Speculation and Negation Scope Detection via Convolutional Neural Networks

Zhong Qian[1], Peifeng Li[1], Qiaoming Zhu[1], Guodong Zhou[1], Zhunchen Luo[2] and Wei Luo[2]
School of Computer Science and Technology, Soochow University, Suzhou, 215006, China[1]
China Defense Science and Technology Information Center, Beijing, 100142, China[2]
qianzhongqz@163.com, {pfli, qmzhu, gdzhou}@suda.edu.cn,
zhunchenluo@gmail.com, htqxjj@126.com

Abstract

Speculation and negation are important information to identify text factuality. In this paper, we propose a Convolutional Neural Network (CNN)-based model with probabilistic weighted average pooling to address speculation and negation scope detection. In particular, our CNN-based model extracts those meaningful features from various syntactic paths between the cues and the candidate tokens in both constituency and dependency parse trees. Evaluation on BioScope shows that our CNN-based model significantly outperforms the state-of-the-art systems on Abstracts, a sub-corpus in BioScope, and achieves comparable performances on Clinical Records, another sub-corpus in BioScope.

1 Introduction

Factual information is critical to understand a sentence or a document in most typical NLP applications. Speculation and negation extraction has been drawing more and more attentions in recent years due to its importance in distinguishing counterfactual or uncertain information from the facts. Generally speaking, speculation is a type of uncertain expression between certainty and negation, while negation is a grammatical category which reverses the truth value of a proposition.

Commonly, speculation and negation extraction involves two typical subtasks: cue identification and scope detection. Here, a cue is a word or phrase that has speculative or negative meaning (e.g., suspect, guess, deny, not), while a scope is a text fragment governed by the corresponding cue in a sentence. Consider the following two sentences for examples:

(S1) *The doctors warn that smoking [**may** harm our lungs]*.

(S2) *He does [**not** like playing football] but likes swimming.*[1]

In sentence S1, the speculative cue "***may***" governs the scope "***may** harm our lungs*", while the negative cue "***not***" governs the scope "***not** like playing football*" in sentence S2.

Previous work have achieved quite success on cue identification (e.g., with F1-score of 86.79 for speculative cue detection in Tang et al. (2010)). In comparison, speculation and negation scope detection is still a challenge due to its inherent difficulties and those upstream errors. In this paper, we focus on scope detection. Previous work on scope detection can be classified into heuristic rules based methods (e.g., Özgür et al., 2009; Øvrelid et al., 2010), machine learning based methods (e.g., Tang et al., 2010; Zou et al., 2013), and hybrid approaches which integrate empirical models with manual rules (Velldal et al., 2012).

Different from those previous studies, this paper presents a Convolutional Neural Network (CNN)-based approach for scope detection. CNN models, firstly invented to capture more abstract features for computer vision (LeCun et al., 1989), have achieved certain success on various NLP tasks in recent years, such as semantic role labeling (Collobert et al., 2011), machine translation (Meng et al., 2015; Hu et al., 2015), event extraction (Chen et al., 2015; Nguyen et al., 2015), etc. These studies have proved the ability of CNN models in learning meaningful features.

[1] In this paper, cues are in ***bold face***, and scopes are in *[brackets]* in the example sentences.

In particular, our CNN-based model extracts various kinds of meaningful features from the syntactic paths between the cue and the candidate token in both constituency and dependency parse trees. The importance of syntactic information in scope detection has been justified in previous work (Velldal et al., 2012; Lapponi et al., 2012; Zou et al., 2013, etc). Our model can also benefit from the ability of neural networks in extracting useful information from syntactic paths (Xu et al., 2015a; Xu et al., 2015b) or more complex syntactic trees (Ma et al., 2015; Tai et al., 2015). Moreover, instead of traditional average pooling, our CNN-based model utilizes probabilistic weighted average pooling to alleviate the overfitting problem (Zeiler et al., 2013). Experimental results on BioScope prove the effectiveness of our CNN-based model.

The reminder of this paper is organized as follows: Section 2 gives an overview of the related work. Section 3 describes our CNN-based model with probabilistic weighted average pooling for scope detection. Section 4 illustrates the experimental settings, and reports the experimental results and analysis. Finally, Section 5 draws the conclusion.

2 Related Work

In this section, we give an overview of previous work on both scope detection and utilization of CNNs in NLP applications.

2.1 Scope Detection

Earlier studies on speculation and negation scope detection focused on developing various heuristic rules manually to detect scopes.

Chapman et al. (2001) developed various regular expressions for negation scope detection. Subsequently, various kinds of heuristic rules began to emerge. Özgür et al. (2009) resorted to the part-of-speech of the speculative cues and the syntactic structures of the current sentences for identifying scopes, and developed heuristic rules according to the syntactic trees. Øvrelid et al. (2010) constructed a set of heuristic rules on dependency structures and obtained the accuracy of 66.73% on the CoNLL evaluation data. The approaches based on heuristic rules were effective because the sentence structures in BioScope satisfy some grammatical rules to a certain extent.

With the release of the BioScope corpus (Szarvas et al., 2008), machine learning based methods began to dominate the research of speculation and negation scope detection.

