

RESEARCH ARTICLE

# Hedge Scope Detection in Biomedical Texts: An Effective Dependency-Based Method

Huiwei Zhou<sup>1\*</sup>, Huijie Deng<sup>1</sup>, Degen Huang<sup>1</sup>, Minling Zhu<sup>2</sup>

**1** School of Computer Science and Technology, Dalian University of Technology, Dalian, Liaoning, China, **2** School of Computer, Beijing Information Science and Technology University, Beijing, China

\* [zhouhuiwei@dlut.edu.cn](mailto:zhouhuiwei@dlut.edu.cn)

## Abstract

Hedge detection is used to distinguish uncertain information from facts, which is of essential importance in biomedical information extraction. The task of hedge detection is often divided into two subtasks: detecting uncertain cues and their linguistic scope. Hedge scope is a sequence of tokens including the hedge cue in a sentence. Previous hedge scope detection methods usually take all tokens in a sentence as candidate boundaries, which inevitably generate a large number of negatives for classifiers. The imbalanced instances seriously mislead classifiers and result in lower performance. This paper proposes a **dependency-based candidate boundary selection method** (DCBS), which selects the most likely tokens as candidate boundaries and removes the exceptional tokens which have less potential to improve the performance based on dependency tree. In addition, we employ the composite kernel to integrate lexical and syntactic information and demonstrate the effectiveness of structured syntactic features for hedge scope detection. Experiments on the CoNLL-2010 Shared Task corpus show that our method achieves 71.92% F1-score on the golden standard cues, which is 4.11% higher than the system without using DCBS. Although the candidate boundary selection method is only evaluated on hedge scope detection here, it can be popularized to other kinds of scope learning tasks.



## OPEN ACCESS

**Citation:** Zhou H, Deng H, Huang D, Zhu M (2015) Hedge Scope Detection in Biomedical Texts: An Effective Dependency-Based Method. PLoS ONE 10 (7): e0133715. doi:10.1371/journal.pone.0133715

**Editor:** Christian Lovis, University Hospitals of Geneva, SWITZERLAND

**Received:** November 15, 2014

**Accepted:** June 30, 2015

**Published:** July 28, 2015

**Copyright:** © 2015 Zhou et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are available from Figshare: <http://dx.doi.org/10.6084/m9.figshare.1476411>.

**Funding:** This work was supported by a grant from the National Natural Science Foundation of China (No. 61272375 and 61173100, <http://www.nsf.gov.cn/Portal0/default152.htm>).

**Competing Interests:** The authors have declared that no competing interests exist.

## Introduction

Hedges are linguistic devices that indicate uncertain or unreliable information. Hedged information is usually used in science texts, especially in the biomedical domain to express impressions or hypothesized explanations of experimental results. According to the statistics on BioScope corpus [1], 17.69% of the sentences in abstract section and 22.29% of the sentences in full paper section contain speculative fragments. In order to distinguish factual and uncertain information, detecting hedged information is an increasingly important task in biomedical information extraction. The CoNLL-2010 Shared Task [2] is dedicated to the detection of uncertain information. The shared task contains two subtasks: Task 1 aims to identify hedge cues and Task 2 devotes to detecting the in-sentence scope of a given cue.

A hedged sentence taken from the CoNLL-2010 Shared Task corpus is shown as follows:

Sentence 1: *Our data indicate that < xscope > the mutagenic DNA deaminases are < cue > potentially < /cue > an important target for hormonal regulation < /xscope >.*

The token “*potentially*”, namely hedge cue, indicates that the following statement is not backed up by facts. The in-sentence scope of the hedge cue “*potentially*” is the statement “*the mutagenic DNA deaminases are potentially an important target for hormonal regulation*”.

Researches on hedge cue identification have been developed rapidly [3–5]. However, the results of hedge scope detection are not satisfied. Hedge scope detection is a difficult task, since it falls within the scope of semantic analysis of sentences exploiting syntactic patterns. This paper focuses on the hedge scope detection task.

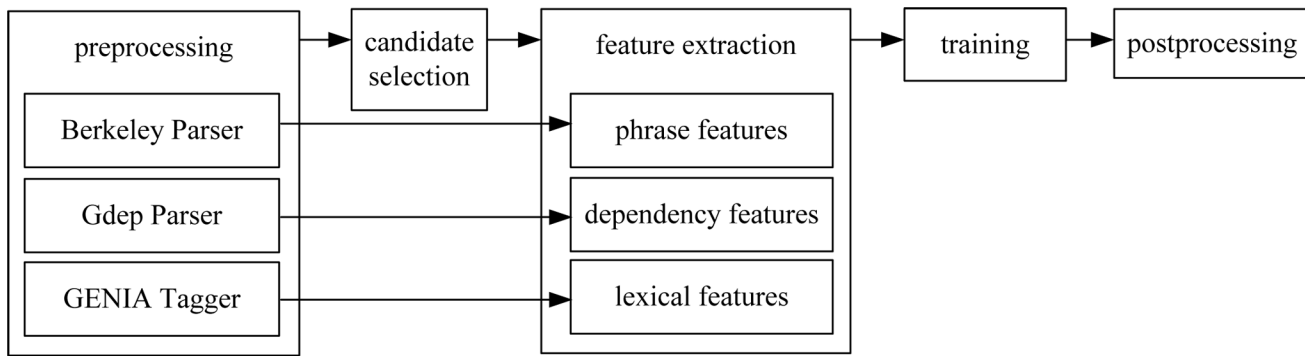
Generally, hedge scope detection approaches can be divided into two categories: rule based methods and machine learning based methods. Rule based methods detect scope by constructing syntactic rules. Özgür et al. [6] and Øvrelid et al. [7] manually compiled rules, and Apostolova et al. [8] and Zhou et al. [9] automatically extracted rules based on syntactic structures. Rule based methods could make full use of syntactic information and have achieved good performance in the specific resource, but the extracted rules are hard to be developed to a new resource.

Machine learning based methods formulate the hedge scope detection task as a token classification problem, which usually classify each token in a sentence as being the first element of the scope (F-scope), the last (L-scope), or neither (None). We refer to such traditional token-based methods as TTB in this article. Morante and Daelemans [10] utilized lexical and contextual features to learn three classifiers to predict F-scope, L-scope, and None respectively. Morante et al. [11] extended this work by extracting flat dependency features to improve the detection performance. They achieved the highest F1-score (57.32%) on CoNLL-2010 Shared Task 2. Velldal et al. [12] combined manual rules and machine learning to exploit syntactic and surface-oriented information for the scope detection task.

TTB regards scope boundary tokens as positives, and the others as negatives for scope boundary classifiers. For example, the F-scope classifier takes the F-scope tokens as positives and the others on the left side of a given cue (including the cue itself) as negatives. This inevitably generates plenty of negatives. Excessive negatives mislead classifier and degrade classification performance. Zhu et al. [13] altered the classification unit from token to phrase by formulating the scope learning as shallow semantic parsing problem, which decreased candidate instances significantly. However, regarding the great granularity phrase as candidate unit is too coarse or general to capture semantic information.

