Exploiting Rich Syntactic Features for Hedge Detection and Scope Finding*

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, China 200240
[3]School of Computer Science and Technology, Soochow University, Suzhou, China 215006
zhangsd.sjtu@gmail.com, zhaohai@cs.sjtu.edu.cn
gdzhou@suda.edu.cn, blu@cs.sjtu.edu.cn

Abstract

Hedge detection and scope finding are increasingly important tasks in information extraction, especially in the biomedical natural language processing community. In this paper, a novel approach detecting hedge cues and their scopes by sequence labeling is explored. It should be emphasized that syntactic dependencies are systematically exploited and effectively integrated by a large-scale feature selection procedure. Experimental results demonstrate that our method outperforms previous works, and the selected syntactic features effectively promote the performances in both tasks of hedge detection and scope finding.

1 Introduction

In the NLP community, in order to mark off uncertain elements from factual information in specific literature, linguistic devices such as hedges, which indicate that authors do not authenticate their propositions or statements, have to be identified by detecting their hedge cues and linguistic scopes. *Hedge cues*, generally some keywords or phrases, are directly responsible for the uncertain and speculative nature. They reduce the credibility of some neighboring words which are defined as in the *scope* of the hedge cue. The speculative parts can be discarded or presented with lower confidence by specific applications to eliminate negative influences.

1.1 Hedge: Speculative Language

Hedge is used to represent a proposition or statement which is speculative, tentative or uncertain. Hedges can be found in almost all kinds of scientific literature, and they are particularly common in articles of experimental natural sciences (Medlock, 2008), because making hypothesis based on experimental results is an important part in this sort of literature. According to Light et al. (2004), the lack of definite belief is also reflected in the way scientists discuss their works. The following is an example which shows a subtle difference between statements with and without a hedge:

a) *Some patients with this multifactorial disease **may** have a **putative** systemic disorder at this level.*

b) *Some patients with this multifactorial disease have a systemic disorder at this level.*

The first sentence contains hedges, whose cues are *may* and *putative*, leading to a reduced reliability of the proposition. In this example, scopes of the two hedge cues are *"may have a putative systemic disorder at this level"* and *"putative systemic disorder"* for *may* and *putative*, respectively.

1.2 Hedge Detection and Scope Finding

Research papers addressing the detection of hedge devices in biomedical texts (Light et al., 2004; Medlock, 2008; Medlock and Briscoe, 2008; Kilicoglu

* 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).

and Bergler, 2008; Szarvas, 2008; Morante and Daelemans, 2009) and Wikipedia literature (Ganter and Strube, 2009) reveal the increasing interests in automatic hedge detection and scope finding. This paper investigates the two tasks using rich syntactic features and proposes effective solutions by sequence labeling. We regard syntactic features as useful structural information for such tasks since hedge is a context-sensitive linguistic phenomenon and hedge detection is basically a kind of semantic analysis rather than simply a keyword matching. On the other hand, scope finding is treated as a task mostly at syntactic level, which can also benefit from the parsing results.

The rest of the paper is organized as follows. The next section reviews previous works. Section 3 presents the technical details of our formulations. Section 4 describes feature template sets for tasks. Section 5 discusses how to label scopes for multi-cue sentences exploiting additional method and feature. Section 6 presents details of our experiments. Finally, section 7 concludes the paper.

2 Related Work

The linguistic concept of hedge is firstly brought forward by Lakoff (1972) and defined as *words whose meaning implicitly involves fuzziness*. Palmer (1986) distinguishes three types of modality: epistemic, deontic and dynamic. Among them, the epistemic modality, which expresses the speaker or author's degree of commitment to the truth of a proposition, is much related to hedge. Modality of events is further investigated by Sauri et al. (2006). They also identify the scope of modality in natural language text and propose a solution for its automatic identification.

Hyland (1996) is the first to investigate hedges in scientific research articles. A corpus is manually analyzed. Non-factive statements are categorized into content-oriented hedges and reader-oriented ones. Hyland claims that *hedges are abundant in science and play a critical role in academic writing more generally*, which gets to be a great motivation for subsequent research. Then, expression of levels of belief is discussed by Light et al. (2004). They also explore the speculative languages in biomedical abstracts, present an annotation guideline for manual hedge analysis and firstly use automatic classifier to select speculative sentences from literature. In their method, a small-scale hedge cue list is used to decide whether a sentence contains hedge or not. On the other hand, Light et al. (2004) do not manage to characterize the distinction between high and low speculation, which is later discussed by Kilicoglu et al. (2008) via assigning weights to hedge cues. Besides, some linguistic tools, e.g. syntactic patterns, are introduced into their system to decide hedges. Meanwhile, exploring on keyword list inherited from (Light et al., 2004) helps Thompson et al. (2008) to annotate 202 biomedical abstracts.