Morante et al. (2008) regarded negation scope detection as a chunk classification task utilizing lexical and syntactic features. Morante et al. (2009a) further implemented a scope detection system combining three classifiers, i.e., TiMBL, SVM and CRF, based on shallow syntactic features, and achieved the performance of 77.13% and 73.36% in Percentage of Correct Scopes (PCS) on speculation and negation scope detection on Abstracts, a sub-corpus of BioScope. Velldal et al. (2012) explored a hybrid method, adopting manually crafted rules over dependency parse trees and a discriminative ranking function over nodes in constituent parse trees. Zou et al. (2013) proposed a tree kernel based approach on the syntactic parse trees to detect speculation and negation scopes.

Alternative studies treated scope detection as a sequential labeling task. Tang et al. (2010) proposed a CRF model with POS, chunks, NERs, dependency relations as features. Similarly, Lapponi et al. (2012) employed a CRF model with lexical and dependency features for negation scope and event resolution on the Conan Doyle corpus. These machine learning methods manifest the effectiveness of syntactic features.

2.2 CNN based NLP Applications

Currently, CNNs have obtained certain success on various NLP tasks, e.g., part-of-speech tagging, chunking, named entity recognition (Collobert et al., 2011). Specifically, CNNs have been proven effective in extracting sentence-level features. For instance, Zeng et al. (2014) utilized a CNN-based model to extract sentence-level features for relation classification. Zhang et al. (2015) proposed a shallow CNN-based model for implicit discourse relation recognition. Chen et al. (2015) presented a CNN-based model with dynamic multi-pooling on event extraction.

More recently, researchers tend to learn features from complex syntactic trees. Ma et al. (2015) used a CNN-based model for sentence embedding, utilizing dependency tree-based n-grams. Xu et al. (2015a) exploited a CNN-based model to learn features from the shortest dependency path between the subject and the object for semantic relation classification.

3 CNN-based Modeling with Probabilistic Weighted Average Pooling

This section describes our CNN-based model for speculation and negation scope detection, which is recast as a classification task to determine

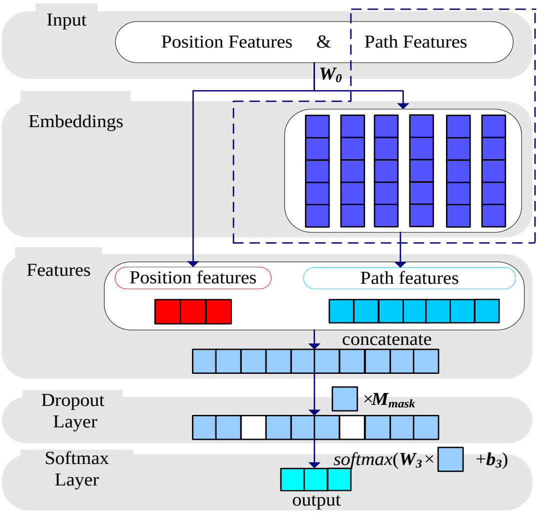


Figure 1: The framework of CNN for scope detection.

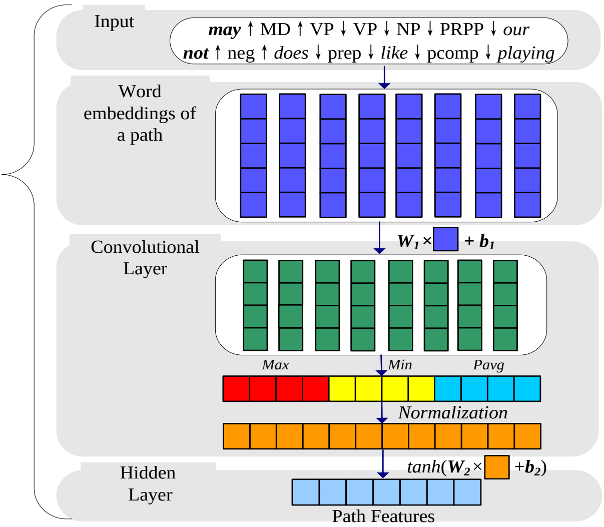


Figure 2: The architecture of CNN-based model to extract path features.

whether each token in a sentence belongs to the scope of the corresponding cue or not. Principally, our CNN-based model first extracts various path features from syntactic trees with a convolutional layer and concatenates them with their relative positions into one feature vector, which is then fed into a softmax layer to compute the confidence scores of its location labels, as described in subsection 3.1.

3.1 Token Labeling

We employ following labeling scheme for each candidate token:

- A token is labeled as ***O*** if it is NOT an element of a speculation or negation scope;
- A token is labeled as ***B*** if it is inside a scope and occurs before the cue, i.e., $P_{token} < P_{cue}$, where P_{token} and P_{cue} are the positions of the token and the cue in a sentence, respectively;
- A token is labeled as ***A*** if it is inside a scope and occurs after the cue (inclusive), i.e., $P_{token} \geqslant P_{cue}$.

Under this scheme, each token in a sentence is classified into ***B***, ***A*** or ***O***. For example, the labels of all the tokens in sentence S3 are shown in sentence S4.

