

Tree Kernel-based Negation and Speculation Scope Detection with Structured Syntactic Parse Features

Bowei Zou Guodong Zhou Qiaoming Zhu*

Natural Language Processing Lab, School of Computer Science and Technology
Soochow University, Suzhou, 215006, China

zoubowei@gmail.com, {gdzhou, qmzhu}@suda.edu.cn

Abstract

Scope detection is a key task in information extraction. This paper proposes a new approach for tree kernel-based scope detection by using the structured syntactic parse information. In addition, we have explored the way of selecting compatible features for different part-of-speech cues. Experiments on the BioScope corpus show that both constituent and dependency structured syntactic parse features have the advantage in capturing the potential relationships between cues and their scopes. Compared with the state of the art scope detection systems, our system achieves substantial improvement.

1 Introduction

The task of scope detection is to detect the linguistic scope dominated by a specific cue. Current researches in this field focus on two semantic aspects: negation and speculation. The negative scope detection is to detect the linguistic scope which is repudiated by a negative word (viz., negative cue, e.g., “*not*”). In other side, the speculative scope detection is to detect the uncertain part in a sentence corresponding to the speculative word (viz., speculative cue, e.g., “*seems*”). See the sentence 1) below, the negative cue “*not*” dominates the scope of “*not expensive*”. Similarly, the speculative cue “*possible*” in sentence 2) dominates the uncertain scope “*the possible future scenarios*”.

- 1) *The chair is [not expensive] but comfortable.*
- 2) *Considering all that we have seen, what are now [the possible future scenarios]?*

The negative and speculative scope detection task consists of two basic stages. The first one is to identify the sentences involving negative or speculative meaning. The second stage is to detect the linguistic scope of the cue in sentences (Velldal et al, 2012). In this paper, we focus on the second stage. That is, by given golden cues, we detect their linguistic scopes.

We propose a tree kernel-based negation and speculation scope detection with structured syntactic parse features. In detail, we regard the scope detection task as a binary classification issue, which is to classify the tokens in a sentence as being inside or outside the scope. In the basic framework, we focus on the analysis and application of structured syntactic parse features as follows:

Both constituent and dependency syntactic features have been proved to be effective in scope detection (Özgür et al, 2009; Øvrelid et al, 2010). However, these flat features are hardly to reflect the information implicit in syntactic parse tree structures. Our intuition is that the segments of the syntactic parse tree around a negative or speculative cue is effective for scope detection. The related structures normally underlay the indirect clues to identify the relations between cues and their scopes, e.g., in sentence 1), “*but something*”, as a frequently co-occurred syntactic structure with “*not something*”, is an effective clue to determine the linguistic scope of “*not*”.

The tree kernel classifier (Moschitti, 2006) based on support vector machines uses a kernel function between two trees, affording a comparison between their substructures. Therefore, a tree kernel-based scope detection approach with structured syntactic parse tree is employed. The tree

* Corresponding author

kernel has been already proved to be effective in semantic role labeling (Che et al, 2006) and relation extraction (Zhou et al, 2007).

In addition, the empirical observation shows that features have imbalanced efficiency for scope classification, which is normally affected by the part-of-speech (abbr., POS) of cues. Hence, we build the discriminative classifiers for each kind of POS of cues, then explore and select the most compatible features for them.

We construct a scope detection system by using the structured syntactic parse features based tree kernel classification. Compared with the state of the art scope detection systems, our system achieves the performance of accuracy 76.90% on negation and 84.21% on speculation (on Abstracts sub-corpus). Additionally, we test our system on different sub-corpus (Clinical Reports and Full Papers). The results show that our approach has better cross-domain performance.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the corpus and corresponding usage in our experiments. Section 4 describes our approach and the experiments are presented in Section 5. Finally, there is a conclusion in Section 6.

2 Related Work

Most of the previous studies on negation and speculation scope detection task can be divided into two main aspects: the heuristic rule based methods and the machine learning based methods. We respectively introduce the aspects in below.

2.1 Heuristic Rule based Methods

The initial studies for scope detection are to compile effective heuristic rules (Chapman et al, 2001; Goldin et al, 2003). Recently, the heuristic rule based methods have further involved the syntactic features.

Huang et al (2007) implemented a hybrid approach to automated negation scope detection. They combined the regular expression matching with grammatical parsing: negations are classified on the basis of syntactic categories and located in parse trees. Their hybrid approach is able to identify negated concepts in radiology reports even when they are located at some distance from the negative term.