Most of the explorations focus on sentence-level hedge detection and formulate the problem as a sentence classification task by keyword extraction. Medlock and Briscoe (2008) define hedge classification as a weakly supervised machine learning task. Then Medlock (2008) uses richer features such as part-of-speech and lemma to strengthen the system. He also proposes that part-of-speech does not lead to an increase in performance while lemma is quite effective. Szarvas et al. (2008) formulate the problem with a weakly supervised selection of keywords. Ganter and Strube (2009) turns to learn hedges for Wikipedia, exploiting weasels [1] provided online and shallow linguistic features.

A corpus of biomedical texts, BioScope, which includes hedge cue and scope annotations for biomedical papers and clinical texts, is given by Vincze et al. (2008). It contains corpora on which previous works train and evaluate such as the Hedge Classification Corpus used by Medlock and Briscoe (2008). The corpus is composed of three parts: paper abstracts, full scientific articles and clinical texts. Based on the BioScope corpus, Morante and Daelemans (2009) formulate hedge detection and scope finding using one classifier for the first task and three for the second.

3 Formulations

Basically, hedge detection and scope finding are formulated as sequence labeling in our methods. Different label representations are adopted for the two tasks.

[1] http://en.wikipedia.org/wiki/Weasel_word

3.1 Hedge Detection

Two tags, "I" and "O", are used for representing in and outside a hedge cue in hedge detection:

- **I** Current token is in a hedge cue
- **O** Current token is outside a hedge cue

These representations permits word-level investigation into hedges by labeling every token in the sentence.

3.2 Scope Finding

The representations in scope labeling are similar to the IOB notations that have been applied in chunking task (Ramshaw and Marcus, 1995). Available label set contains "B", "E", "I" and "O":

- **B** Current token is the first one in the scope
- **E** Current token is the last one in the scope
- **I** Current token is inside the scope
- **O** Current token is outside the scope

Tokens labeled B, E and I are regarded as in the scope of corresponding hedge cue. Notice that a scope labeling is conducted for only a hedge cue, instead of a sentence. If a sentence has more than one hedge cue, copies of the sentence are made. In every copy only one hedge cue is investigated by labeling its scope.

As an example of hedge cue and scope labeling, words in a sentence, "*Furthermore, inhibition can be blocked by actinomycin D, indicating a requirement for de novo transcription.*", with their hedge cue and scope labels are given in Table 1. Hedge cue in the sentence is *indicating* and its scope contains all tokens in "*indicating a requirement for de novo transcription*".

Word	Hedge Cue Label	Scope Label
Furthermore	O	O
,	O	O
inhibition	O	O
can	O	O
be	O	O
blocked	O	O
by	O	O
actinomycin	O	O
D	O	O
,	O	O
indicating	I	B
a	O	I
requirement	O	I
for	O	I
de	O	I
novo	O	I
transcription	O	E
.	O	O

Table 1: A sentence with hedge cue and scope label

4 Notations of Feature Templates

Hedge cue and scope labeling adopt independent feature selections. 152 and 164 feature templates are initially considered for them. Except for original ones, features inspired by the following resources are also put in our initial feature sets:

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

b) Related works such as named entity recognition (Collins and Singer, 1999) and text chunking (Ramshaw and Marcus, 1995; Zhang et al., 2001)

c) Some literature on dependency parsing (McDonald and Pereira, 2006; Nivre, 2009)

Since an optimal feature template subset cannot be expected to be extracted from so large a set by hand, a greedy feature selection algorithm according to Zhao et al. (2009b) is applied. The algorithm is basically conducted by randomly choosing 1/10 features at first, followed by adding useful templates and removing ineffective ones greedily.

Most feature templates in our feature sets are formed by syntactic elements, including syntactic connections, paths, families etc. Many of these syntactic features are originally for semantic tasks. The parser given by Zhao et al. (2009a) is used to generate dependency structures.

Feature templates are from various combinations or integrations of the following elements:

Basic Properties. This part of features includes word form (*form*), lemma (*lemma*), part-of-speech tag (*pos*) and syntactic dependency type (*dprel*).