(S3) *They think that [those bacteria **may** be killed by white blood cells] , but other researchers do not think so.*

(S4) *They/O think/O that/O [those/B bacteria/B **may/A** be/A killed/A by/A white/A blood/A cells/A] ,/O but/O other/O researchers/O do/O not/O think/O so/O ./O*

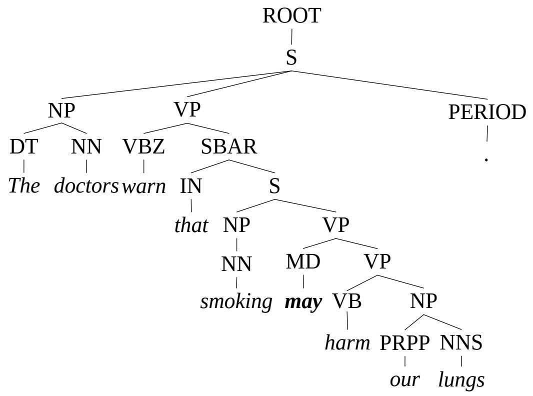
The advantage of our scheme is that it can describe the location relationship among the tokens, cues and scopes more precisely than some previous studies, which regarded scope detection as a binary classification task (Øvrelid et al., 2010; Zou et al., 2013). Compared to other schemes with more than two labels (Morante et al., 2009a; Tang et al., 2010; Lapponi et al., 2012), our scheme can much alleviate the imbalance of labels, because the tokens occurring at the first or last positions of the scopes are much fewer than other tokens.

3.2 Input Representation

Figure 1 shows the framework of our neural network for scope detection. We concentrate on two aspects, i.e., **Position Feature** and **Path Feature**. After extracted with the convolutional layer, path features, together with position features, are concatenated into one feature vector, which is finally fed into the softmax layer to obtain the output vector.

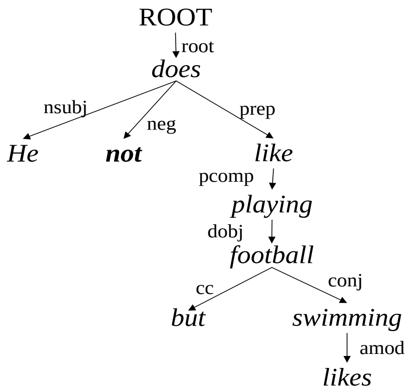
Relative Position has been proven useful in previous studies (Zeng et al., 2014; Chen et al., 2015). In this paper, relative position is defined as the relative distance of the cue to the current candidate token. For instance, in sentence S1, the relative distances of the cue "***may***" to the candidate tokens "*warn*" and "*our*" are 3 and -2, respectively. The values of position features are mapped into a vector ***P*** of dimension d_p, with ***P*** initialized randomly.

Instead of the word sequence (e.g., Zeng et al., 2014; Zhang et al. 2015; Chen et al., 2015), we argue that the **Shortest Syntactic Path** from the cue to the candidate token can offer effective



Cue: ***may***
Current candidate token: *our*
Constituency path:
may ↑ MD ↑ VP ↓ VP ↓ NP ↓ PRPP ↓ *our*

Figure 3: An example for the constituency parse tree of sentence S1 and the path from the cue to the candidate token.



Cue: ***not***
Current candidate token: *playing*
Dependency path:
not ↑ neg ↑ *does* ↓ prep ↓ *like* ↓ pcomp ↓ *playing*

Figure 4: An example for the dependency parse tree of sentence S2 and the path from the cue to the candidate token.

features to determine whether a token belongs to the scope. It is remarkable that the lowest common ancestor node of the cue and the token is the highest tree node in the path.

Figure 2 illustrates the architecture of our CNN-based model to extract path features. Here, convolutional features are first extracted from the matrix of embeddings of the path, and then fed into the hidden layer to produce more complicated features.

In this paper, the syntactic paths between the cues and the candidate tokens in constituency and dependency parse trees are both considered. Figure 3 presents the constituency parse tree of sentence S1 and the constituency path from the cue "***may***" to the candidate token "*our*". It shows that the tokens are at both the beginning and the end of the path with the arrows indicating the directions. Meanwhile, Figure 4 displays the dependency parse tree of sentence S2 and the dependency path from the cue "***not***" to the token "*playing*".

As the input of our CNN-based model, both the constituency path and the dependency path between the cue and the token can be regarded as a special "sentence" $S=(t_1, t_2,..., t_n)$, whose "words" can be tokens of sentences, syntactic categories, dependency relations, and arrows.

Similar to other CNN-based models, we also consider a fixed size window of tokens around the current token to capture its local features in the path. Here, the window size is set as an odd number *w*, indicating that there are (*w*-1)/2 tokens before and after the candidate token, respectively. In this case, path *S* is transferred into matrix $\boldsymbol{X_0} \in \mathbb{R}^{wd_0 \times n}$ according to embedding table $\boldsymbol{W_0} \in \mathbb{R}^{d_0 \times |T_0|}$, where d_0 is the dimension of the embeddings and $|T_0|$ is the size of the table.

3.3 Convolution Neural Networking

After fed into the convolutional layer, the matrix of the syntactic path $\boldsymbol{X_0}$ is processed with a linear operation:

$$\boldsymbol{Y_1} = \boldsymbol{W_1}\boldsymbol{X_0} + \boldsymbol{b_1} \qquad (1)$$

where $\boldsymbol{W_1} \in \mathbb{R}^{n_1 \times wd_0}$ is the parameter matrix, and $\boldsymbol{b_1} \in \mathbb{R}^{n_1}$ is the bias term. To extract the most active convolutional features from $\boldsymbol{Y_1} \in \mathbb{R}^{n_1 \times n}$, we consider two features ***Cmax*** and ***Cmin*** whose elements are maximum, minimum values of rows in $\boldsymbol{Y_1}$, respectively:

$$\boldsymbol{Cmax}(r) = max[\boldsymbol{Y_1}(r,0), \boldsymbol{Y_1}(r,1), ..., \boldsymbol{Y_1}(r,n-1)] \quad (2)$$

$$\boldsymbol{Cmin}(r) = min[\boldsymbol{Y_1}(r,0), \boldsymbol{Y_1}(r,1), ..., \boldsymbol{Y_1}(r,n-1)] \quad (3)$$

where $0 \le r \le n_1 - 1$. Moreover, we extract a convolutional feature ***Cpavg***, whose elements are probabilistic weighted average values of rows in $\boldsymbol{Y_1}$. Formally, ***Cpavg*** can be written as:

$$\boldsymbol{Cpavg}(r) = \sum_{i=0}^{n-1} p_i \cdot \boldsymbol{Y_1}(r,i) \qquad (4)$$

In Equation (4), p_i is the probability of the element $\boldsymbol{Y_1}(r,i)$ in the vector $\boldsymbol{Y_1}(r,\cdot)$:

$$p_i = \frac{|\boldsymbol{Y_1}(r,i)|}{\sum_{j=0}^{n-1} |\boldsymbol{Y_1}(r,j)|} \qquad (5)$$

Cpavg is a variant probabilistic weighted average pooling used by Zeiler et al. (2013). Compared to the standard average pooling, each element in ***Cpavg*** has a weight depending on its value. That is, during computing ***Cpavg***, the most active elements with the largest absolute values (i.e., the maximum and minimum values) play the leading roles, while those less active elements with smaller absolute values have less effect. In this way, we can reduce the influence of less active elements, and can capture more active information in $\boldsymbol{Y_1}(r,\cdot)$. From this respect, ***Cpavg*** can be regarded as a meaningful convolutional feature.

The extracted convolutional features above are first concatenated into $\boldsymbol{C} \in \mathbb{R}^{3n_1}$, as the output of the convolutional layer:

$$\boldsymbol{C} = [\boldsymbol{Cmax}, \boldsymbol{Cmin}, \boldsymbol{Cpavg}] \tag{6}$$

Then, ***C*** is fed into the hidden layer to learn more complex and meaningful features. Here, we process ***C*** with a linear operation just like in the convolutional layer, and choose hyperbolic *tanh* as the activation function to get $\boldsymbol{Y}_2 \in \mathbb{R}^{n_2}$:

$$\boldsymbol{Y_2} = tanh(\boldsymbol{W_2 C} + \boldsymbol{b_2}) \tag{7}$$

where $\boldsymbol{W_2} \in \mathbb{R}^{n_2 \times 3n_1}$ is the parameter matrix, and $\boldsymbol{b_2} \in \mathbb{R}^{n_2}$ is the bias term. To produce the output of the hidden layer, a normalization operation is applied to eliminate the manifold differences among various features:

$$\boldsymbol{H} = \boldsymbol{Y_2} / \|\boldsymbol{Y_2}\| \tag{8}$$

In this way, we can obtain the path feature $\boldsymbol{H} \in \mathbb{R}^{n_2}$ for each candidate token and then concatenate it with the position feature ***P*** into one vector $\boldsymbol{F_0}$:

$$\boldsymbol{F_0} = [\boldsymbol{P}^T, \boldsymbol{H}^T]^T \tag{9}$$

where $\boldsymbol{F_0} \in \mathbb{R}^{n_f}$ is the feature vector of a candidate token with the dimension equaling the sum of n_2 and the dimension of ***P***. Besides, we also consider the dropout operation for regularization to prevent the co-adaptation of hidden units on the penultimate layer:

$$\boldsymbol{F_1} = \boldsymbol{F_0} \circ \boldsymbol{M} \tag{10}$$

where $\circ$ is an element-wise multiplication and ***M*** is a mask vector whose elements follow the Bernoulli distribution with the probability *p* of being 1. We determine whether the candidate token is in the scope of the current cue according to its $\boldsymbol{F_1}$.

3.4 Output

Finally, $\boldsymbol{F_1}$ is fed into the softmax layer:

$$\boldsymbol{O} = softmax(\boldsymbol{W_3 F_1} + \boldsymbol{b_3}) \tag{11}$$

where $\boldsymbol{W_3} \in \mathbb{R}^{n_3 \times n_f}$ is the parameter matrix, and $\boldsymbol{b_3} \in \mathbb{R}^{n_3}$ is the bias term. The dimension of ***O*** is n_3=3, which is equal to the number of labels representing whether the token is an element of the scope, just as described in subsection 3.1, and the elements of ***O*** can be interpreted as the confidence scores of the three labels, i.e., ***B***, ***A*** and ***O***.

To learn the parameters of the network, we supervise the predicted labels of ***O*** with the gold labels in the training set, and utilize the following training objection function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \log p(y_i \mid x_i, \theta) + \frac{\lambda}{2}\|\theta\|^2 \tag{12}$$

where $p(y_i \mid x_i, \theta)$ is the confidence score of the golden label y_i (***B***, ***A***, ***O***) of the training instance x_i, *m* is the number of the training instances, λ is the regularization coefficient and θ={$\boldsymbol{W_0}$, $\boldsymbol{W_1}$, $\boldsymbol{b_1}$, $\boldsymbol{W_2}$, $\boldsymbol{b_2}$, $\boldsymbol{W_3}$, $\boldsymbol{b_3}$} is the set of parameters. To train the CNN-based model, the Stochastic Gradient Descent algorithm is applied to fine-tune θ.