Özgür et al (2009) hypothesized that the scope of a speculation cue can be characterized by its part-of-speech and the syntactic structure of the sentence and developed rules to map the scope of a cue to the nodes in the syntactic parse tree. By given golden speculation cues, their rule-based method achieves the accuracies of 79.89% and 61.13% on the Abstracts and the Full-Papers sub-corpus, respectively.

Øvrelid et al (2010) constructed a small set of heuristic rules which define the scope for each cue. In developing these rules, they made use of the information provided by the guidelines for scope annotation in the BioScope corpus, combined with manual inspection of the training data in order to further generalize over the phenomena discussed by Vincze et al (2008) and work out interactions of constructions for various types of cues.

Apostolova et al (2011) presented a linguistically motivated rule-based system for the detection of negation and speculation scopes that performs on par with state-of-the-art machine learning systems. The rules are automatically extracted from the BioScope corpus and encode lexico-syntactic patterns in a user-friendly format. While their system was developed and tested using a biomedical corpus, the rule extraction mechanism is not domain-specific.

The heuristic rule based methods have bad robustness in detecting scopes crossing different meaning aspects (e.g., negative vs. speculative) and crossing different linguistic resources (e.g., Technical Papers vs. Clinical Reports).

2.2 Machine Learning based Methods

The machine learning based methods have been ignored until the release of the BioScope corpus (Szarvas et al, 2008), where the large-scale data of manually annotated cues and corresponding scopes can support machine learning well.

Morante et al (2008) formulated scope detection as a chunk classification problem. It is worth noting that they also proposed an effective proper post-processing approach to ensure the consecutiveness of scope. Then, for further improving the scope detection, Morante et al (2009a) applied a meta-learner that uses the predictions of the three classifiers (TiMBL/SVM/CRF) to predict the scope.

For the competitive task in CoNLL'2010 (Farkas et al, 2010), Morante et al (2010) used a

memory-based classifier based on the k-nearest neighbor rule to determine if a token is the first token in a scope sequence, the last, or neither. Therefore, in order to guarantee that all scopes are continuous sequences of tokens they apply a first post-processing step that builds the sequence of scope.

The existing machine learning based approaches substantially improve the robustness of scope detection, and have nearly 80% accuracy. However, the approaches ignore the availability of the structured syntactic parse information. This information involves more clues which can well reflect the relations between cues and scopes. Sánchez et al (2010) employed a tree kernel based classifier with CCG structures to identify speculative sentences on Wikipedia dataset. However, in Sánchez’s approach, not all sentences are covered by the classifier.

3 Corpus

We have employed the BioScope corpus (Szarvas et al, 2008; Vincze et al, 2008)¹, an open resource from the biomedical domain, as the benchmark corpus. The corpus contains annotations at the token level for negative and speculative cues and at the sentence level for their linguistic scope (as shown in Figure 1).

```
<sentence id="S26.8"> These findings <xcope id="X26.8.2">
<cue type="speculation" ref="X26.8.2"> indicate that </cue>
<xcope id="X26.8.1"> corticosteroid resistance in bronchial
asthma <cue type="negation" ref="X26.8.1"> can not </cue>
be explained by abnormalities in corticosteroid receptor char-
acteristics </xcope></xcope> . </sentence>
```

(Note: <Sentence> denotes one sentence and the tag “id” denotes its serial number; <xcope> denotes the scope of a cue; <cue> denotes the cue, the tag “type” denotes the specific kind of cues and the tag “ref” is the cue’s serial number.)

Figure 1. An annotated sentence in BioScope.

The BioScope corpus consists of three sub-corpora: biological Full Papers from FlyBase and BMC Bioinformatics, biological paper Abstracts from the GENIA corpus (Collier et al, 1999), and Clinical Reports. Among them, the Full Papers sub-corpus and the Abstracts sub-corpus come from the same genre. In comparison, the Clinical Reports sub-corpus consists of clinical radiology reports with short sentences.

In our experiments, if there is more than one cue in a sentence, we treat them as different cue and scope (two independent instances). The statistical data for our corpus is presented in Table 1 in below.

The average length of sentences in the negation portion is almost as long as that in speculation, while the average length of scope in negation is shorter than that in speculation. In addition, the length of sentence and scope in both Abstracts and Full Papers sub-corpora is comparative. But in Clinical Reports sub-corpus, it is shorter than that in Abstracts and Full Papers. Thus, looking for the effective features in short sentences is especially important for improving the robustness for scope detection.