For both rule based and machine learning based methods, syntactic information plays an important role in hedge scope detection. Tree kernel [14] can capture structured syntactic information and has been applied in various NLP tasks like relation extraction [15, 16], semantic role labeling [17], events extraction [18] and drug-drug interaction detection [19, 20], etc. Zhou et al. [21] applied tree kernel with structured phrase features to hedge scope resolution task. Zou et al. [22] utilized tree kernel to capture phrase and dependency structure information to optimize the scope detection performance. However, adjacent tokens in a sentence are hard to be classified by tree kernel, as adjacent tokens have similar structure and contextual information.

This paper proposes a dependency-based candidate boundary selection method (DCBS) to decrease candidate instances and enhance the discriminability of instances for classifiers effectively. DCBS selects the most likely tokens as candidate boundary and removes the exceptional tokens which have less potential to improve the performance based on dependency tree. Furthermore, the composite kernel consisting of the polynomial kernel and the tree kernel is employed to capture lexical and syntactic structure information. Although our approach is



**Fig 1. System architecture.**

doi:10.1371/journal.pone.0133715.g001

only evaluated on hedge scope detection task in this paper, it is portable to other kinds of scope learning tasks.

## Materials and Methods

Our scope detection system consists of five steps: preprocessing, candidate selection, feature extraction, training and postprocessing as shown in Fig 1. In the preprocessing step, the CoNLL-2010 Shared Task corpus are processed with Berkeley Parser (<http://code.google.com/p/berkeyparser/>), Gdep Parser (<http://pepple.ict.usc.edu/sagae/parser/gdep/>) and GENIA Tagger (<http://www-tsujii.is.s.utokyo.ac.jp/GENIA/tagger/>) to get phrase tree, dependency tree and lexical information, respectively. In the candidate selection step, we use DCBS to select candidate boundaries. In the feature extraction step, phrase features, dependency features and lexical features are extracted based on phrase tree, dependency tree and lexical information, which are generated in the preprocessing step. In the training step, classifiers are learned by using SVM-LIGHT-TK 1.2 toolkit (<http://disi.unitn.it/moschitti/Tree-Kernel.htm>) that provides the convolution tree kernel. Finally, the postprocessing rules are adopted to get the complete sequence of the scope in the postprocessing step. In the following part of this section, we will describe the detailed implementation of the hedge scope detection system.

### Dependency-based Candidate Boundary Selection Method

Token dependency reflects semantic modification relationships of in-sentence tokens. For two tokens with dependency relation, if one token falls within the hedge scope, the other is likely to be in the scope too. Therefore, the left one of the two tokens with dependency relation is more probable as the F-scope candidate boundary than the right one. The right one is not necessary to be selected as the F-scope candidate boundary and should be eliminated from F-scope candidates. Similarly, the right one of the two tokens with dependency relation is more likely as the L-scope candidate, and the left one should be eliminated from L-scope candidates. According to these analyses, we propose a dependency-based candidate boundary selection method (DCBS) for hedge scope detection. To concisely describe DCBS, each token in a sentence is numbered sequentially as shown in Fig 2. The L-scope (F-scope) candidate boundary selection algorithm based on dependency tree is described in Fig 3.

Taking the sentence 1 as an example, the L-scope candidate selection process with DCBS is shown in Fig 4. A brief description is as follows:

Sentence 1: Our data indicate that the mutagenic DNA deaminases are potentially an important target for hormonal regulation .																
Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

**Fig 2. The sequence number of tokens in sentence 1.**

doi:10.1371/journal.pone.0133715.g002

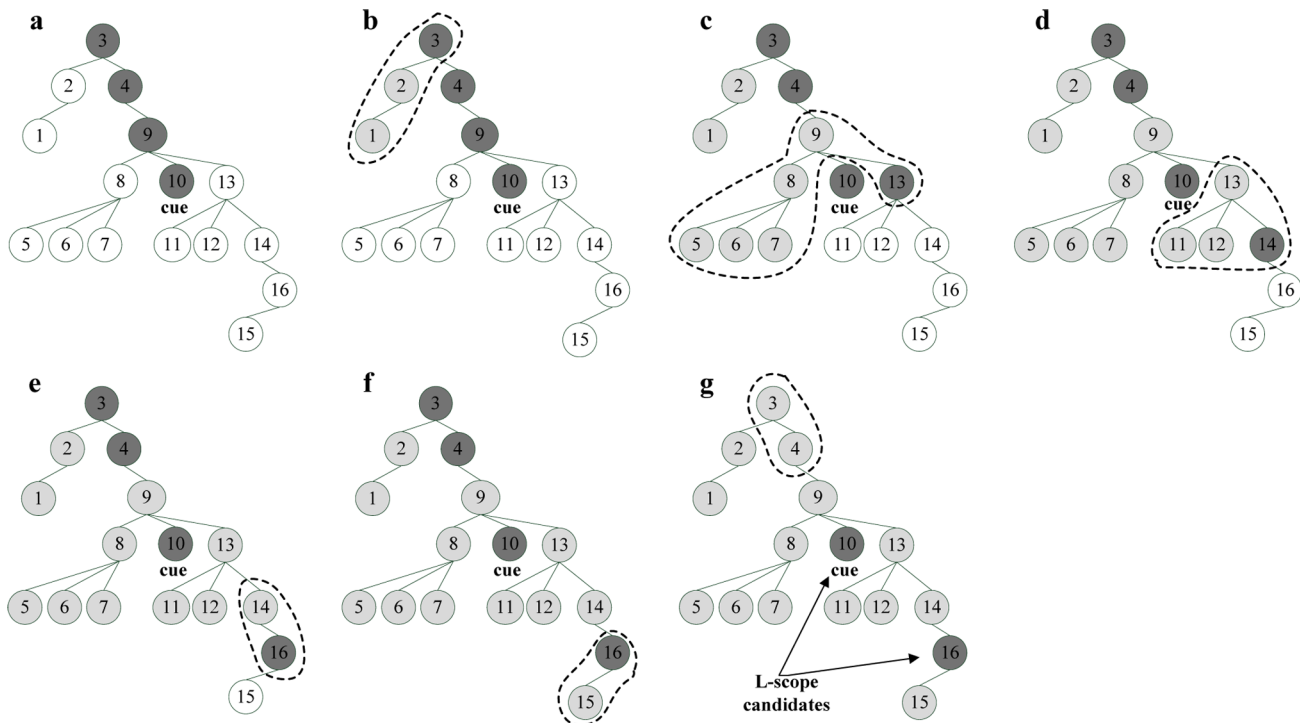
**Input:**  $T$ , a dependency tree with  $n$  nodes  
**Output:** The L-scope (F-scope) candidate boundary nodes  
**Remarks:** Color meanings are black = potential candidate node, grey = non-candidate node, and white = undetermined node. A node  $w$  is a child of a node  $v$  in a dependency tree. A node  $c$  is the cue node.  
**Procedure:**

