dic + x_1.dic + x_{-1}.dic + x_2.dic + x_{-2}.dic$
-	$x.pos + x_{-1}.pos$
-	$x.dic$
Set H	$x.dic + x.lemma + x.pos + x.form$
-	$x.pos + x_1.pos + x_{-1}.pos + x_2.pos + x_{-2}.pos$
-	$x_{-2}.form + x_{-2}.lemma$
-	$x_{-1}.form + x.form$
-	$x.dic + x_1.dic + x_{-1}.dic$
-	$x.dic + x_1.dic + x_{-1}.dic + x_2.dic + x_{-2}.dic + x_3.dic + x_{-3}.dic$
-	$x.indicator$
-	$x.hedge + x_1.hedge + x_{-1}.hedge$
Set S	$x.lemma + x_1.pos + x_{-1}.pos + x.pos + x_1.lemma + x_{-1}.lemma$
-	$x.pos + x.hedge + x.dp + x.dprel$
-	$x_1.pos$
-	$x.pos + x_1.pos + x_{-1}.pos + x_2.pos + x_{-2}.pos$
-	textit$x$.lemma + x_1.lemma + x_{-1}.lemma + x.dic
-	$+ x_1.dic + x_{-1}.dic$
-	$x.lemma + x_1.lemma + x_{-1}.lemma + x_2.lemma + x_{-2}.lemma + x.dic + x_1.dic + x_{-1}.dic + x_2.dic + x_{-2}.dic$
-	$x.lemma + x_1.lemma$
Set W	$x.hedge + x_1.hedge + x_{-1}.hedge + x_2.hedge + x_{-2}.hedge + x_3.hedge + x_{-3}.hedge$
-	$x.pos + x_1.pos + x_{-1}.pos + x_2.pos + x_{-2}.pos + x.dic + x_1.dic + x_{-1}.dic + x_2.dic + x_{-2}.dic$
-	$x.pos + x.dic$
-	$x.num + x.dic$

Table 5: Selected feature template sets

Some a little complicated syntactic features based on dependencies are systemically exploited as features for tasks. Table 6 enumerates some of the syntactic features which proves to be highly effective. We noticed that *lowSupportNoun*, *highSupportNoun* and features derived from *dpPath* is notably useful. It can be explained by the awareness that hedge labeling and scope finding are to process literatures in the level of semantics where syntactic features are often helpful.

-	*x.lowSupportNoun:x | dpPathArgu.dprel.seq*
-	*x.lowSupportNoun:x|dpPathArgu.dprel.seq + x.lowSupportProp:x|dpPathArgu.dprel.seq*
-	*x.lowSupoortNoun.pos*
-	*x.pos + x.children.dprel.bag*
-	*x.rm.dprel + x.form*
Set B	*x.pphead.lemma*
-	*x.form + x.children.dprel.bag*
-	*x.lowSupportNoun:x—dpTreeRelation*
-	*x.lowSupportProp.lemma*
-	*x.form + x.children.dprel.noDup*
-	*x.highSupportNoun:x|dpTreeRelation + x.form*
-	*x.lowSupportVerb.form*
-	*x.lowSupportProp:x|dpPathShared.dprel.seq*
-	*x.lowSupportProp:x|dpPathShared.pos.seq*
-	*x.highSupportNoun.pos*
-	*x.highSupportNoun:x|dpTreeRelation*
-	*x.highSupportNoun:x|dpPathArgu.dprel.seq*
Set H	*+ x.highSupportProp:x|dpPathArgu.dprel.seq*
-	*xlowSupportProp.lemma*
-	*x.rm.dprel*
-	*x.lm.form*
-	*x.lemma + x.pphead.form*
-	*x.lowSupportVerb.form*
-	*x.rm.lemma + x.rm.form*
-	*x.children.dprel.noDup*
-	*x.children.dprel.bag*
-	*x.highSupportNoun:x|dpTreeRelation*
-	*x.lemma + x.pphead.form*
Set S	*x.highSupportNoun:x|dpTreeRelation + x.form*
-	*x.lowSupportVerb.form*
-	*x.lowSupportVerb.lemma*
-	*x.h.children.dprel.bag*
-	*x.highSupportVerb.form*
-	*x.lm.form*
-	*x.lemma + x.pphead.form*
-	*x.lm.dprel + x.pos*
-	*x.lowSupportProp:x|dpPathPred.dprel.seq*
-	*x.pphead.lemma*
Set W	*x.rm.lemma*
-	*x.lowSupportProp:x|dpTreeRelation*
-	*x.lowSupportVerb:x|dpPathPred.dprel.seq*
-	*x.lowSupportVerb:x|dpPathPred.pos.seq*
-	*x.lowSupportVerb:x|dpPathShared.pos.seq*
-	*x.lowSupportProp:x|dpPathShared.pos.seq*
-	*x.lowSupportProp.form*