		Abstract	Paper	Clinical
Negation	Sentences	1594	336	441
	Words	46849	10246	3613
	Scopes	1667	359	442
	Av. Len Sentence	29.39	30.49	8.19
	Av. Len Scope	9.62	9.36	5.28
Speculation	Sentences	2084	519	854
	Words	62449	16248	10241
	Scopes	2693	682	1137
	Av. Len Sentence	29.97	31.31	11.99
	Av. Len Scope	17.24	15.58	6.99

(Note: “Av. Len” stands for average length.)

Table 1. Statistics for our corpus in BioScope.

4 Methodology

We regard the scope detection task as a binary classification problem, which is to classify each token in sentence as being the element of the scope or not. Under this framework, we describe the flat syntactic features and employ them in our benchmark system. Then, we propose a tree kernel-based scope detection approach using the structured syntactic parse features. Finally, we construct the discriminative classifier for each kind of POS of cues, and select the most compatible features for each classifier.

4.1 Flat Syntactic Features

In our benchmark classification system, the features relevant to the cues or tokens are selected. Then, we have explored the constituent and dependency syntactic features for scope detection. These features are all flat ones which reflect the characteristic of tokens, cues, scopes, and the relation between them.

¹ <http://www.inf.u-szeged.hu/rgai/bioscope>

Basic Features: Table 2 shows the basic features which directly relate to the characteristic of cues or tokens in our basic classification.

Feature	Remark
B1	Cue.
B2	Candidate token.
B3	Part-of-speech of candidate token.
B4	Left token of candidate token.
B5	Right token of candidate token.
B6	Positional relation between cue and token.

Table 2. Basic features.

Constituent Syntactic Features: For improving the basic classification, we employ 10 constituent features belonging to two aspects. On the one hand, we regard the linguistic information of the neighbor locating around the candidate tokens as the coherent features (CS1~CS6 in Table 3). These features are used for detecting the close cooperation of a candidate token co-occurring with its neighbors in a scope. On the other hand, we regard the linguistic characteristics of the candidate tokens themselves in a syntactic tree as the inherent features (CS7~CS10 in Table 3). These features are used for determining whether the token has the direct relationship with the cue or not.

Features	Remarks
CS1	POS of left token.
CS2	POS of right token.
CS3	Syntactic category of left token.
CS4	Syntactic category of right token.
CS5	Syntactic path from left token to the cue.
CS6	Syntactic path from right token to the cue.
CS7	Syntactic category of the token.
CS8	Syntactic path from the token to the cue.
CS9	Whether the syntactic category of the token is the ancestor of the cue.
CS10	Whether the syntactic category of the cue is the ancestor of the token.

Table 3. Constituent syntactic features.

Features	Remarks
DS1	Dependency direction (“head” or “dependent”).
DS2	Dependency syntactic path from the token to cue.
DS3	The kind of dependency relation between the token and cue.
DS4	Whether the token is the ancestor of the cue.
DS5	Whether the cue is the ancestor of the token.

Table 4. Dependency syntactic features.

Dependency Syntactic Features: For the effectiveness to obtain the syntactic information far apart from cues, we use 5 dependency syntactic features which emphasize the dominant relation-

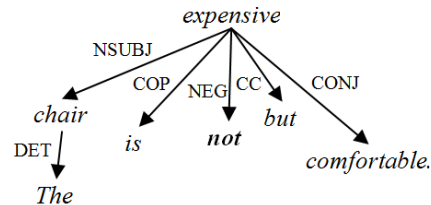
ship between cues and tokens by dependency arcs as shown in Table 4.

The features in Table 2, 3, and 4 have imbalanced classification for the scope classification. Therefore, we adopt the greedy feature selection algorithm as described in Jiang et al (2006) to pick up positive features incrementally according to their contributions. The algorithm repeatedly selects one feature each time, which contributes most, and stops when adding any of the remaining features fails to improve the performance.

4.2 Structured Syntactic Features

Syntactic trees involve not only the direct bridge (e.g., syntactic path) between cue and its scope but also the related structures to support the bridge (e.g., sub-tree). The related structures normally involve implicit clues which underlay the relation between cue and its scope. Therefore, we use the constituent and dependency syntactic structures as the supplementary features to further improve the benchmark system.