- 1) // Initialize all nodes' color.
  - change cue node  $c$  and its ancestors to black;
  - change the others to white;
- 2) // Select L-scope (F-scope) candidate boundaries.
  - for** each black node  $v$  with white children;
  - for** each white child  $w$  of black node  $v$ ;
  - if**  $w < v$ , change  $w$  and its descendants to grey; (**if**  $w > v$ , change  $w$  and its descendants to grey;)
  - else if**  $w > v$ , change  $w$  to black and  $v$  to grey; (**else if**  $w < v$ , change  $w$  to black and  $v$  to grey;)
  - end for**
  - end for**
- 3) // Output L-scope (F-scope) candidate nodes.
  - for** each black node  $v$ ;
  - if**  $v < c$ , change  $v$  to grey; (**if**  $v > c$ , change  $v$  to grey;)
  - else** output black node  $v$ ;
  - end for**

**Fig 3. The L-scope (F-scope) candidate boundary selection algorithm.**

doi:10.1371/journal.pone.0133715.g003

1. Initial all nodes' color as shown in Fig 4a. Change cue node 10 and its ancestors (node 3, 4, 9) to black. Change the others to white.
2. Select the L-scope candidate boundaries from Fig 4b to 4f one by one. For each black node with white children, the black node is compared to each of its white children. If its white child is smaller than the black node, change its white child and the descendants of its white child to grey. If its white child is larger than the black node, change its white child to black and the black node to grey. The changes of node's color are shown in the dash line in Fig 4 (Fig 4b for node 3, Fig 4c for node 9, Fig 4d for node 13, Fig 4e for node 14, Fig 4f for node 16). Taking black node 9 in Fig 4c as an example, black node 9 is compared to each of its white children (node 8, 13). 8 is smaller than 9, so change node 8 and its descendants (node 5, 6, 7) to grey. 13 is larger than 9, so change node 13 to black and node 9 to grey.
3. 3 and 4 are smaller than 10, so change node 3 and 4 to grey. Output node 10 and 16 as L-scope candidate boundaries, as shown in Fig 4g.



**Fig 4. An example of the L-scope candidate selection process with DCBS.** (a) Initialize all nodes' color; Select the L-scope candidate boundary from (b) to (f); (g) Output the L-scope candidate nodes.

doi:10.1371/journal.pone.0133715.g004

## The Composite Kernel for Hedge Scope Detection

The hedge scope detection task falls within the scope of semantic analysis of sentences exploiting syntactic structure. To integrate lexical and syntactic information, the composite kernel consisting of the polynomial kernel and the convolution tree kernel is applied to our system.

**The polynomial kernel.** The polynomial kernel  $K_{poly}(x_i, x_j) = (x_i \cdot x_j + 1)^d$  is applied to obtain lexical information. The  $d$ -th polynomial kernel function can build an optimal separating hyperplane which takes into account all combination of features up to  $d$ . The parameter  $d$  is set to 2 in this paper. The **lexical features** used in the polynomial kernel are listed as follows:

- $CandidateToken(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $CandidateStem(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $CandidatePos(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $CandidateChunk(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $HedgeToken(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $HedgeStem(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $HedgePos(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- $HedgeChunk(i) (i = -3, -2, -1, 0, 1, 2, 3)$
- *Distance: The number of tokens from the hedge cue to its candidate*

	#	Word	Stem	Pos	Chunk
	...	...	...	...	...
	7	DNA	DNA	NN	I-NP
	8	deaminases	deaminas	NNS	I-NP
Hedge cue	9	are	be	VBP	B-VP
	10	<b>potentially</b>	<b>potentially</b>	<b>RB</b>	<b>B-ADVP</b>
	11	an	an	DT	B-NP
	12	important	important	JJ	I-NP
	13	target	target	NN	I-NP
	14	for	for	IN	B-PP
L-scope	15	hormonal	hormonal	JJ	B-NP
	16	<b>regulation</b>	<b>regulation</b>	<b>NN</b>	<b>I-NP</b>
	17	.	.	.	O

**Fig 5. The partial lexical information of sentence 1 preprocessed by GENIA Tagger.**

doi:10.1371/journal.pone.0133715.g005

where  $i$  specifies the relative position from the current token. For example,  $CandidateToken(0)$  denotes the current candidate token,  $CandidateToken(-1)$  is the first token to the left,  $CandidateToken(1)$  is the first token to the right, etc. Using the sentence 1, its partial lexical information preprocessed by GENIA Tagger is shown in Fig 5. And the lexical features of the L-scope candidate “regulation” for the cue “potentially” are listed in Table 1.

**The convolution tree kernel.** The convolution tree kernel is used to obtain syntactic information, which is defined as follows:

$$K_{tree}(T_1, T_2) = \sum_{n_1 \in N_1, n_2 \in N_2} \Delta(n_1, n_2) \tag{1}$$

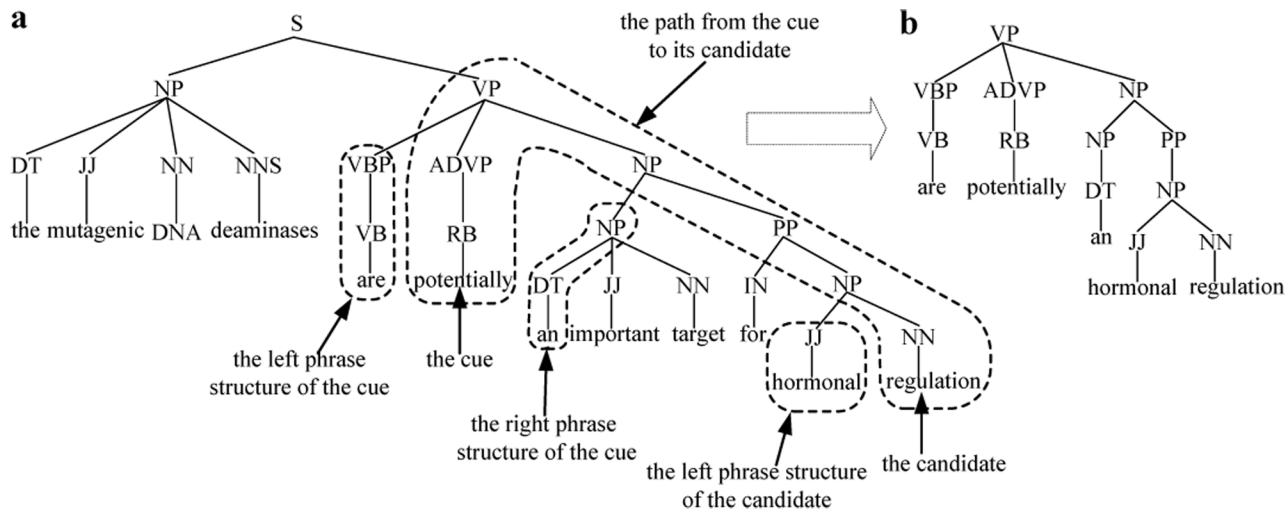
where  $N_1$  and  $N_2$  are the sets of all nodes in trees  $T_1$  and  $T_2$ , and  $\Delta(n_1, n_2)$  evaluates the number of the common sub-trees rooted at  $n_1$  and  $n_2$ , which can be computed by the following recursive rules:

1. if the productions at  $n_1$  and  $n_2$  are different, then  $\Delta(n_1, n_2) = 0$ ;
2. else if both  $n_1$  and  $n_2$  are pre-terminals, then  $\Delta(n_1, n_2) = \lambda$ ;

**Table 1. The lexical features of the L-scope candidate “regulation”.**

Feature name	$i = -3$	$i = -2$	$i = -1$	$i = 0$	$i = 1$	$i = 2$	$i = 3$
<i>CandidateToken (i)</i>	target	for	hormonal	regulation	.	null	null
<i>CandidateStem (i)</i>	target	for	hormonal	regulation	.	null	null
<i>CandidatePos (i)</i>	NN	IN	JJ	NN	.	null	null
<i>CandidateChunk (i)</i>	I-NP	B-PP	B-NP	I-NP	O	null	null
<i>HedgeToken (i)</i>	DNA	deaminases	are	potentially	an	important	target
<i>HedgeStem (i)</i>	DNA	deaminas	be	potentially	an	important	target
<i>HedgePos (i)</i>	NN	NNS	VBP	RB	DT	JJ	NN
<i>HedgeChunk (i)</i>	I-NP	I-NP	B-VP	B-ADVP	B-NP	I-NP	I-NP
<i>Distance</i>					<b>6</b>		

doi:10.1371/journal.pone.0133715.t001



**Fig 6. Phrase features extraction.** (a) The phrase tree segment of sentence 1; (b) Phrase features.

doi:10.1371/journal.pone.0133715.g006

$$3. \text{ else, } \Delta(n_1, n_2) = \lambda \prod_{k=1}^{\#ch(n_1)} (1 + \Delta(ch(n_1, k), ch(n_2, k))),$$

where  $\#ch(n_1)$  is the number of children of  $n_1$  in the tree,  $ch(n, k)$  is the  $k$ -th child node of  $n$ , and  $\lambda$  ( $0 < \lambda < 1$ ) is the decay factor.

Syntactic structure information in phrase and dependency trees can be captured by the convolution tree kernel. Phrase structure usually provides the nearer syntactic information of cues. Meanwhile, dependency structure can offer the farther syntactic information between cues and scopes [22]. Both phrase and dependency structures are effective for scope detection. We consider the two syntactic structures as features.

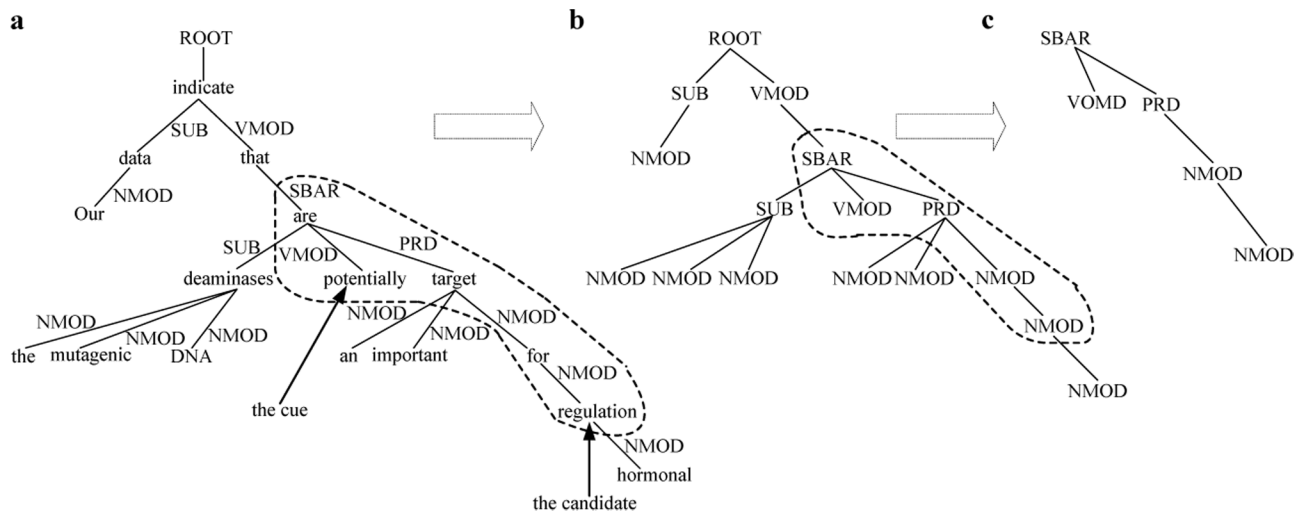
**Phrase features:** The path from the cue to its candidate boundary, including the phrase structure of the nearest neighbor tokens of both the cue and its candidate boundary, is extracted as phrase features. For the phrase tree segment of sentence 1 in Fig 6a, the phrase features from the cue “potentially” to its L-scope candidate “regulation” are shown in Fig 6b.

**Dependency features:** For the dependency tree of sentence 1 as shown in Fig 7a, tree kernel can not represent dependency relation on the arcs (e.g., “SUB” relation between node “indicate” and node “data”). To capture dependency relation, the dependency relation labels are usually used to replace the corresponding tokens on the nodes of original dependency tree as shown in Fig 7b. We define dependency features as critical path from the cue “potentially” to its candidate boundary “regulation” as shown in Fig 7c.

**The composite kernel.** To integrate the lexical and syntactic features, the composite kernel is defined by combining the above two individual kernels:

$$K_{com} = \gamma K_{tree} + (1 - \gamma) K_{poly} \tag{2}$$

where  $\gamma$  ( $0 \leq \gamma \leq 1$ ) is the composite factor. The two syntactic features are combined with the lexical features by the composite kernel respectively. We refer to the combination of the phrase and lexical features as the **phrase\_lexical** feature set, the combination of the dependency and lexical features as the **dep\_lexical** feature set. The combination of the phrase, dependency and lexical features is also implemented by the composite kernel function in Eq (3), which is called



**Fig 7. Dependency features extraction.** (a) The dependency tree of sentence 1; (b) The dependency node tree; (c) Dependency features.

doi:10.1371/journal.pone.0133715.g007

the **phrase\_dep\_lexical** feature set.

$$K_{com} = \frac{1}{2} \gamma (K_{phrase} + K_{dep}) + (1 - \gamma) K_{poly} \quad (3)$$

## Postprocessing

Hedge scope is a sequence of tokens including the hedge cue in a sentence. However, sometimes classifiers only predict F-scope or L-scope in a sentence. To guarantee that all scopes are continuous sequences of tokens, we apply the following postprocessing rules to the output of the classifiers.

1. If one token has been predicted as F-scope and one as L-scope, the sequence will start at the token predicted as F-scope, and end at the token predicted as L-scope.
2. If one token has been predicted as F-scope and more than one has been predicted as L-scope, the sequence will start at the token predicted as F-scope and end at the last token predicted as L-scope.
3. If one token has been predicted as L-scope and more than one has been predicted as F-scope, the sequence will start at the first token predicted as F-scope and end at the token predicted as L-scope.
4. If one token has been predicted as F-scope and none has been predicted as L-scope, the sequence will start at the token predicted as F-scope and end at the last token of a sentence.
5. If one token has been predicted as L-scope and none has been predicted as F-scope, the sequence will start at the hedge cue and end at the token predicted as L-scope.
6. If the hedge is passive voice, the scope will start at the subject of the hedge.
7. If the hedge is "or", the scope will start at the first token of the parallel structure conducted by the "or" and end at the last token of the parallel structure.