Table 6: Syntactic features

We continue our feature selection procedures for BioScope corpus after official outputs collection and obtain feature template sets that bring better performance. Table 7 gives some of the features in the optimized sets for BioScope corpus resolution. One difference between the new sets and the old ones is the former contain more syntactic elements, indicating that exploiting syntactic feature is a correct choice. Another difference is the new sets assemble more information of words

-	x_{-1}.*lemma*
-	x.*dic* + x_1.*dic* + x_{-1}.*dic* + x_2.*dic* + x_{-2}.*dic* + x_3.*dic* + x_{-3}.*dic*
-	x_{-1}.*pos* + x_1.*pos*
Set H	x.*rm.lemma*
-	x.*rm.dprel*
-	x.*lm.dprel* + x.*pos*
-	x.*lowSupportNoun:x* | *dpPathArgu.dprel.seq*
-	x.*lowSupportNoun:x*|*dpPathArgu.dprel.seq* + x.*lowSupportProp:x*|*dpPathArgu.dprel.seq*
-	x_{-1}.*lemma*
-	x.*lemma* + x_1.*lemma* + x_{-1}.*lemma* + x.*dic* + x_1.*dic* + x_{-1}.*dic*
-	x.*form* + x.*lemma* + x.*pos* + x.*dic*
Set B	x_{-2}.*form* + x_{-1}.*form*
-	x.*highSupportNoun:x*|*dpTreeRelation*
-	x.*highSupportNoun:x*|*dpPathArgu.dprel.seq*
-	x.*lowSupportProp:x*|*dpPathShared.dprel.seq*
-	x_{-1}.*lm.form*
-	x_1.*form*
-	x.*pos* + x.*dic*
-	x.*hedge* + x_1.*hedge* + x_{-1}.*hedge*
-	x.*pos* + x_1.*pos* + x_{-1}.*pos* + x_2.*pos* + x_{-2}.*pos*
Set S	x.*children.dprel.bag*
-	x.*lemma* + x.*pphead.form*
-	x.*highSupportVerb.form*
-	x.*highSupportNoun:x*|*dpTreeRelation* + *x.form*
-	x.*lowSupportNoun:x*|*dpTreeRelation* + *x.form*

Table 7: Selected improved feature template sets for BioScope corpus

before or after the current word, especially words linearly far away but close in syntax tree. Appearance of combination of these two factors such as x_{-1}.*lm.form* seems to provide an evidence of the insufficiency training and development of our system submitted to some extent.

4 Evaluation results

Two tracks (closed and open challenges) are provided for CoNLL-2010 shared task. We participated in the closed challenge, select features based on the in-domain data and evaluated our system on the in-domain and cross-domain evaluation set. All the experiments are implemented and run by Maximum Entropy Markov Models (McCallum, 2000).

4.1 Official results

The official results for tasks are in Table 8, in which three in-domain tests and cue matching result for biomedical texts are listed. For the first task for BioCorpus, our system gives F-score 0.8363 in in-domain test and for Wikipedia we give F-score 0.5618 in closed evaluation. For the second task, our system gives results in closed and open test, with F-score 0.4425 and 0.4441 respectively.

We compare the F-score of our system with the best in the final result in Table 9. We rank pretty high in Wikipedia hedge detection, while other three are quite steady but not prominent. This is mainly due to two reasons:

1. Feature selection procedures are not perfectly conducted.

2. Abstracts and fulltexts in BioScope are mixed to be the training set, which proves quite inappropriate when the evaluation set contains only fulltext literature, since abstract and fulltext are quite different in terms of hedging.

Dataset		F-score	Best
BioScope	Task1-closed	0.8363	0.8636
	Task2-closed	0.4425	0.5732
	Cue-matching	0.7853	0.8134
Wikipedia	Task1-closed	0.5618	0.6017

Table 9: Comparing results with the best

4.2 Further results

Intact feature selection procedures for BioScope corpus are conducted after official outputs collections. The results of evaluation with completely selected features compared with the incomplete one are given in Table 7. The system performs a higher score on evaluation data (Table 10), which is more competitive in both tasks on BioScope corpus. The improvement for task 2 is significant, but the increase of performance of hedge cue detection is less remarkable. We believe that a larger fulltext training set and a more considerate training plan will help us to do better job in the future work.