Furthermore, we employ the tree kernel-based classifier to capture the structured information both in constituent and dependency parsing trees. The results of the constituent syntactic parser are typical trees which always consist of the syntactic category nodes and the terminal nodes. Thus, the constituent syntactic tree structures could be used in tree kernel-based classifier directly, but not for the dependency syntactic tree structures. As Figure 2 shows, in sentence “*The chair is not expensive but comfortable.*” the tree kernels cannot represent the relations on the arcs (e.g., “*CONJ*” between “*expensive*” and “*comfortable*”). It is hard to use the relations between tokens and cues in tree kernels.



Structure of Dependency Tree

Cue: *not*

Scope: *not expensive*

Figure 2: The dependency tree of sentence “*The chair is not expensive but comfortable.*”

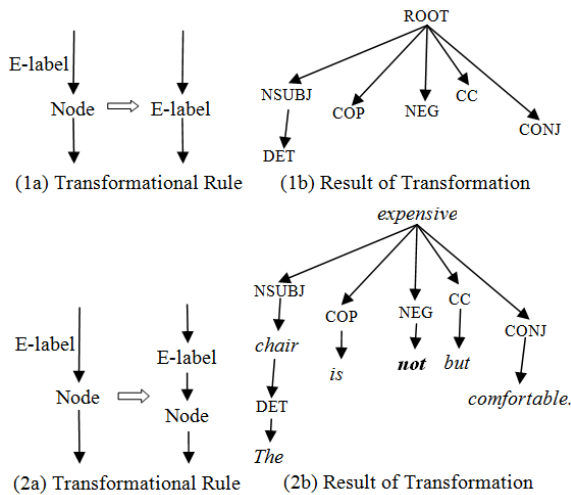


Figure 3. Two transformational rules.

To solve the problem, we transform the dependency tree into other two forms capable of being used directly as the compatible features in tree-kernel based classification. The transformational rules are described as below:

(1) Extracting the dependency relations to generate a tree of pure relations (named dependency relational frame), where the tokens on the nodes of original dependency tree are ignored and only the relation labels are used. E.g., the tokens “chair”, “is”, etc in Figure 2 are all deleted and replaced by the corresponding relation labels. E.g., “NSUBJ”, “COP”, etc are used as nodes in the dependency relational frame, see (1a) & (1b) in Figure 3.

(2) Inserting the tokens which have been deleted in step (1) into the dependency relational frame and making them follow and link with their original dependency relations. E.g., the tokens “chair”, “is”, etc are added below the nodes “NSUBJ”, “COP”, etc, see (2a) & (2b) in Figure 3.

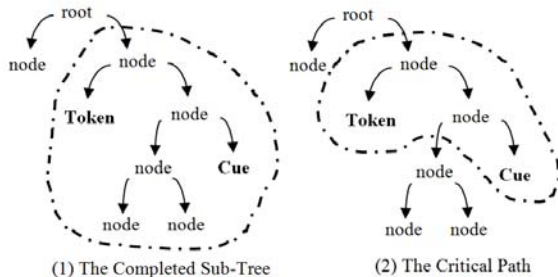


Figure 4. Two transformations for tree-kernel.

Within the constituent and dependency syntactic trees, we have employed both the Completed Sub-Tree and the Critical Path as the syntactic structure

features for our classification. The former is a minimum sub-tree that involves the cues and the tokens, while the latter is the path from the cues to the tokens in the completed tree containing the primary structural information. Figure 4 shows them.

4.3 Part-of-Speech Based Classification Optimization

Motivating in part by the rule-based approach of Özgür et al (2009), we infer that features have imbalanced efficiency for scope classification, normally affected by the part-of-speech (POS) of cues.

POS of Cues	Number	POS of Cues	Number
CC	157	VB	31
IN	115	VBD	131
JJ	238	VBG	225
MD	733	VBN	112
NN	43	VBP	561
RB	137	VBZ	207

Table 5. Distribution of different POSs of speculative cues in Abstracts sub-corpus.