**Table 2. Statistical information of positives and negatives amount.**

Method	TTB			DCBS		
	#Positives	#Negatives	Ratio	#Positives	#Negatives	Ratio
<b>F-scope for training</b>	3376	41416	1:12	3376	8898	1:2.6
<b>L-scope for training</b>	3376	60962	1:18	3376	8001	1:2.4
<b>F-scope for testing</b>	1047	14942	1:14	996 (51 dropped)	3133	1:3.1
<b>L-scope for testing</b>	1047	18961	1:18	995 (52 dropped)	2779	1:2.8

doi:10.1371/journal.pone.0133715.t002

## Results

Experiments are conducted on the CoNLL-2010 Shared Task corpus. We detect the linguistic scope with golden standard cues, which provide 3376 sentences for training and 1047 sentences for testing. The evaluation of hedge scope detection is reported by F1-score for two different levels: tag-level and sentence-level. The tag-level evaluates the performance of the F-scope and L-scope classifiers respectively. The sentence-level evaluation corresponds to the exact match of scope boundaries for each cue.

### Statistics of Positives and Negatives Amount

[Table 2](#) reports statistics of the training and testing instances required for DCBS and TTB. As can be seen from [Table 2](#), the number of negatives is over ten times that of positives in TTB. By using DCBS, negatives decreases to about three times the number of positives. The training and testing instances required for DCBS are much less than that for TTB. It also shows that a few positives in the testing data are dropped due to parsing errors (51 of 1047 F-scope positives and 52 of 1047 L-scope positives are dropped).

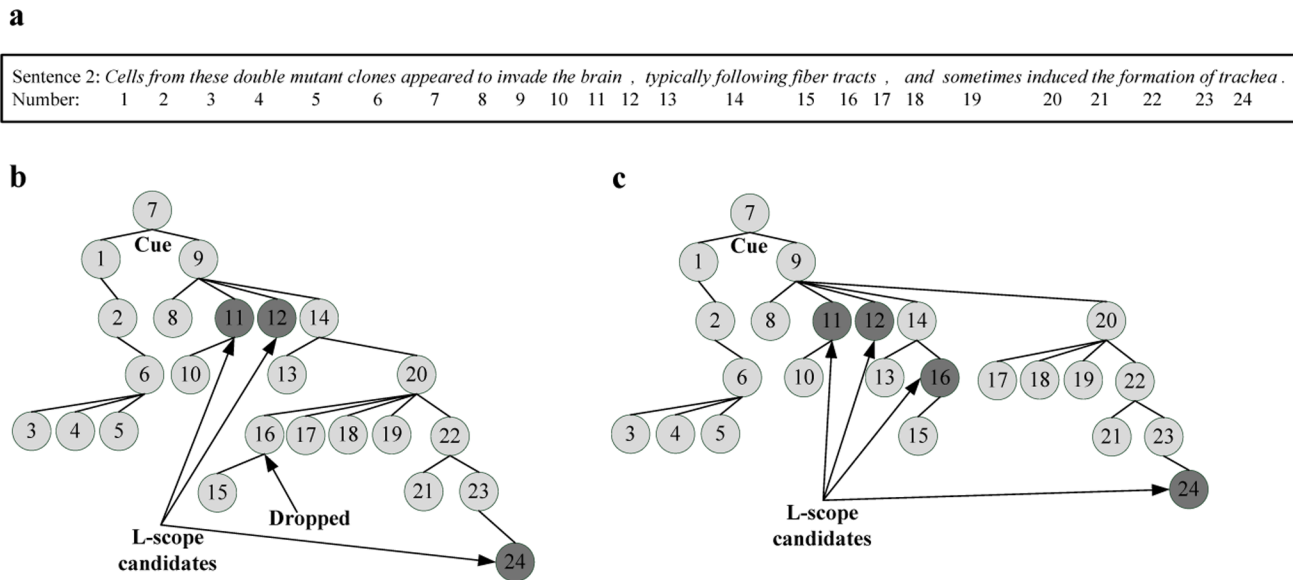
[Fig 8](#) shows an example of a sentence with the L-scope token dropped due to parsing errors. For the following example sentence 2 in the dataset, [Fig 8b](#) shows its dependency tree parsed by Gdep Parser, in which the L-scope token (node 16) is parsed incorrectly. The parsing errors lead to the dropping of a positive instance (the L-scope token “tracts”) by DCBS as shown in [Fig 8b](#). The corresponding correct dependency tree for the sentence 2 is shown in [Fig 8c](#), in which the L-scope token (node 16) is selected accurately by DCBS.

Sentence 2: < xscope > Cells from these double mutant clones < cue > appeared < /cue > to invade the brain, typically following fiber tracts < /xscope >, and sometimes induced the formation of trachea.

The dropped positives (51 F-scope positives and 52 L-scope positives) account for 4.92% of the total number of existing positives. However, the subsequent experiments show that DCBS could offset the loss of the dropped positives.

### Hedge Scope Detection Performance

**Effects of candidate boundary selection.** We compare DCBS with TTB on the tag-level and sentence-level F1-scores. Statistical significance analysis between DCBS and TTB is performed by Student *t*-tests using sentence-level F1-scores. [Fig 9](#) shows the F1-score curves with different composite factor  $\gamma$ . We vary  $\gamma$  from 0 (sole lexical features) to 1 (sole syntactic features) with an interval of 0.1. The phrase\_lexical ([Fig 9a](#)), dep\_lexical ([Fig 9b](#)) and phrase\_dep\_lexical ([Fig 9c](#)) feature sets are investigated in our experiment. From [Fig 9](#), we can see that:



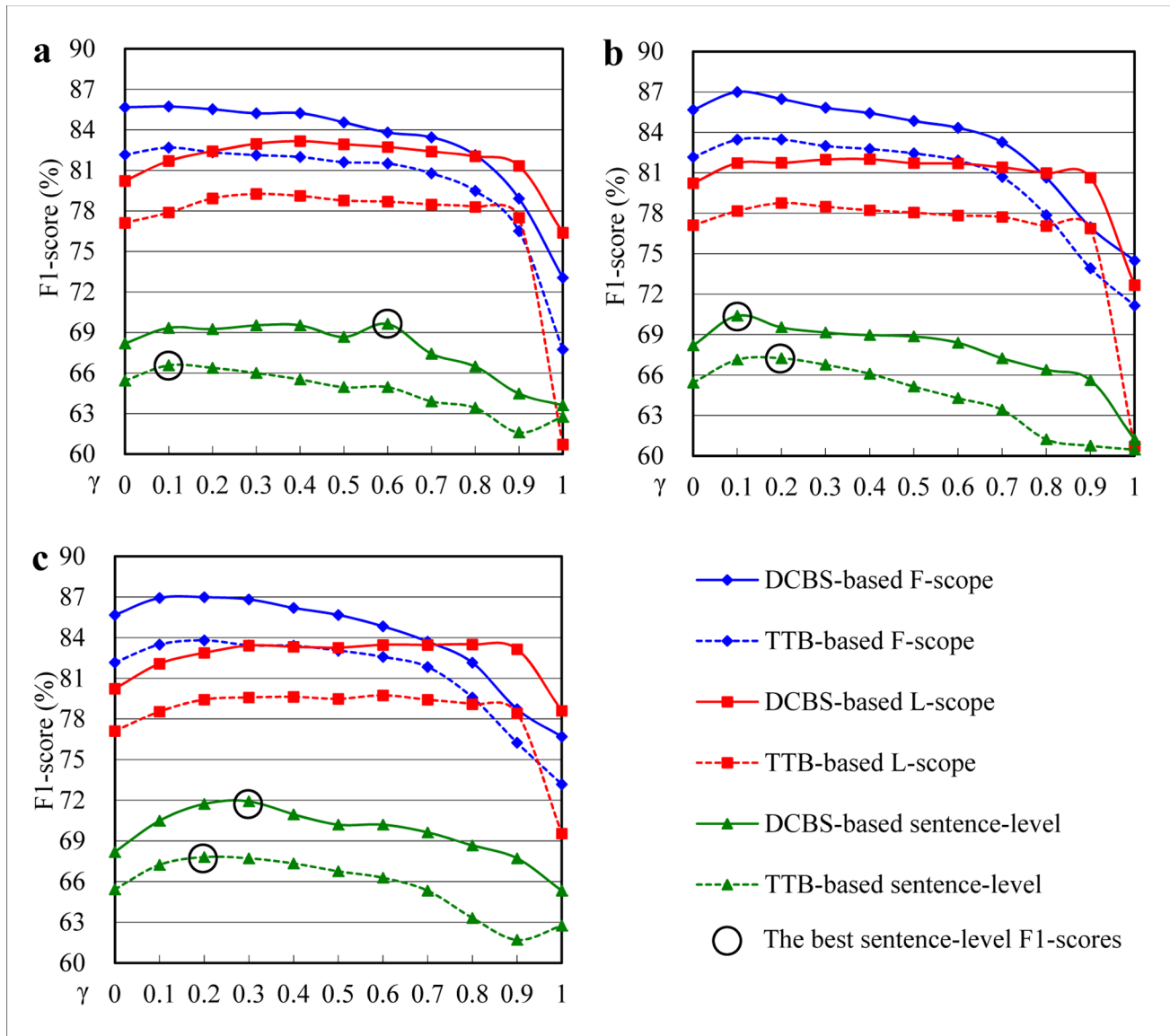
**Fig 8. An example of a sentence with the L-scope token dropped due to parsing errors.** (a) The sequence number of tokens in sentence 2; (b) DCBS drops the L-scope node 16 due to incorrect dependency tree; (c) DCBS selects the L-scope node 16 accurately based on correct dependency tree.

doi:10.1371/journal.pone.0133715.g008

1. For the three feature sets, DCBS steadily outperforms TTB in terms of tag-level and sentence-level with  $\gamma$  varying from 0 to 1, though a few positives are dropped due to parsing errors.
2. All  $p$ -values of the three feature sets are less than 0.01 when compared sentence-level F1-scores of DCBS with that of TTB. Statistical analysis shows significant differences between DCBS and TTB.
3. Both the polynomial kernel with lexical features ( $\gamma = 0$ ) and the tree kernel with syntactic features ( $\gamma = 1$ ) could get acceptable results. The sole polynomial kernel outperforms the sole tree kernel. The composite kernel combining lexical and syntactic features with appropriate composite factor  $\gamma$  could achieve higher F1-score than either one of them.
4. The performance of F-scope classifiers is usually better than that of L-scope classifiers. The main reason is that the distance of L-scope to its cue is longer than that of F-scope in a sentence. The longer the distance from the scope boundary to its cue is, the harder the scope detection is.

The best sentence-level F1-scores of DCBS and TTB with the three feature sets (marked with solid-circle in Fig 9) are summarized in Table 3. Even though the number of the training instances used for DCBS is less than that for TTB and the used feature sets are the same, the best F1-scores are improved using DCBS. Furthermore, the time required for training and testing is significantly reduced. For example, when using the phrase\_dep\_lexical feature set, DCBS achieves the best F1-score 71.92% under the condition  $\gamma = 0.3$ , which is 4.11% higher than that of TTB under the condition  $\gamma = 0.2$ . Meanwhile, the time of training and testing is reduced (19.81min.  $\rightarrow$  2.17min., 3.41min.  $\rightarrow$  0.47min.) by applying DCBS.

**Effects of syntactic features.** To obtain the better effects of the syntactic features, the F1-score curves of sentence-level with the three feature sets depicted in Fig 9 are extracted and



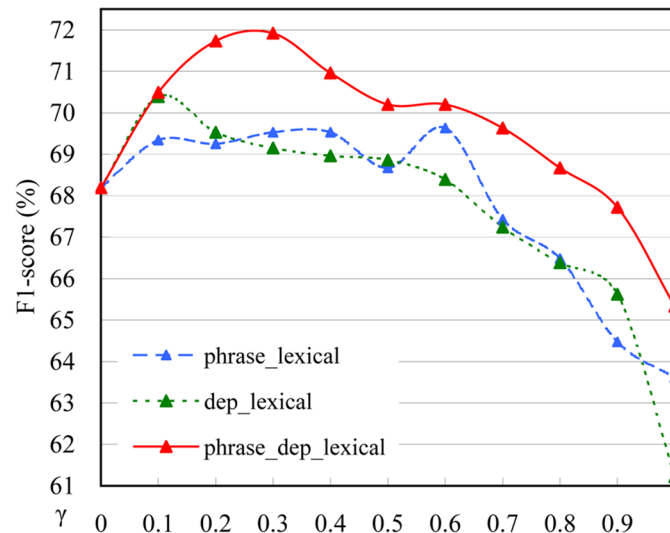
**Fig 9. DCBS vs. TTB with the different composite factor  $\gamma$ .** (a) The phrase\_lexical feature set ( $p = 8.05e-04$  for sentence-level F1-scores); (b) The dep\_lexical feature set ( $p = 6.92e-03$  for sentence-level F1-scores); (c) The phrase\_dep\_lexical feature set ( $p = 2.06e-04$  for sentence-level F1-scores). All  $p$ -values  $< 0.01$  for sentence-level F1-scores.

doi:10.1371/journal.pone.0133715.g009

**Table 3. Performance comparison of the best DCBS and TTB systems with the different feature sets.**

Feature set	Phrase_lexical		Dep_lexical		Phrase_dep_lexical	
	TTB	DCBS	TTB	DCBS	TTB	DCBS
<b>F1-score(%)</b>	66.57( $\gamma = 0.1$ )	69.63( $\gamma = 0.6$ )	67.24( $\gamma = 0.2$ )	70.39( $\gamma = 0.1$ )	67.81( $\gamma = 0.2$ )	71.92( $\gamma = 0.3$ )
<b>Training time</b>	18.92min.	2.43min.	7.59min.	1.23min.	19.81min.	2.17min.
<b>Testing time</b>	3.08min.	0.46min.	1.16min.	0.19min.	3.41min.	0.47min.