Dataset		Complete	Incomplete
BioScope	Task1-closed	0.8522	0.8363
	Task2-closed	**0.5151**	0.4425
	Cue-matching	0.7990	0.7853

Table 10: Comparing improved outputs with the best

5 Conclusion

We describe the system that uses sequence labeling with normalized feature selection and rich features to detect hedges and find scopes for hedge cues. Syntactic features which are derived from dependencies are exploited, which prove to be

Dataset		TP	FP	FN	precision	recall	F-score
BioScope	Task1-closed	669	141	121	0.8259	0.8468	0.8363
	Task2-closed	441	519	592	0.4594	0.4269	0.4425
	Cue-matching	788	172	259	0.8208	0.7526	0.7853
Wikipedia	Task1-closed	991	303	1243	0.7658	0.4436	0.5618

Table 8: Official results of our submission for in-domain tasks

quite favorable. The evaluation results show that our system is steady in performance and does pretty good hedging and scope finding in both BioScope corpus and Wikipedia, especially when the feature selection procedure is carefully and totally conducted. The results suggest that sequence labeling and a feature-oriented method are effective in such NLP tasks.

References

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Richard Farkas, Veronika Vincze, Gyorgy Mora, Janos Csirik, and Gyorgy Szarvas. 2010. The conll 2010 shared task: Learning to detect hedges and their scope in natural language text. In *In Proceedings of the CoNLL2010 Shared Task*.

Viola Ganter and Michael Strube. 2009. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 173–176, Suntec, Singapore, 4, August.

Ken Hyland. 1996. Writing without conviction: Hedging in science research articles. *Applied Linguistics*, 17:433–54.

Zheng Ping Jiang and Hwee Tou Ng. 2006. Semantic role labeling of nombank: A maximum entropy approach. In *Proceedings of the EMNLP-2006*, pages 138–145, Sydney, Australia.

Halil Kilicoglu and Sabine Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9.

Marc Light, Xin Ying Qiu, and Padimini Srinivasan. 2004. The language of biosciences: Facts, speculations, and statements in between. In *In Proc. of the BioLINK 2004*, pages 17–24.

Chrysanne Di Marco and Robert E. Mercer. 2004. Hedging in scientific articles as a means of classifying citations. In *In Working Notes of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 50–54.

Andrew McCallum. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proc. ICML 2000*, pages 591–598, Stanford, California.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT/EMNLP 05*, pages 523–530, Vancouver, Canada, October.

Ben Medlock and Ted Briscoe. 2008. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of 45th Annual Meeting of the ACL*, pages 992–999, Prague, Czech Republic, June.

Ben Medlock. 2008. Exploring hedge identification in biomedical literature. *Journal of Biomedical Informatics*, 41:636–654.

Roser Morante and Walter Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the Workshop on BioNLP*, pages 28–36, Boulder, Colorado, June.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of english text. In *Proceedings of (COLING-2004)*, pages 64–70, Geneva, Switzerland, August 23rd-27th.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of ACL-IJCNLP 2009*, pages 351–359, Suntec, Singapore, 2-7 August.

Gyorgy Szarvas, Veronika Vincze, Richard Farkas, and Janos Csirik. 2008. The bioscope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of BioNLP 2008*, pages 38–45, Columbus, Ohio, USA, June.

Gyorgy Szarvas. 2008. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of ACL-08*, pages 281–289, Columbus, Ohio, USA, June.

Paul Thompson, Giulia Venturi, John McNaught, Simonetta Montemagni, and Sophia Ananiadou. 2008. Categorising modality in biomedical texts. In *In Proc. of the LREC 2008 Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pages 27–34, Marrakech.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of ACL-2005*, pages 589–596, Ann Arbor, USA.

Nianwen Xue. 2006. Semantic role labeling of nominalized predicates in chinese. In *Proceedings of the Human Language Technology Conference of the NAACL (NAACL-2006)*, pages 431–438, New York City, USA, June.

Tong Zhang, Fred Damerau, and David Johnson. 2001. Text chunking using regularized winnow. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 539–546, Toulouse, France.

Hai Zhao, Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009a. Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies. In *Proceedings of (CoNLL-2009), June 4-5*, Boulder, Colorado, USA.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009b. Semantic dependency parsing of nombank and propbank: An efficient integrated approach via a large-scale feature selection. In *Proceedings of EMNLP-2009*, pages 30–39, Singapore.

Hai Zhao, Wenliang Chen, Chunyu Kit, and Guodong Zhou. 2009c. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of (CoNLL-2009), June 4-5*, Boulder, Colorado, USA.