Table 5 shows the distribution for different POSs of cues in the Abstracts sub-corpus of BioScope for speculation detection task. The cues of different POS usually undertake different syntactic roles. Thus, there are different characteristics in triggering linguistic scopes. See the two examples below:

- 3) *TCF-1 contained a single DNA box in the [putative mammalian sex-determining gene SRY].*
- 4) *The circadian rhythm of plasma cortisol [either disappeared or was inverted].*

The speculative cue “putative” in sentence 3) is an adjective. The corresponding scope is its modificatory structure (“putative mammalian sex-determining gene SRY”). In sentence 4), “either...or...” is a conjunction speculation cue. Its scope is the two connected components (“either disappeared or was inverted”). Thus, the effective features for the adjectival cue are normally the dependency features, e.g., the features of DS1 and DS5 in Table 4, while the features for the conjunction cue are normally the constituent information, e.g., the features of CS9 in Table 3.

In Table 5, considering the different function of verb voice, we cannot combine the “VB(*)” POS. For instance, the POS of “suggest” in sentence 5) is “VBP” (the verb present tense). The corresponding scope does not involve the sentence subject.

The POS of “*suggested*” in sentence 6) is “VBN” (the past participle). The scope involves the subject “*An age-related decrease*”.

- 5) *These results [suggest that the genes might be involved in terminal granulocyte differentiation].*
- 6) *[An age-related decrease was suggested between subjects younger than 20 years].*

As a result, we have built a discriminative classifier for each kind of POS of cues, and then explored and selected the most compatible features for each classifier.

5 Experiments and Results

5.1 Experimental Setting

Considering the effectiveness of different features, we have split the Abstracts sub-corpus into 5 equal parts, within which 2 parts are used for feature selection (Feature Selection Data) and the rest for the scope detection experiments (Scope Detection Data). The Feature Selection Data are divided into 5 equal parts, within which 4 parts for training and the rest for developing. In our scope detection experiments, we divide the Scope Detection Data into 10 folds randomly, so as to perform 10-fold cross validation. As the experiment data is easily confusable, Figure 5 illustrates the allocation.

Checking the validity of our method, we use the Abstracts sub-corpus in Section 5.2, 5.3 and 5.4, while in Section 5.5 we use all of the three sub-corpora (Abstracts, Full Papers, and Clinical Reports) to test the robustness of our system when applied to different text types within the same domain.

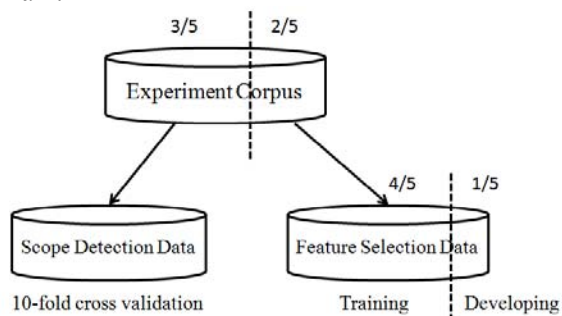


Figure 5. The allocation for experiment data.

The evaluation is made using the precision, recall and their harmonic mean, F1-score. Additionally, we report the accuracy in PCS (Percentage of Correct Scopes) applied in CoNLL’2010, within which a scope is fully correct if all tokens in a sen-

tence have been assigned to the correct scope class for a given cue. The evaluation in terms of precision and recall measures takes a token as a unit, whereas the evaluation in terms of PCS takes a scope as a unit. The key toolkits for scope classification include:

Constituent and Dependency Parser: All the sentences in BioScope corpus are tokenized and parsed using the Berkeley Parser (Petrov et al, 2007)² which have been trained on the GENIA TreeBank 1.0 (Tateisi et al, 2005)³, a bracketed corpus in PTB style. 10-fold cross-validation on GTB1.0 shows that the parser achieves 87.12% in F1-score. On the other hand, we obtain the dependency relations by the Stanford Dependencies Parser⁴.

Support Vector Machine Classifier: SVM^{Light}⁵ is selected as our classifier, which provides a way to combine the tree kernels with the default and custom SVM^{Light} kernels. We use the default parameter computed by SVM^{Light}.

Besides, according to the guideline of the BioScope corpus, scope must be a continuous chunk. The scope classifier may result in discontinuous blocks, as each token may be classified inside or outside the scope. Therefore, we perform the rule based post-processing algorithm proposed by Morante et al (2008) to obtain continuous scopes.