doi:10.1371/journal.pone.0133715.t003



**Fig 10. Effects of different syntactic features in DCBS.** Student *t*-test for phrase\_dep\_lexical vs. phrase\_lexical results in a *p*-value of 2.14e-02; student *t*-test for phrase\_dep\_lexical vs. dep\_lexical results in a *p*-value of 1.44e-02 (all *p*-values < 0.05).

doi:10.1371/journal.pone.0133715.g010

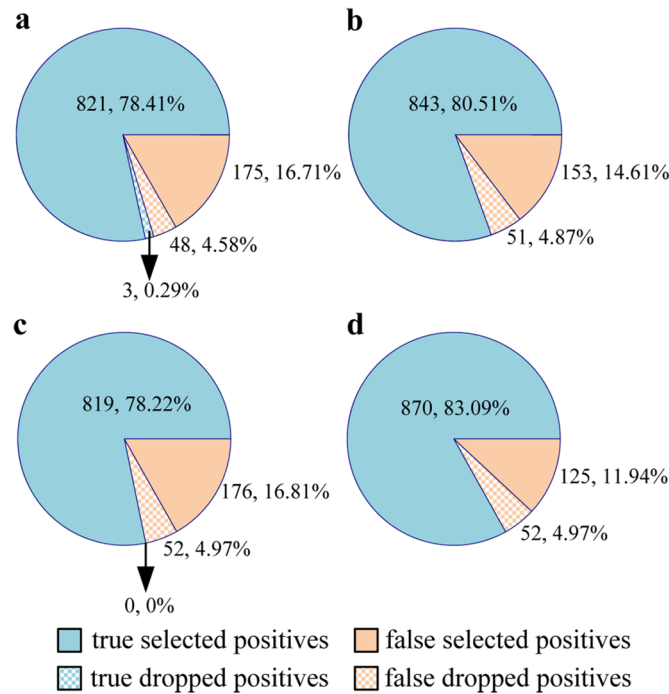
shown in Fig 10. From Fig 10 we can see that the trends of the three curves are similar. All starts from the initial F1-score of the sole lexical features (68.19%), and then increases to the individual highest F1-score, finally falls below the initial F1-score. Generally, the results with the syntactic features are better than those without them. Both the phrase and dependency features are effective in hedge scope detection and the combination of the two syntactic features outperforms either one of them obviously. The addition of the two syntactic features can improve F1-score from 68.19% to 71.92% (3.73% increases). *P*-values indicate that adding dependency and phrase structured features is statistically significant.

Table 4 compares the performance of the structured syntactic features with the flat syntactic features. Compared with the sole lexical features (68.19%), flat syntactic features improve the performance slightly. The structured phrase and dependency features outperform the corresponding flat features by 1.44% and 2.10% in F1-score, respectively. Moreover, the combination of the two structured syntactic features significantly improves F1-score by 3.53% (from 68.39% to 71.92%). *P*-values by a 10-fold cross-validation on the training data set clearly show significant differences between structured syntactic features and flat syntactic features (all *p*-values < 0.01).

**Table 4. Structured syntactic features vs. flat syntactic features.**

Feature set	F1-score with flat features	F1-score with structured features	<i>p</i> -values
Phrase_lexical	68.19%	69.63% (1.44 ↑)	4.45e-03
Dep_lexical	68.29%	70.39% (2.10 ↑)	9.02e-09
Phrase_dep_lexical	68.39%	71.92% (3.53 ↑)	2.80e-12

doi:10.1371/journal.pone.0133715.t004



**Fig 11. Summary of the tag-level classification results of DCBS and TTB.** (a) F-scope classification results with TTB; (b) F-scope classification results with DCBS; (c) L-scope classification results with TTB; (d) L-scope classification results with DCBS.

doi:10.1371/journal.pone.0133715.g011

### Summary of the Classification Results of DCBS and TTB

The detailed F-scope and L-scope classification results of DCBS ( $\gamma = 0.3$ ) and TTB ( $\gamma = 0.2$ ) with the phrase\_dep\_lexical feature set are summarized in Fig 11. The 1047 positives in testing data are divided into two categories: **selected positives** and **dropped positives** by DCBS. For 51 dropped positives in F-scope classification, the true positives classified by TTB are only three as shown in Fig 11a. Meanwhile for 52 dropped positives in L-scope classification, no positive is correctly classified by TTB as shown in Fig 11c. On the other hand, for the selected positives, DCBS-based F-scope classifier detects 22 true positives (Fig 11a and 11b) and DCBS-based L-scope classifier identifies 51 true positives (Fig 11c and 11d) more than TTB-based classifiers. These results indicate that DCBS could offset the loss of the dropped positives.

### Comparison with Related Work

Table 5 compares the results of our method with the state-of-the-art results on golden standard cues of the CoNLL-2010 Shared Task corpus. From Table 4 we can see that our method

**Table 5. Comparison with the related work.**

System	Our system	Zhou et al. [9]	Velldal et al. [12]
<b>F1-score (%)</b>	71.92	70.28	69.60

doi:10.1371/journal.pone.0133715.t005

achieves the best performance with an F1-score of 71.92%. Zhou et al. [9] and Velldal et al. [12] also employed dependency and phrase parsing trees. Zhou et al. [9] employed decision tree to construct the dependency constraint set and phrase constraint set based on dependency structure and phrase structure respectively, which were used to generate the syntactic constraint features for hedge scope detection. They reached 70.28% F1-score. Velldal et al. [12] combined a rule-based approach over dependency structure and a data-driven approach over phrase structure. They obtained an F1-score of 69.60%. Our method simply combines the structured dependency and phrase features, and outperforms Zhou et al. [9] and Velldal et al. [12] by 1.64% and 2.32% in F1-score, respectively. The performance improvement is obvious. The superiority of our system benefits from our candidate boundary selection method and the structured syntactic representation method.

## Discussion

We present a dependency-based candidate boundary selection method for hedge scope detection. Experimental results show that our method outperforms most of the state-of-the-art systems. The analysis is as follows.

### Effectiveness of candidate boundary selection

The proposed DCBS is effective for hedge scope detection. The absolute superiority of DCBS for scope detection benefits from the removal of a large number of confusing negatives generated by TTB. From the above experimental results, we can conclude that DCBS has the following advantages over TTB.

1. Simple and efficient: DCBS is simple, which selects candidates only based on dependency tree. Moreover, the method could reduce the training and testing cost significantly by decreasing the number of candidates.
2. Balance instance bias: TTB generates a large number of negatives and results in instance bias for classifiers. DCBS can filter out the most likely non-boundary tokens of a given cue and mitigate potential imbalance of positives and negatives.
3. Enhance the discriminability of instances: TTB takes the boundary tokens as positives and the other tokens including the neighbors of boundary tokens as negatives. As adjacent tokens have similar structure and contextual information, the boundaries and their neighbors are extremely difficult to distinguish for classifiers. DCBS can eliminate most of the neighbors of boundaries to improve the classification performance.