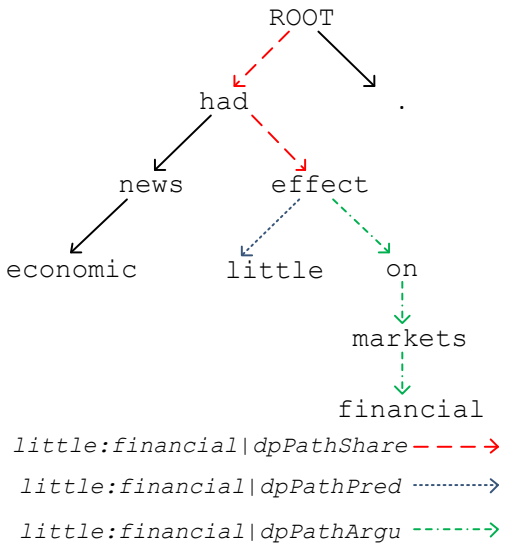

Figure 1: Syntactic paths

Syntactic Connections. This includes syntactic head (*head*), left (right) farthest (nearest) child (*lm, rm, ln*, and *rn*) and high (low) support verb, noun or preposition (*highSupportVerb, lowSupportVerb, highSupportNoun, lowSupportNoun, highSupportPrep, lowSupportPrep*). Here we specify support noun as an example: From a given word to the syntactic root along the syntactic tree, the first noun is defined as its low support noun, and the nearest one to the root (farthest to the given word) is defined as its high support noun. Figure 1 illustrates the dependency tree of sentence: "*Economic news had little effect on financial markets.*" In this parsing tree, path from word *financial* to the root is: *financial-markets-on-effect-had-ROOT*, so the low support noun of *financial* is *markets* and the high support noun is *effect*. The concept of support verb was broadly used (Toutanova et al., 2005; Xue, 2006; Jiang and Ng, 2006) , and it is extended to noun and preposition.

Paths. There are two basic types of paths. One is the linear path (*linePath*) in the sequence, the other is the path in the syntactic parsing tree (*dpPath*). Starting and ending token of a path is separated by ":", followed by a "|" and the path type. For example, x.*lowSupportVerb:*x|*dpPath* represents the path in syntactic tree from x's low support verb to x itself. We further divide paths in parsing tree into four sub-types: *dpPath* itself is the path in the syntactic tree, from starting to ending word; Assume that *r* is the least common ancestor of the starting and ending word, then *dpPathShare* is the path from *r* to the root, *dpPathPred* is from the starting word to *r*, and *dpPathArgu* is from the the ending word to *r*. Figure 1 illustrates these three types of path from *little* to *financial*. Finally, features like *dpPath.dprel* collects all the dependency types along the path and *dpPath* without a *dprel* collects all the tokens along the path.

Family. *children*, gathering all the syntactic children of current token, is used in the templates.

Concatenation of Elements. For all the elements collected by *dpPath*, *children* etc., we use three strategies to concatenate them 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 neighboring strings.For instance, x.*lowSupportVerb:*x|*dpPath.dprel.seq* means that we collect all the dependency types along the path from x's low support verb to x and make up a feature string using these tokens without any removing or sorting.

Hedge Cue Dictionary and Others. Hedge cues in BioScope corpus are collected and put in a dictionary. In order to extend the keyword list, Wikipedia "weasels"[2] are taken advantage of by collecting another dictionary. Whether a word is in the two dictionaries (*dicB* and *dicW*) are added to feature templates. In feature set for scope labeling, *cue* represents that the word is in a hedge cue or not.

In all feature templates, we take x as current token to be labeled, and x_m to denote neighbor words. $m > 0$ represents that it is the m_{th} word after current word and $m < 0$ for word $-m_{th}$ before current word. Finally, "+" is used to simply concatenate different feature strings.

5 Scope Finding for Multi-cue Sentences

Since scopes must be intact constituents (phrases, clauses, etc) of sentences, namely, subtrees in syntax trees that never *partly* cover each other, scope of a hedge cue in a sentence should not overlap one another. To specify, assuming that A and B are two scopes for two hedge cues in one sentence. Then if

[2]http://en.wikipedia.org/wiki/wikipedia:avoid_weasel_words

Group Name	Training Set	Test Set	Feature Set	Baseline
abs-abs	**abs** (10-fold)		$f^{abs-abs}$	**baseline**^{abs}
abs-full	**abs**	**full**	$f^{abs-abs}$	**baseline**^{full}
abs-clin	**abs**	**clin**	$f^{abs-abs}$	**baseline**^{clin}
abs-full*	**abs**	**full**	$f^{abs-full}$	**abs-full**
abs-clin*	**abs**	**clin**	$f^{abs-clin}$	**abs-clin**
full-full	**full** (6-fold)		$f^{full-full}$	**abs-full***
clin-clin	**clin** (6-fold)		$f^{clin-clin}$	**abs-clin***

Table 2: Groups of Evaluations

$A \cap B \neq \emptyset$, there will be only two possibilities: $A \subseteq B$ or $B \subseteq A$. Results of scope finding for multi-cue sentences are ensured non-overlapping by introducing an instructional feature called *IPS* (In Previous Scopes), which is denoted as *ips* in feature template sets. The *IPS* has two possible values: "*inPS*" and "*outPS*". For each multi-cue sentence, we record a set of tokens denoted by **S**, which is empty initially. In scope labeling for hedge cues, the set is maintained as follows:

a) Before labeling scope for a cue in a multi-cue sentence, tokens in **S** are collected and a feature tag "*inPS*" is given to them. "*outPS*" is given to others.

b) After labeling scope for a cue in a multi-cue sentence, tokens labeled B, I and E are put in **S**.