5.2 Results on Flat Syntactic Features

Relying on the results of the greedy feature selection algorithm (described in Section 4.1), we obtain 9 effective features {B1, B3, B6, CS3, CS4, CS9, DS1, DS3, DS5} (see Table 2, 3 and 4) for negation scope detection and 13 effective features {B3, B4, B5, B6, CS1, CS5, CS6, CS8, CS9, CS10, DS1, DS4, DS5} for speculation. Table 6 lists the performances on the Scope Detection Data by performing 10-fold cross validation. It shows that flat constituent and dependency syntactic features significantly improve the basic scope detection by 13.48% PCS for negation and 30.46% for speculation ($\chi^2; p < 0.01$). It demonstrates that the selected syntactic features are effective for scope detection.

² <http://code.google.com/p/berkeleyparser>

³ <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA>

⁴ <http://nlp.stanford.edu/software/lex-parser.shtml>

⁵ <http://svmlight.joachims.org>

Negation	Features	P	R	F	PCS
	Basic	89.89	68.72	77.86	39.50
	Con.	85.72	67.80	75.66	41.81
	Dep.	90.31	69.01	78.19	40.08
	Bas.&Con.	88.86	79.07	83.61	51.64
	Bas.&Dep.	90.44	73.62	81.17	49.36
All	91.21	76.57	83.25	52.98	
Speculation	Features	P	R	F	PCS
	Basic	89.67	86.86	88.24	40.09
	Con.	96.43	87.46	91.72	66.57
	Dep.	90.84	87.04	88.89	44.45
	Bas.&Con.	95.66	92.08	93.83	69.59
	Bas.&Dep.	92.39	88.27	90.28	67.49
All	95.71	92.09	93.86	70.55	

(Note: “Bas.” denotes basic features; “Con.” denotes Constituent features; “Dep.” denotes Dependency features; “All” contains Basic, Constituent, and Dependency features being selected.)

Table 6. Performance of flat syntactic features.

The results also show that the speculative scope detection achieves higher performance (16.98% higher in PCS) ($\chi^2; p < 0.01$) than the negation scope detection. The main reason is that although the average sentence length of negation and speculation are comparable (29.97 vs. 29.39 words, in Table 1), the average length of speculation scopes is much longer than the negation (17.24 vs. 9.62 words, in Table 1) in Abstracts sub-corpus. With the shorter scopes in training data, the classifier inevitably have more negative samples. Thus, by using a token as the basic unit in our classification, the imbalanced samples will seriously mislead the classifier and result in bias on the negative samples.

In addition, both constituent and dependency flat features can improve the scope classification, for the reason that the constituent features usually provide the nearer syntactic information of the cues, and that the further syntactic information between cues and scopes have been obtained by the dependency features.

5.3 Results on Structured Syntactic Parse Features

Table 7 and Table 8 give the scope detection performance using the different structured syntactic parse features on negation and speculation respectively. Compared to the optimal system (using all of the selected flat features in Table 6) in Section 5.2, the structured syntactic parse features at best improve the scope classification nearly 17.29% on negation (PCS=70.27%) and 12.32% on speculation (PCS=82.87%) ($\chi^2; p < 0.01$). It indicates that the structured syntactic parse features can provide more implicit linguistic information, as supplementary clues, to support scope classification.

The improvements also show that both the completed syntactic sub-trees and critical paths in constituent and dependency parsing trees are effective. The reason is that the completed syntactic sub-trees contain the surrounding information related to cues and tokens, while there are more direct syntactic information in the critical paths between cue and its scope.

Features	P	R	F	PCS
Con. CT	91.12	83.25	86.89	54.57
Con. CT&CP	93.31	89.32	91.20	66.58
Dep. T1 CT	87.29	84.37	85.81	53.07
Dep. T1 CT&CP	90.03	86.77	88.37	59.53
Dep. T2 CT	88.17	84.58	86.34	53.76
Dep. T2 CT&CP	91.09	87.31	89.16	60.11
All	93.84	91.94	92.88	70.27

(Note: “Con.” denotes Constituent features; “Dep.” denotes Dependency features; “T1” use the transformational rule (1) in Section 4.2 to get the dependency tree; “T2” use the transformational rule (2) in Section 4.2 to get the dependency tree; CT-“Completed syntactic sub-Tree”; CP-“Critical Path”; “All” contains Con CT&CP, Dep T1 CT&CP and Dep T2 CT&CP)

Table 7. Performance of structured syntactic parse features on negation.

Features	P	R	F	PCS
Con. CT	95.89	93.37	94.61	75.17
Con. CT&CP	96.05	94.36	95.20	76.73
Dep. T1 CT	93.24	90.77	91.99	72.31
Dep. T1 CT&CP	94.28	92.30	93.28	73.75
Dep. T2 CT	93.76	89.68	91.67	73.06
Dep. T2 CT&CP	95.29	94.55	94.92	75.69
All	96.93	96.86	96.89	82.87

Table 8. Performance of structured syntactic parse features on speculation.