4 Experimentation

In this section, we first introduce the evaluation data, and then describe the experimental settings. Finally, we report the experimental results and analysis.

4.1 Corpus

We evaluate our CNN-based model on the BioScope corpus (Szarvas et al., 2008; Vincze et al., 2008), a widely used and freely available resource consisting of sentences annotated with speculative and negative cues and their scopes in biomedical domain.

BioScope includes 3 different sub-corpora: Abstracts of biological papers from the GENIA corpus (Collier et al., 1999), Full scientific Papers from Flybase and BMC Bioinformatics website, and Clinical radiology Records corpus. These texts in three sub-corpora ensure that BioScope can capture the heterogeneity of language use in biomedical domain. While Abstracts and Full Papers share the same genre, Clinical Records consists of shorter sentences. Previous studies regarded Abstracts as the main resource for text mining applications due to its public accessibility (e.g. through PubMed).

Table 1 shows the statistics of the BioScope corpus. While in both Abstracts and Full Papers, the average lengths of speculation and negation sentences are comparable (Abstracts: 29.77 vs 29.28; Full Papers: 30.76 vs 30.55). However, their average lengths of the negation scopes are shorter than those of speculation ones (Abstracts: 7.60 vs 15.10; Full Papers: 7.35 vs 13.38). Moreover, both the average lengths of sentences and scopes in Clinical Records are shorter than those of other two sub-corpora (Average length: 11.96 (speculation sentence), 8.53 (negation sentence), 4.92 (speculation scope) and 3.87 (negation scope)).

		Abs	Papers	Cli
Total	#Documents	1273	9	1954
	#Sentences	11871	2670	6383
	Ave. Len Sentences	25.47	24.54	7.71
Spe	#Sentences	2101	519	855
	#Scopes	2659	672	1112
	Ave. Len Sentences	29.77	30.76	11.96
	Ave. Len Scopes	15.10	13.38	4.92
Neg	#Sentences	1597	339	865
	#Scopes	1719	376	870
	Ave. Len Sentences	29.28	30.55	8.53
	Ave. Len Scopes	7.60	7.35	3.87

(Notes: “Ave. Len” denotes average length; “Abs”, “Papers” and “Cli” denote Abstracts, Full Papers and Clinical Records, respectively; “Spe” and “Neg” denote speculation and negation, respectively.)

Table 1: Statistics on the BioScope corpus.

4.2 Experimental Settings

Following the previous work (e.g., Özgür et al., 2009; Morante et al., 2009a, 2009b; Zou et al., 2013), we divide the Abstracts sub-corpus into 10 folds to perform 10-fold cross-validation. Moreover, to examine the robustness of our CNN-based model towards different text types within biomedical domain, all the models are trained on the same Abstracts sub-corpus. Therefore, the results on Abstracts can be regarded as in-domain evaluation while the results on Clinical Records and Full Papers can be regarded as cross-domain evaluation.

For the measurement, traditional Precision, Recall, and F1-score are used to report the token-based performance in scope detection, while the Percentage of Correct Scopes (PCS) is adopted to report the scope-based performance, which considers a scope correct if all the tokens in the sentence have been assigned the correct scope classes for a specific cue. Obviously, PCS can better describe the overall performance in scope detection. Besides, Percentage of Correct Left Boundaries (PCLB) and Percentage of Correct Right Boundaries (PCRB) are reported as partial measurements.

In all our experiments, both the constituency and dependency parse trees are produced by Stanford Parser[2]. Specially, we train the parser on the GENIA Treebank 1.0[3] (Tateisi et al., 2005), which contains Penn Treebank-style syntactic (phrase structure) annotation for the GENIA corpus. The parser achieves the performance of 87.12% in F1-score in terms of 10-fold cross-validation on GENIA TreeBank 1.0.

For the hyper-parameters in our CNN-based model, we set d_0=100, d_p=10, w=3, n_1=200, n_2=500, λ=10^{-4}, p=0.8. The embeddings of the tokens in ordinary sentences (as word sequences) are initialized by Word2Vec[4] (Mikolov et al., 2013).

For the baseline, we utilize the classifier-based baseline developed by Zou et al. (2013). Besides those typical features, constituency and dependency syntactic features are also included. Furthermore, Mallet[5] is selected as the classifier in our baseline.

In addition, since our CNN-based model may result in discontinuous blocks, we utilize a post-processing algorithm (Morante et al., 2008) to ensure the continuity of scopes. Meanwhile, the cue must be in its scope as defined in Bioscope.

4.3 Experimental Results on Abstracts

Table 2 summarizes the performances of scope detection on Abstracts. In Table 2, CNN_C and CNN_D refer the CNN-based model with constituency paths and dependency paths, respectively (the same below). It shows that our CNN-based models (both CNN_C and CNN_D) can achieve better performances than the baseline in most measurements. This indicates that our CNN-based models can better extract and model effective features. Besides, compared to the baseline, our CNN-based models consider fewer features and need less human intervention. It also manifests that our CNN-based models improve significantly more on negation scope detection than on speculation scope detection. Much of this is due to the better ability of our CNN-based models in identifying the right boundaries of scopes than the left ones on negation scope detection, with the huge gains of 29.44% and 25.25% on PCRB using CNN_C and CNN_D, respectively.