The operations are conducted in both training and decoding. By assigning this feature tag before labeling scope for next cue, every word that has already been in one or more scopes is marked off from others. Then the model can guided by the *IPS* tags to avoid illegal outputs. A labeling output is illegal if words labeled B and E have different values in *IPS*, which means one of them is already in scopes of other cues while the other one is not, indicating that the scope is overlapped by the other scope.

6 Experiments

As both tasks mentioned in this paper are formulated as sequence labeling, it is natural to use the framework of conditional random fields (Lafferty et al., 2001). The tasks are implemented and run by the frequently-used tool for sequence labeling: CRF++[3]. Models in this paper are trained and developed upon the BioScope corpus. The corpus consists of texts from 3 different types: biological paper abstracts (**abs**), biological full papers (**full**) and clinical free-texts (**clin**), which contain about 20000, 3000 and 4000 sentences, respectively. Annotation guideline and corpus details can be found in (Szarvas et al., 2008).

Based on the mentioned elements in section 4 and 5, five groups of feature template sets are finally selected. Each group contains two sets, one for hedge detection and the other for scope finding. These sets are denoted in the format of f^{m-n}. *m* and *n* stand for the training and development set while selecting the feature set. They can be *abs*, *full* and *clin*, representing **abs**, **full** and **clin** of BioScope Corpus. f^{m-n}_{hedge} is the set for hedge cue labeling and f^{m-n}_{scope} is the set for scope labeling. For example, the template set $f^{abs\_clin}_{hedge}$ is got from following operations: A Model for hedge cue labeling is trained on **abs** and tested on **clin** with a feature set, then greedy algorithm is applied to modify the set. Training, testing and greedy modification are repeatedly conducted until there is no promotion in performance. Thus, the feature set with highest performance is adopted as $f^{abs\_clin}_{hedge}$.

Seven groups of evaluations are conducted on all the three parts of the BioScope corpus. Table 2 lists the configurations of all the evaluations. A group name is denoted as **m-n**, in which **m** and **n** indicate the training and test set: **abs**, **full** or **clin**. In a group, two experiments, **m-n**$_{hedge}$ and **m-n**$_{scope}$, are included to represent hedge cue and scope labeling.

Subscript for each group name and feature set, *hedge* or *scope*, is not specified in the table. For example, the feature set $f^{abs-abs}$ for group **abs-abs** means we use $f^{abs-abs}_{hedge}$ and $f^{abs-abs}_{scope}$ for **abs-abs**$_{hedge}$ and **abs-abs**$_{scope}$, respectively.

In the first three groups, results in Morante

[3] http://crfpp.sourceforge.net/

and Daelemans (2009), which are best ones so far in hedge detection and scope finding, are brought as baselines (**baseline**^{abs}, **baseline**^{full} and **baseline**^{clin}). For **abs** literature we perform a 10-fold cross validation as **baseline**^{abs} does. Models for testing **full** and **clin** in **baseline**^{full}, **baseline**^{clin}, **abs-full** and **abs-clin** are trained on the whole **abs** corpus, which can be used to test whether the feature sets and models can be applied in different types of literature. In these three groups, template sets $f^{abs-abs}$ is uniformly used. Then we change the template sets to $f^{abs-full}$ and $f^{abs-clin}$, experiments with group **abs-full*** and **abs-clin*** using the same training and test sets as **abs-full** and **abs-clin** to find out the differences brought by specialized selected features. Besides, in order to investigate how training sets influence labeling outputs, the last two groups evaluate **full** and **clin** by models trained on their owns via performing 6-cross validation.

6.1 Selected Feature Template Sets

Feature template sets selected for hedge labeling are listed in Table 3. The first part of the table enumerates features without dependency elements and the second part lists syntactic ones. Only $f^{abs-abs}_{hedge}$, $f^{full-full}_{hedge}$ and $f^{clin-clin}_{hedge}$ are listed and are denoted as *A*, *F* and *C*.

Features in the second part of Table 3 indicate that hedge labeling is a task more than keyword identification and can also benefit from sophisticated syntactic features. Experimental results showed in section 6.4 can also support the claim.