### Effectiveness of syntactic features

The composite kernel consisting of the polynomial kernel and the tree kernel is employed to integrate lexical and syntactic information. Lexical features contain semantic and contextual information. Syntactic features capture the structured information. Therefore, syntactic features and lexical features are complementary for hedge scope detection, and their combination could improve the performance further.

Phrase syntactic features represent constituent information of neighbor words, and it is well suitable to capturing the local syntactic information. Dependency syntactic features are compact and can capture global syntactic information between cues and scopes. Thus, the phrase and dependency syntactic features are complementary for hedge scope detection. The combination of the two features could obviously outperform either one of them.

In addition, the structured syntactic features with the tree kernel outperform the flat syntactic features with the polynomial kernel. The main reason is that the polynomial kernel cannot capture the structured syntactic features, while the tree kernel can effectively capture the structured features by counting the number of the common subtrees.

## Conclusions

We present a dependency-based candidate boundary selection method for hedge scope detection which achieves 71.92% F1-score on the CoNLL-2010 Shared Task corpus. Our method is designed along with the heuristics that two tokens with dependency relation should be both in the scope and selects the further token away from the cue as candidate boundary. DCBS can eliminate most of confusing negatives generated by TTB and therefore enhances the discriminability of candidate instances. Our method outperforms previous state-of-the-art methods with respect to performance and efficiency. In addition, the composite kernel consisting of the polynomial kernel and the tree kernel is employed to integrate lexical and syntactic information. The structured phrase and dependency features both outperform the corresponding flat features and the combination of structured phrase and dependency features achieves even better results. To the best of our knowledge, our method obtains the best published results so far on the CoNLL-2010 Shared Task corpus for scope detection.

Besides lexical and syntactic information, semantic information of words plays an important role in hedge scope detection. How to extract the semantic knowledge of words, especially how to calculate the semantic similarity between two cues and combine semantic information with our lexical and syntactic information will be studied in our future work.

## Acknowledgments

We wish to thank the anonymous reviewers for their valuable comments. This research is supported by National Natural Science Foundation of China (Grant No. 61272375 and 61173100).

## Author Contributions

Conceived and designed the experiments: HZ HD. Performed the experiments: HZ HD. Analyzed the data: HZ HD DH. Contributed reagents/materials/analysis tools: HZ DH. Wrote the paper: HZ HD MZ. Manuscript revision: HZ HD MZ.

## References

1. Szarvas G, Vincze V, Farkas R, Csirik J (2008) The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. *BioNLP 2008: Current Trends in Biomedical Natural Language Processing*. Columbus, Ohio, USA: 38–45.
2. Farkas R, Vincze V, Móra G, Csirik J, Szarvas G (2010) The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*. Uppsala, Sweden: 1–12.
3. Szarvas G, Vincze V, Farkas R, Móra G, Gurevych I (2012) Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics* 38: 335–367. doi: [10.1162/COLI\\_a\\_00098](https://doi.org/10.1162/COLI_a_00098)
4. Su Q, Lou HQ, Liu PY (2013) Hedge Detection with Latent Features. *Chinese Lexical Semantics*: 436–441.
5. Wei ZY, Chen JW, Gao W, Li BY, Zhou LJ, et al. (2013) An Empirical Study on Uncertainty Identification in Social Media Context. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria: 58–62.
6. Özgür A, Radev D R (2009) Detecting Speculations and their Scopes in Scientific Text. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: 1398–1407.
7. Øvrelid L, Velldal E, Oepen S (2010) Syntactic Scope Resolution in Uncertainty Analysis. *Proceedings of the 23rd international conference on computational linguistics*. Beijing: 1379–1387.

8. Apostolova E, Tomuro N, Demner-Fushman D (2011) Automatic Extraction of Lexico-Syntactic Patterns for Detection of Negation and Speculation Scopes. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Portland, Oregon: 283–287.
9. Zhou HW, Yang H, Huang DG, Li Y, Li LS (2013) Hedge Scope Detection Based on Syntactic Structural Constraints. *Journal of Chinese Information Processing* 27: 137–141.
10. Morante R, Daelemans W (2009) Learning the scope of hedge cues in biomedical texts. *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*. Boulder, Colorado: 28–36.
11. Morante R, Van Asch V, Daelemans W (2010) Memory-Based Resolution of In-Sentence Scopes of Hedge Cues. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*. Uppsala, Sweden: 40–47.
12. Vellidal E, Øvrelid L, Read J, Oepen S (2012) Speculation and Negation: Rules, Rankers, and the Role of Syntax. *Computational Linguistics* 38: 369–410. doi: [10.1162/COLI\\_a\\_00126](https://doi.org/10.1162/COLI_a_00126)
13. Zhu QM, Li JH, Wang HL, Zhou GD (2010) A Unified Framework for Scope Learning via Simplified Shallow Semantic Parsing. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. MIT, Massachusetts, USA: 714–724.
14. Collins M, Duffy N (2002) Convolution Kernels for Natural Language. *Advances in neural information processing systems* 14. Cambridge, MA: MIT Press, 625–632.
15. Zhang M, Zhang J, Su J (2006) Exploring syntactic features for relation extraction using a convolution tree kernel. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. New York: 288–295.
16. Plank B, Moschitti A (2013) Embedding Semantic Similarity in Tree Kernels for Domain Adaptation of Relation Extraction. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria: 1498–1507.
17. Moschitti A, Pighin D, Basili R (2006) Semantic role labeling via tree kernel joint inference. *Proceedings of the Tenth Conference on Computational Natural Language Learning*. New York: 61–68.
18. Liu H, Hunter L, Kešelj V, Verspoor K (2013) Approximate Subgraph Matching-Based Literature Mining for Biomedical Events and Relations. *PLoS ONE* 8(4): e60954. doi: [10.1371/journal.pone.0060954](https://doi.org/10.1371/journal.pone.0060954) PMID: [23613763](https://pubmed.ncbi.nlm.nih.gov/23613763/)
19. He LN, Yang ZH, Zhao ZH, Lin HF, Li YP (2013) Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach. *PLoS One* 8: e65814. doi: [10.1371/journal.pone.0065814](https://doi.org/10.1371/journal.pone.0065814) PMID: [23785452](https://pubmed.ncbi.nlm.nih.gov/23785452/)
20. Bokharaeian B, Díaz A (2013) NIL UCM: Extracting Drug-Drug interactions from text through combination of sequence and tree kernels. *Second Joint Conference on Lexical and Computational Semantics*. Atlanta, Georgia, USA: 644–650.
21. Zhou HW, Huang DG, Li XY, Yang YS (2011) Combining Structured and Flat Features by a Composite Kernel to Detect Hedges Scope in Biological Texts. *Chinese Journal of Electronics* 20: 476–482.
22. Zou BW, Zhou GD, Zhu QM (2013) Tree Kernel-based Negation and Speculation Scope Detection with Structured Syntactic Parse Features. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: 968–976.