[2] http://nlp.stanford.edu/software/lex-parser.shtml
[3] http://www.geniaproject.org/genia-corpus/treebank
[4] https://code.google.com/archive/p/word2vec/
[5] http://mallet.cs.umass.edu/

	Systems	P(%)	R(%)	F1	PCLB(%)	PCRB(%)	PCS(%)
Speculation	Baseline	94.71	90.54	92.56	84.81	85.11	72.47
	CNN_C	**95.95**	**95.19**	**95.56**	**93.16**	**91.50**	**85.75**
	CNN_D	92.25	94.98	93.55	86.39	84.50	74.43
Negation	Baseline	85.46	72.95	78.63	84.00	58.29	46.42
	CNN_C	85.10	**92.74**	89.64	81.04	**87.73**	70.86
	CNN_D	**89.49**	90.54	**89.91**	**91.91**	83.54	**77.14**

Table 2: The performances on the Abstracts sub-corpus.

Table 2 illustrates that the performance of speculation scope detection is higher than that of negation (Best PCS: 85.75% vs 77.14%). It is mainly attributed to the shorter scopes of negation cues. Under the circumstances that the average length of negation sentences is almost as long as that of speculation ones (29.28 vs 29.77), shorter negation scopes mean that more tokens do not belong to the scopes, indicating more negative instances. The imbalance between positive and negative instances has negative effects on both the baseline and the CNN-based models for negation scope detection.

Table 2 also shows that our CNN_D outperforms CNN_C in negation scope detection (PCS: 77.14% vs 70.86%), while our CNN_C performs better than CNN_D in speculation scope detection (PCS: 85.75% vs 74.43%). To explore the results of our CNN-based models in details, we present the analysis of top 10 speculative and negative cues below on CNN_C and CNN_D, respectively.

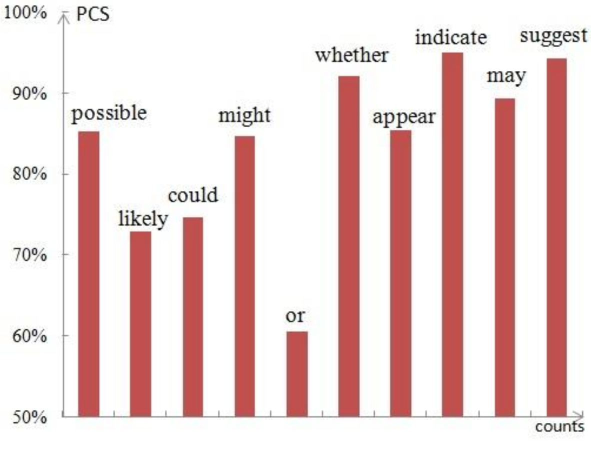


Figure 5: PCSs of top 10 speculative cues for scope detection in Abstracts sub-corpus.

Figure 5 illustrates the PCSs of the most frequent 10 speculative cues using CNN_C. The cues in the horizontal axis are in the order of lowest to highest in frequency. Among those cues, "*suggest*", "*may*", "*indicate*", and "*appear*" are commonly used to express opinions of certain individuals. The scopes of these cues are integrated semantic fragments (probably clauses) governed by corresponding cues in grammatical sense, and the tokens in the scope tend to share the same chunk with the cue in the constituency parse tree. Hence, constituency paths are more useful for speculation scope detection. Figure 5 also shows that the PCSs of all the top 10 speculative cues are higher than 70% except "*or*" (PCS: 60.44%), mainly due to the flexible usage of "*or*", which can connect two words, two professional terms, or even two clauses.

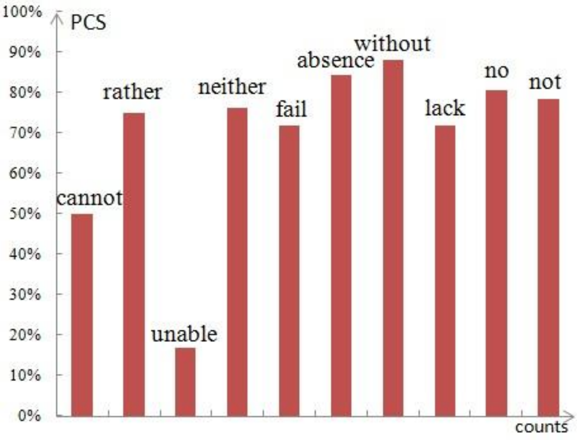


Figure 6: PCSs of top 10 negative cues for scope detection in Abstracts sub-corpus.

Figure 6 illustrates the performances of the most frequent 10 negative cues using CNN_D. In those negative cues, "*not*" is in the absolute majority, and "*not*" and "*no*" cover over 70%. We have noticed that most negative cues (e.g., "*not*", "*no*", "*without*", "*fail*") are often applied to negate phrases, and the tokens in negation scope tend to have the tight dependency relationship with them. Therefore, our model can achieve better results using dependency paths for negation scope detection.