The two parts of Table 4 enumerates non-syntactic and syntactic features in $f^{abs-abs}_{scope}$, $f^{full-full}_{scope}$ and $f^{clin-clin}_{scope}$. In $f^{abs-abs}_{scope}$ and $f^{full-full}_{scope}$, *ips* plays important roles, which demonstrates that this instructional feature is effective for scope labeling in multi-cue sentences. *ips* is not contained in $f^{clin-clin}_{scope}$ because the percentage of multi-cue sentences in **clin** is smaller than that in **abs** and **full**.

6.2 Results of Hedge Cue Labeling

In the hedge cue labeling task, a true positive case in evaluation is a token labeled "I" which is in a hedge cue according to gold standard. Our method gives scores in Table 5 with baselines correspondingly.

Generally, the method of sequence labeling

-	x_{-1}.*form*
-	x_1.*lemma*
-	x_{-1}.*pos*
-	*x.dicW*
A	x_{-1}.*lemma* + *x.lemma*
-	*x.lemma* + x_1.*lemma* + x_{-1}.*lemma* + *x.dicW* + x_1.*dicW* + x_{-1}.*dicW*
-	*x.dicB* + x_1.*dicB* + x_{-1}.*dicB* + x_2.*dicB* + x_{-2}.*dicB* + x_3.*dicB* + x_{-3}.*dicB*
-	*x.form*
-	x_{-2}.*form*
-	x_{-1}.*pos* + x_1.*pos*
-	x_{-1}.*lemma* + *x.lemma*
-	*x.lemma* + *x.dicB*
F	*x.pos* + x_1.*pos* + x_{-1}.*pos* + *x.dicW* + x_1.*dicW* + x_{-1}.*dicW*
-	*x.dicB* + *x.dicW* + x_{-1}.*dicB* + x_{-1}.*dicW* + x_1.*dicB* + x_1.*dicW* + x_{-2}.*dicB* + x_{-2}.*dicW* + x_2.*dicB* + x_2.*dicW*
-	*x.form*
-	x_1.*pos*
-	*x.dicW*
-	x_{-1}.*lemma* + *x.lemma* + x_1.*lemma*
C	*x.lemma* + *x.dicB*
-	*x.lemma* + *x.pos* + *x.dicB* + *x.dicW*
-	*x.lemma* + x_{-1}.*lemma* + x_1.*lemma* + x_{-2}.*lemma* + x_2.*lemma* + *x.dicW* + x_{-1}.*dicW* + x_1.*dicW* + x_{-2}.*dicW* + x_2.*dicW*
-	*x*:*x.head*|*dpPath.dprel*
-	*x.form* + *x*:*x.children*|*dpPath.dprel.bag*
-	*x.lowSupportPrep*:*x*|*dpPathShared.seq*
A	*x.lowSupportVerb*:*x*|*dpPathPred.dprel.seq*
-	*x.highSupportVerb*:*x*|*dpPathPred.dprel.seq*
-	*x.lowSupportVerb*:*x*|*dpPathShared.dprel.seq*
-	*x.highSupportVerb*:*x*|*dpPathShared.dprel.seq*
-	*x*:*x.lm*|*dpPath.dprel*
-	*x.highSupportNoun.pos*
-	*x*:*x.children*|*dpPath.dprel.noDup*
-	*x*:*x.children*|*dpPath.dprel.bag*
F	*x.highSupportVerb.form*
-	*x.lowSupportNoun.lemma*
-	*x.lowSupportVerb.form*
-	*x.lowSupportVerb*:*x*|*dpPathShared.dprel.seq*
-	*x.lowSupportVerb*:*x*|*dpPathArgu.dprel.seq*
-	*x*:*x.children*|*dpPath.dprel.noDup*
-	*x*:*x.children*|*dpPath.dprel.bag*
C	*x.highSupportVerb.form*
-	*x.lowSupportNoun.lemma*
-	*x.highSupportNoun*:*x*|*dpPathShared.dprel.seq*