5.4 Results on Part-of-Speech Based Classification

To confirm the assumption in Section 4.3, we have built a discriminative classifier for each kind of POS of cues. Considering that the features involving the global structured syntactic parse information in Section 4.2 are almost effective to all instances, we only use the flat syntactic features in Section 4.1.

Negation	System	P	R	F	PCS
	All Features	91.21	76.57	83.25	52.98
	POS Classifier	91.79	78.29	84.50	56.77
Speculation	System	P	R	F	PCS
	All Features	95.71	92.09	93.86	70.55
	POS Classifier	95.79	93.13	94.44	71.68

(Note: “All Features” System is the optimal system in Section 5.2)

Table 9. Performances of POS based classification.

Table 9 shows the performance of POS based classification. Compared with the system which only uses one classifier for all cues in Section 5.2,

the POS based classification improves 1.13% on PCS (χ^2 ; $p < 0.01$), as different POS kinds of cues involve respectively effective features with more related clues between cue and its scope.

Table 10 lists the performance of each POS kind of cues in speculation scope classification. There are still some low performances in some kinds of POS of cues. We consider it caused by two reasons. Firstly, some kinds of POS of cues (e.g. NN etc.) have fewer samples (just 43 samples shown in Table 5). For this reason, the training for classifier is limited. Then, for these low performance kinds of POS of cues, we may have not found the effective features for them. Although there are some kinds of cues with low performance, the whole performance of part-of-speech based classification is improved.

Cue's POS	B1-B6						CS1-CS10										DS1-DS5					PCS	
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5		
CC	✓				✓	✓				✓						✓	✓	✓	✓	✓	✓	✓	38.45
IN	✓				✓	✓				✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	87.99
JJ	✓					✓				✓				✓	✓							✓	31.83
MD				✓	✓	✓		✓	✓					✓	✓	✓	✓	✓	✓	✓	✓	✓	79.84
NN	✓					✓	✓	✓		✓				✓	✓		✓	✓					65.83
RB					✓	✓	✓		✓	✓						✓						✓	37.03
VB						✓	✓									✓	✓						44.29
VBD				✓	✓			✓	✓					✓	✓	✓	✓	✓	✓	✓	✓	✓	63.57
VBG				✓	✓			✓	✓					✓	✓	✓	✓	✓	✓	✓	✓	✓	82.89
VBN				✓	✓			✓						✓	✓	✓	✓	✓	✓	✓	✓	✓	66.38
VBP				✓	✓	✓	✓		✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	81.91
VBZ				✓	✓	✓	✓		✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	77.16

Table 10. Performance of each POS kind of cues in speculation scope classification.

5.5 Results of Comparison Experiments

To get the final performance of our approach, we train the classifiers respectively by different effective features in Section 4.1 for POS kinds of cues, and use the structured syntactic parse features in Section 4.2 on Abstracts sub-corpus by performing 10-fold cross validation.

Negation	System	Abstract	Paper	Clinical
	Morante (2008)	57.33	N/A	N/A
	Morante (2009a)	73.36	50.26	87.27
	Ours	76.90	61.19	85.31
Speculation	System	Abstract	Paper	Clinical
	Morante (2009b)	77.13	47.94	60.59
	Özgür (2009)	79.89	61.13	N/A
	Ours	84.21	67.24	72.92

Table 11. Performance comparison of our system with the state-of-the-art ones in PCS.

The results in Table 11 show that our system outperforms the state of the art ones both on nega-

tion and speculation scope detection. Results also show that the system is portable to different types of documents, although performance varies depending on the characteristics of the corpus.

In addition, on both negation and speculation, the results on Clinical Reports sub-corpus are better than those on Full Papers sub-corpus. It is mainly due to that the clinical reports are easier to process than full papers and abstracts. The average length of sentence for negative clinical reports is 8.19 tokens, whereas for abstracts it is 29.39 and for full papers 30.49. Shorter sentences imply shorter scopes. The more unambiguous sentence structure of short sentence can make the structured constituent and dependency syntactic features easier to be processed.

6 Conclusion

This paper proposes a new approach for tree kernel-based scope detection by using the structured syntactic parse information. In particular, we have explored the way of selecting compatible features for different part-of-speech cues. Experiments show substantial improvements of our scope classification and better robustness.