In Figure 6, most negative cues have good PCSs (higher than 70%). However, "*unable*" has poor PCS of 16.67%. This is due to the fact that "*unable*" usually occurs in the phrase structure "*be unable to*", which often follows a subject. It is notable that a cue is always in its scope and most cues in BioScope are much closer to the left boundaries than to the right ones. Hence, the tokens labeled as ***B*** (i.e., inside the scope and before the cue) are much fewer than the ones labeled as ***A*** or ***O***. Such imbalance makes it hard to

	Systems		P(%)	R(%)	F1	PCLB(%)	PCRB(%)	PCS(%)
Speculation	Clinical Records	CNN_C	86.85	**93.84**	**90.21**	**84.35**	**86.87**	**73.92**
		CNN_D	**89.02**	85.41	87.18	82.91	76.17	64.39
	Full Papers	CNN_C	**86.78**	**86.59**	**86.69**	**86.01**	**68.60**	**59.82**
		CNN_D	86.13	85.09	85.61	80.95	64.14	52.98
Negation	Clinical Records	CNN_C	88.29	97.00	92.44	**95.98**	**93.45**	**89.66**
		CNN_D	**91.97**	**97.03**	**94.43**	**95.98**	90.57	87.82
	Full Papers	CNN_C	80.92	82.26	81.58	82.71	**67.29**	**55.32**
		CNN_D	**82.08**	**84.90**	**83.46**	**84.04**	64.89	53.99

Table 3: The performances of our CNN-based models on Clinical Records and Full Papers.

judge whether the tokens before "*unable"* are the elements of its scope or not.

4.4 Experimental Results on Clinical Records and Full Papers

The performances of our CNN-based models on the other two sub-corpora, i.e., Clinical Records and Full Papers, are presented in Table 3. Although Abstracts and Clinical Records have different genres, our CNN-based models can obtain satisfactory results on Clinical Records using both constituency paths and dependency paths, proving the portability of our models.

Table 3 also shows that the results of negation scope are better than those of speculation scope on Clinical Records (PCS: 89.66% vs 73.92%). We argue the reason is that both the lengths of negation sentences and scopes (8.53 and 3.87, respectively) in Clinical Records are much shorter, indicating that the structures of negation sentences are simpler than those of speculation ones. After error analysis of speculation scopes, we find that 54.83% of our error scopes contain the annotated scopes, just like sentence S5:

(S5) *This does not [**appear** to represent a stone] and is not mobile.*

The annotated scope of the cue "***appear***" is "***appear** to represent a stone*". However, our CNN-based model identifies the whole sentence as the scope. These errors indicate that some words may be wrongly identified as the components of scopes because the scopes in Clinical Records are short and their structures are simple.

Compared with Abstracts and Clinical Records, the results on Full Papers are much lower. This is mainly due to the poor PCRBs, indicating that a considerable quantity of right boundaries of scopes cannot be identified correctly. We should note that the average lengths of both speculation and negation sentences (30.76 and 30.55, respectively) in Full Papers are longer than those in Abstracts and Clinical Records. Normally, longer sentences mean more complicated structures of syntactic parse trees.

Besides the results trained on Abstracts, we also consider the 10-fold cross-validation on Clinical Records and Full Papers. The PCSs of speculation and negation scope detection are 74.73% (CNN_C) and 91.03% (CNN_C) on Clinical Records, which are both higher than the ones trained on Abstracts. Remember that Abstracts and Clinical Records come from the different genres. However, we get lower PCSs on Full Papers (49.54% for speculation scope detection using CNN_C, and 44.67% for negation scope detection using CNN_C). In addition to the complex structures of long sentences, another reason is that the smaller size of the Full Papers subcorpus compared to the other two sub-corpora. Fewer sentences and scopes (only 672 speculation scopes in 519 speculation sentences and 376 negation scopes in 339 negation sentences) mean that we cannot get an excellent model.

4.5 Comparison with the State-of-the-Art

Table 4 compares our CNN-based models with the state-of-the-art systems. It shows that our CNN-based models can achieve higher PCSs (+1.54%) than those of the state-of-the-art systems for speculation scope detection and the second highest PCS for negation scope detection on Abstracts, and can get comparable PCSs on Clinical Records (73.92% vs 78.69% for speculation scopes, 89.66% vs 90.74% for negation scopes). It is worth noting that Abstracts and Clinical Records come from different genres.

It also displays that our CNN-based models perform worse than the state-of-the-art on Full Papers. The main reasons are the complex syntactic structures of the sentences in Full Papers and the cross-domain nature of our evaluation on Full Papers. Although our evaluation on Clinical Records is cross-domain, the sentences in Clinical Records are much simpler and the results on Clinical Records are satisfactory. Remind that our CNN-based models are all trained on Abstracts. Another reason is that those state-of-the-art systems on Full Papers (e.g., Li et al., 2010; Velldal et al., 2012) are tree-based, instead of token-based. Li et al. (2010) proposed a semantic parsing framework and focused on determining whether a constituent, rather than a word, is in the scope of a negative cue. Velldal et al. (2012) presented a hybrid framework, combining a rule-based approach using dependency structures and a data-driven approach using a SVM ranker for selecting appropriate subtrees in constituent structures. Normally, tree-based models can better capture long-distance syntactic dependency than token-based ones. Compared to those tree-based models, however, our CNN-based model needs less manual intervention. To improve the performances of scope detection task, we will explore this alternative in our future work.

	System	**Abs**	**Cli**	**Papers**
Spe	Morante (2009a)	77.13	60.59	47.94
	Özgür (2009)	79.89	N/A	61.13
	Velldal (2012)	79.56	**78.69**	**75.15**
	Zou (2013)	84.21	72.92	67.24
	Ours	**85.75**	73.92	59.82
Neg	Morante (2008)	57.33	N/A	N/A
	Morante (2009b)	73.36	87.27	50.26
	Li (2010)	**81.84**	89.79	64.02
	Velldal (2012)	74.35	**90.74**	**70.21**
	Zou (2013)	76.90	85.31	61.19
	Ours	77.14	89.66	55.32

Table 4: Comparison of our CNN-based model with the state-of-the-art in PCS.