Table 3: Selected feature template sets for hedge cue labeling

-	x_1.*lemma*
-	x_1.*pos*
-	*x.pos* + *x.dicB* + *x.dicW*
-	*x.form* + x_{-1}.*form* + x_{-2}.*form*
A	*x.lemma* + *x.pos* + *x.dicB* + *x.dicW*
-	x_{-1}.*cue* + *x.cue* + x_1.*cue*
	x_{-2}.*cue* + x_2.*cue*
-	*x.lemma* + x_1.*lemma* + x_{-1}.*lemma*
	+ *x.dicW* + x_1.*dicW* + x_{-1}.*dicW*
	+ *x.ips* + x_1.*ips* + x_{-1}.*ips*
-	*x.form*
-	x_{-1}.*form*
-	x_1.*pos*
-	x_{-1}.*lemma* + *x.lemma*
-	*x.lemma* + *x.dicB*
F	*x.pos* + *x.dicB* + *x.dicW*
-	x_{-1}.*cue* + *x.cue* + x_1.*cue*
	x_{-2}.*cue* + x_2.*cue*
-	*x.form* + *x.ips* + x_{-1}.*form*
	+ x_{-1}.*ips* + x_1.*form* + x_1.*ips*
	+ x_{-2}.*form* + x_{-2}.*ips* + x_2.*form*
	+ x_2.*ips*
-	*x.form*
-	x_{-1}.*pos* + *x.pos*
-	*x.lemma* + *x.dicB*
C	*x.lemma* + *x.dicW*
-	*x.pos* + *x.dicB* + *x.dicW*
-	*x.lemma* + *x.pos* + *x.dicB* + *x.dicW*
-	*x.cue* + x_{-1}.*cue* + x_1.*cue*
-	*x:x.rm|dpPath.dprel*
-	*x.lm.form*
-	*x.lowSupportVerb.form*
-	*x.rm.lemma* + *x.rm.form*
A	*x.lowSupportVerb.form*
-	*x:x.children|dpPath.dprel.noDup*
-	*x:x.children|dpPath.dprel.bag*
-	*x.lowSupportNoun:x|dpPathShared.dprel.bag*
-	*x.highSupportVerb:x|dpPathShared.dprel.bag*
-	*x.lm.form*
-	*x.lemma* + *x.pphead.form*
-	*x.lm.lemma* + *x.pos*
-	*x.lowSupportVerb.form*
F	*x.lowSupportProp:x|dpPathPred.dprel.seq*
-	*x.lowSupportProp:x|dpPathShared.dprel.seq*
-	*x.lowSupportVerb:x|dpPathPred.deprel.seq*
-	*x.lowSupportVerb:x|dpPathShared.pos.seq*
-	*x.lowSupportProp:x|dpPathShared.pos.seq*
-	x_{-1}.*lm.form*
-	*x:x.children|dprel.noDup*
C	*x.highSupportNoun:x|dpTreeRelation*
-	*x.highSupportNoun:x|dpPathArgu.dprel.seq*
-	*x.lowSupportProp:x|dpPathShared.dprel.seq*

Table 4: Selected feature template sets for scope labeling

Evaluation	Prec.	Recall	F_1
baseline$^{abs}_{hedge}$	0.908	0.798	0.848
abs-abs$_{hedge}$	0.938	0.879	**0.908**
baseline$^{full}_{hedge}$	0.734	0.682	0.716
abs-full$_{hedge}$	0.822	0.772	**0.796**
abs-full*$_{hedge}$	0.861	0.813	**0.836**
full-full$_{hedge}$	0.883	0.776	0.826
baseline$^{clin}_{hedge}$	0.881	0.275	0.419
abs-clin$_{hedge}$	0.545	0.298	0.385
abs-clin*$_{hedge}$	0.771	0.448	**0.567**
clin-clin$_{hedge}$	0.979	0.976	**0.978**

Table 5: Results of hedge cue labeling

x.head
x.form + x_1.*form*
x.form + x_{-1}.*form*
x.form + *x.dicB*
x.form + *x.dicB* + *x.dicW*
x.lemma + *x:x.head|dpPath.dprel*
x.pos + *x.lemma* + x_{-1}.*pos* + x_{-1}.*lemma*

Table 6: Feature template set: $f^{abs-clin}_{hedge}$

adopted in our seven experiments gives distinctly improved results against the baselines. The only exception is **abs-clin**$_{hedge}$ with the F_1 score 0.385, which is lower than the score 0.419 of **baseline**$^{clin}_{hedge}$. The model used in **abs-clin**$_{hedge}$ is trained with the feature set $f^{abs-abs}_{hedge}$. Although this set helps **abs-abs**$_{hedge}$ reach 0.908, a quite satisfying result, it is not so helpful when the test set changes to **clin**. Sentences in **clin** corpus, often conforming to some patterns, e.g. "Cough and fever for X days", and describing state of illness and prescriptions, are generally short and looks quite different from scientific paper abstracts and texts. Differences between them can also be reflected in the scale of feature set $f^{abs-clin}_{hedge}$ (Table 6): only 7 feature templates remain to be useful, indicating that very few common properties, especially the distribution of hedge cues, can be found in **clin** and **abs**. This explains why **abs-clin**$_{hedge}$ and **abs-clin***$_{hedge}$ perform poor, and when the training set and test set are both **clin**, the evaluation **clin-clin**$_{hedge}$ gives significant promotion in performance.