However, the results on the Full Papers and the Clinical Reports sub-corpora are lower than those on the Abstracts sub-corpus for both negation and speculation. That is because the structured syntactic parse features contain some complicated and lengthy components, and the flat features cross corpus are sparse. Our future work will focus on the pruning algorithm for the syntactic structures and analyzing errors in depth in order to get more effective features for the scope detection on different corpora.

Acknowledgments

This research is supported by the National Natural Science Foundation of China, No.61272260, No.61373097, No.61003152, the Natural Science Foundation of Jiangsu Province, No.BK2011282, the Major Project of College Natural Science Foundation of Jiangsu Province, No.11KJA520003 and the Graduates Project of Science and Innovation, No.CXZZ12_0818. Besides, thanks to Yu Hong and the three anonymous reviewers for their valuable comments on an earlier draft.

References

- Emilia Apostolova, Noriko Tomuro and Dina Demner-Fushman. 2011. Automatic Extraction of Lexico-Syntactic Patterns for Detection of Negation and Speculation Scopes. In *Proceedings of ACL-HLT short papers*, pages 283-287.
- Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34 (5): 301-310.
- Wanxiang Che, Min Zhang, Ting Liu and Sheng Li. 2006. A Hybrid Convolution Tree Kernel for Semantic Role Labeling. In *Proceedings of ACL*, pages 73-80.
- Nigel Collier, Hyun S. Park, Norihiro Ogata, et al. 1999. The GENIA Project: Corpus-Based Knowledge Acquisition and Information Extraction from Genome Research Papers. In *Proceedings of EACL*.
- Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of CoNLL: Shared Task*, pages 1-12.
- Ilya M. Goldin and Wendy W. Chapman. 2003. Learning to Detect Negation with ‘Not’ in Medical Texts. In *SIGIR Workshop: Text Analysis and Search for Bioinformatics*.
- Yang Huang and Henry Lowe. 2007. A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. *Journal of the American Medical Informatics Association*, 14(3):304-311.
- Zhengping Jiang and Hwee T. Ng. 2006. Semantic Role Labeling of NomBank: A Maximum Entropy Approach. In *Proceedings of EMNLP*, pages 138-145.
- Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the Scope of Negation in Biomedical Texts. In *Proceedings of EMNLP*, pages 715-724.
- Roser Morante and Walter Daelemans. 2009a. A Metalearning Approach to Processing the Scope of Negation. In *Proceedings of CoNLL*, pages 21-29.
- Roser Morante and Walter Daelemans. 2009b. Learning the Scope of Hedge Cues in Biomedical Texts. In *Proceedings of the BioNLP Workshop*, pages 28-36.
- Roser Morante, Vincent Van Asch and Walter Daelemans. 2010. Memory-Based Resolution of In-Sentence Scopes of Hedge Cues. In *Proceedings of CoNLL Shared Task*, pages 40-47.
- Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113-120.
- Lilja Øvrelid, Erik Velldal, and Stephan Oepen. 2010. Syntactic Scope Resolution in Uncertainty Analysis. In *Proceedings of COLING*, pages 1379-1387.
- Arzucan Özgür and Dragomir R. Radev. 2009. Detecting Speculations and their Scopes in Scientific Text. In *Proceedings of EMNLP*, pages 1398-1407.
- Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of NAACL*, pages 404-411.
- Liliana M. Sánchez, Baoli Li, Carl Vogel. 2007. Exploiting CCG Structures with Tree Kernels for Speculation Detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning: Shared Task*, pages 126-131.
- György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope corpus: Annotation for Negation, Uncertainty and their Scope in Biomedical Texts. In *Proceedings of BioNLP*, pages 38-45.
- Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun’ichi Tsujii. 2005. Syntax Annotation for the GENIA Corpus. In *Proceedings of IJCNLP, Companion volume*, pages 222-227.
- Erik Velldal, Lilja Øvrelid, Jonathon Read and Stephan Oepen. 2012. Speculation and Negation: Rules, Rankers, and the Role of Syntax. *Computational Linguistics*, 38(2):369-410.
- Veronika Vincze, György Szarvas, Richárd Farkas, György Móra and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(Suppl 11):S9.
- Guodong Zhou, Min Zhang, Donghong Ji, and Qiaoming Zhu. 2007. Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages, 728-736.