5 Conclusion

This paper proposes a CNN-based model for speculation and negation scope detection. Compared with various lexical and syntactic features adopted in previous studies (e.g., Lapponi et al., 2012; Zou et al., 2013), our CNN-based model only considers the position feature and syntactic path feature.

Experimental results on the BioScope corpus show that our CNN-based model can get the best performances for speculation scopes and the second highest performances for negation scopes on Abstracts in in-domain evaluation. In cross-domain evaluations, we can achieve comparable results on Clinical Records, but our CNN-based model performs worse on Full Papers. This suggests our future direction to extend the model from token level to parse tree level in better capturing long-distance syntactic dependency and to address the cross-domain adaptation issue.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No. 61472265 and No. 61402314), the State Key Program of National Natural Science Foundation of China (Grant No. 61331011), and partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. In addition, thanks to the three anonymous reviewers for their valuable comments and suggestions.

References

Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. Evaluation of Negation Phrases in Narrative Clinical Reports. In *Proceedings of American Medical Informatics Association Symposium*, 2001, pages 105-109.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL2015)*, Beijing, China, July 26-31, 2015, pages 167-176.

Nigel Collier, Hyun Seok Park, Norihiro Ogata, et al. 1999. The GENIA Project: Corpus-based Knowledge Acquisition and Information Extraction from Genome Research Papers. In *Proceedings of the European Chapter of the ACL 1999 (EACL 1999)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, 2011, 12(1): 2493-2537.

Baotian Hu, Zhaopeng Tu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2015. Context-Dependent Translation Selection Using Convolutional Neural Network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL2015)*, Beijing, China, July 26-31, 2015, pages 536-541.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO$_2$: Sequence-Labeling Negation Using Dependency Features. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, Montreal, Canada, June 7-8, 2012, pages 319–327.

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1989, 1(4): 541-551.

Junhui Li, Guodong Zhou, Hongling Wang, and Qiaoming Zhu. 2010. Learning the Scope of Negation via Shallow Semantic Parsing. In

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), Beijing, 2010, pages 671-679.

Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2015. Dependency-based Convolutional Neural Networks for Sentence Embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers) (ACL2015)*, Beijing, China, 2015, pages 174-179.

Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. 2015. Encoding Source Language with Convolutional Neural Network for Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL2015)*, Beijing, China, July 26-31, 2015, pages 20-30.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, 2013(26): 3111-3119.

Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the Scope of Negation in Biomedical Texts. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, Honolulu, 2008, pages 715-724.

Roser Morante and Walter Daelemans. 2009a. Learning the Scope of Hedge Cues in Biomedical Texts. In *Proceedings of the Workshop on BioNLP*. Boulder, Colorado, 2009, pages 28-36.

Roser Morante and Walter Daelemans. 2009b. A Metalearning Approach to Processing the Scope of Negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009)*, Boulder, Colorado, June 2009, pages 21-29.

Thien Huu Nguyen and Ralph Grishman. 2015. Event Detection and Domain Adaptation with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL2015)*, Beijing, China, July 26-31,2015, pages 365-371.

Arzucan Özgür and Dragomir R. Radev. 2009. Detecting Speculations and their Scopes in Scientific Text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, Singapore, 2009, pages 1398-1407.

Lilja Øvrelid, Erik Velldal, and Stephan Oepen. 2010. Syntactic Scope Resolution in Uncertainty Analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, August 2010, pages 1379-1387.

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 2008: Current Trends in Biomedical Natural Language Processing*, Columbus, Ohio, USA, 2008, pages 38-45.

Kai Sheng Tai, Richard Socher and Christopher D. Manning. 2015. Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, Beijing, China, July 26-31, 2015, pages 1556-1566.

Buzhou Tang, Xiaolong Wang, XuanWang, Bo Yuan, and Shixi Fan. 2010. A Cascade Method for Detecting Hedges and their Scope in Natural Language Text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL 2010): Shared Task*, Uppsala, Sweden, 15-16 July 2010, pages 13-17.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of IJCNLP 2005*, Jeju Island, Korea, October 2005, 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*, 2012, 38(2): 369-410.

Andreas Vlachos and Mark Craven. 2010. Detecting Speculative Language Using Syntactic Dependencies and Logistic Regression. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL 2010): Shared Task*, Uppsala, Sweden, 15-16 July 2010, pages 18-25.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal, 2015, pages 536-540.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, Zhi Jin. 2015b. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal, 2015, pages 1785-1794.

Matthew D. Zeiler and Rob Fergus. 2013. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. *arXiv preprint arXiv: 1301.3557v1*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014)*, Dublin, Ireland, August 23-29, 2014, pages 2335-2344.

Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015. Shallow Convolutional Neural Network for Implicit Discourse Relation Recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal, 17-21 September 2015, pages 2230-2235.

Bowei Zou, Guodong Zhou, and Qiaoming Zhu. 2013. Tree Kernel-based Negation and Speculation Scope Detection with Structured Syntactic Parse Features. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, Seattle, Washington, USA, 2013, pages 968-976.