On the other hand, scores of **abs-full**$_{hedge}$ and **abs-full***$_{hedge}$ suggest that it is practicable to label scientific full papers using models trained on paper

abstracts. The score is even slightly reduced when the training set changes to **full** itself, probably because its scale is much smaller than that of **abs**.

Specifically selected feature sets $f_{hedge}^{abs-full}$ and $f_{hedge}^{abs-clin}$ help **abs-full***$_{hedge}$ and **abs-clin***$_{hedge}$ make visible increase against **abs-full**$_{hedge}$ and **abs-clin**$_{hedge}$. This proves that independent feature selections for specific training and test sets are usually effective. Finally, although **full-full**$_{hedge}$ performs poorer than **abs-full***$_{hedge}$, results of **clin-clin**$_{hedge}$ imply it is better to use the same type of training set as the test set in hedge cue labeling.

6.3 Results of Scope Labeling

In scope labeling, two kinds of measurements: F_1 score and PCS (percentage of correct scope), are adopted. A true positive case in the F-measure is a *token* correctly given one of the following labels: B, E and I. If the token is in more than one scope, all the scope labels should be correct. And PCS is used to measure the percentage of *hedge cues* that have correct scopes. A correct scope in PCS means the scope is rightly given a pair of beginning and ending words. Hedge cues have already been labeled according to gold standard before scope labeling in these experiments. Results of scope labeling with gold standard hedge cues are given in Table 7.

Evaluation	Prec.	Recall	F_1	PCS
baseline$_{abs}$	0.897	0.891	0.894	0.771
abs-abs$_{scope}$	0.918	0.921	**0.920**	**0.898**
baseline$_{full}$	0.778	0.771	0.774	0.479
abs-full$_{scope}$	0.777	0.814	**0.795**	**0.668**
abs-full*$_{scope}$	0.826	0.826	**0.826**	**0.702**
full-full$_{scope}$	0.869	0.826	**0.847**	**0.827**
baseline$_{clin}$	0.792	0.781	0.786	0.606
abs-clin$_{scope}$	0.784	0.834	**0.808**	**0.672**
abs-clin*$_{scope}$	0.812	0.908	**0.857**	**0.807**
clin-clin$_{scope}$	0.853	0.938	**0.894**	**0.895**

Table 7: Results of scope labeling with gold-standard hedge cues

The results suggest similarly to the results of hedge cue labeling, that specialized feature selection according to specific training test set can greatly promote the performance, and it is better for the training and test set to be the same type. On the other hand, when using $f_{scope}^{abs-abs}$ as the feature set, unlike the poor performance of **abs-clin**$_{hedge}$ (F_1 score 0.385) using $f_{hedge}^{abs-abs}$, **abs-clin**$_{scope}$ gives a favorable score 0.672, showing that scope finding is a task less decided by specific hedging characteristics of literature type. This is because scope labeling is a more likely to be a syntactic task compared with hedge cue labeling.

Joint evaluation of hedge cue labeling and scope labeling are also carried out. Results of scope finding with predicted hedge cues are given in Table 8. A true positive case here in the F-measure is a *token* with both correct hedge cue label "I" and scope label "B", "I" or "E". And the PCS here equals to the number of correct hedge cues with correct scopes divided by number of hedge cues in the gold standard. The results indicate that joint detection of hedges and their scopes is practicable.

Evaluation	Prec.	Recall	F_1	PCS
baseline$_{abs}$	0.858	0.724	0.785	0.656
baseline$_{full}$	0.680	0.532	0.597	0.359
baseline$_{clin}$	0.682	0.265	0.382	0.262
abs-abs	0.866	0.853	**0.859**	**0.794**
full-full	0.790	0.701	**0.743**	**0.683**
clin-clin	0.843	0.911	**0.876**	**0.837**

Table 8: Results of scope finding with predicted hedge cues

Finally, hedge detection and scope finding are adopted as subtasks in the shared task of CoNLL-2010. The training set in this shared task are **abs** and **full** of BioScope corpus and evaluating set is some newly extracted paper sentences. Template sets $f_{hedge}^{abs-full}$ and $f_{scope}^{abs-full}$ are used in the two phases of labeling to compare our results with the best score in the official ranking of this event. We only use literature in **abs** to train the model. The results are given in Table 9. Notice that a true positive case here is a *sentence* with all the hedge cues and their scopes correctly labeled in it. The results demonstrate that our methods outperform the best system in this event.

Evaluation	Prec.	Recall	F_1
abs-full	0.481	0.860	**0.617**
CoNLL's best	0.596	0.552	0.573

Table 9: Results on evaluation set of CoNLL 2010 ST

	abs-abs$_{hedge}$	**full-full**$_{hedge}$	**clin-clin**$_{hedge}$
Original (F_1)	0.908	0.826	0.978
NonSyn (F_1)	0.891	0.808	0.963
Improvement	**0.017**	**0.018**	**0.015**

Table 10: Hedge labeling with and without syntactic features

	abs-abs$_{scope}$	**full-full**$_{scope}$	**clin-clin**$_{scope}$
Original (F_1)	0.920	0.847	0.894
NonSyn (F_1)	0.893	0.824	0.877
Improvement	**0.027**	**0.023**	**0.017**
Original (PCS)	0.898	0.827	0.895
NonSyn (PCS)	0.869	0.807	0.870
Improvement	**0.029**	**0.020**	**0.025**

Table 11: Scope labeling with and without syntactic features

6.4 Improvements by Syntactic Features

Among the 154 and 162 features initialized for the two tasks, 96 contains elements derived from syntactic dependencies. In the final selected feature sets, about 50 percent features are syntactic ones. In order to test whether these syntactic features work or not, optimal template sets without these features are also greedily selected for comparison. These sets are denoted as *NonSyn*, in contrast to those *Original* ones that corresponding groups adopt. Results of the tests are shown in Table 10 and Table 11 for hedge labeling and scope labeling, respectively.

It can be seen that both of the two labeling tasks benefit about 2 percent promotion from the syntactic features. The results in Table 10 show that syntactic information is effective in hedge labeling, indicating that a keyword extraction formulation can not cover all the information that hedge detection needs.

	abs	**full**	**clin**
Original (PCS)	0.827	0.758	0.809
NonSyn (PCS)	0.782	0.724	0.759
Improvement	**0.045**	**0.034**	**0.050**

Table 12: Scope labeling performances for scopes longer than 10 words

On the other hand, performances of labeling scopes which are longer than 10 words are also evaluated and given in Table 12, which give us a suggestion about the promotion brought by syntactic features: When syntactic features are removed from initial sets, accuracy of labeling for longer scopes decreases more significantly than that for the shorter ones, indicating that features based on dependency trees are indispensable when the scope is too wide to be reached by non-syntactic features. Syntactic elements such as *children*, *head* and *dpPath* can combine a word with other ones which are far away in sequence but close in syntactic tree. These sequentially-distant but syntactically-close tokens give important instructions while labeling a word. In fact as a task similar to chunking, a kind of shallow parsing, scope finding can naturally benefit from full dependency parsing. If the corpus contains more sentences with long scopes, the increase might be more remarkable.

7 Conclusions

We present a novel method to find hedges and their scopes using sequence labeling. Experimental results show that it is a proper formulation for the problems. Syntactic features derived from dependencies are exploited, which proves quite effective in both tasks. These features have revealed some linguistic characteristics of the hedge device, indicating that hedge cue detection is more than a keyword matching and long scopes depend largely on the syntactic structures, which will help us understand the phenomenon of hedges and their scopes empirically in forthcoming works.

References

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proc. of the JointSIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Viola Ganter and Michael Strube. 2009. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proc. of ACL-IJCNLP 2009*, 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 *Proc. of 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.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML 2001*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

George Lakoff. 1972. Hedges: a study in meaning criteria and the logic of fuzzy concepts. *Chicago Linguistics Society Papers*, 8:183–228.

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

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *European Association for Computational Linguistics (EACL-2006)*, pages 81–88, Trento, Italy, April.

Ben Medlock and Ted Briscoe. 2008. Weakly supervised learning for hedge classification in scientific literature. In *Proc. of ACL 2008*, 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 *Proc. of BioNLP 2009*, pages 28–36, Boulder, Colorado, June.

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

Frank Robert Palmer. 1986. *Mood and modality*. Cambridge University Press, Cambridge.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proc. of the 3rd Workshop on Very Large Corpora*, pages 88–94.

Roser Sauri, Marc Verhagen, and James Pustejovsky. 2006. Annotating and recognizing event modality in text. In *Proc. of FLAIRS 2006*, pages 333–339.

Gyorgy Szarvas, Veronika Vincze, Richard Farkas, and Janos Csirik. 2008. The bioscope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proc. 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 *Proc. of ACL 2008*, 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 *Proc. of ACL-2005*, pages 589–596, Ann Arbor, USA.

Veronika Vincze, Gyorgy Szarvas, Richard Farkas, Gyorgy Mora, and Janos Csirik. 2008. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 11:433–54.

Nianwen Xue. 2006. Semantic role labeling of nominalized predicates in chinese. In *Proc. of NAACL 2006*, pages 431–438, New York City, USA, June.

Tong Zhang, Fred Damerau, and David Johnson. 2001. Text chunking using regularized winnow. In *Proc. of ACL 2001*, 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 *Proc. of (CoNLL-2009)*, 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 *Proc. of EMNLP-2009*, pages 30–39, Singapore.