Incorporating Structural Information for Better Coreference Resolution

Kong Fang[1,2,3] , **Fu Jian**[2,3]

[1]Institute of Artificial Intelligence, Soochow University, China
[2]Laboratory for Natural Language Processing, Soochow University, China
[3]School of Computer Science and Technology, Soochow University, China
kongfang@suda.edu.cn, 20164227014@stu.suda.edu.cn, gdzhou@suda.edu.cn

Abstract

Coreference resolution plays an important role in text understanding. In the literature, various neural approaches have been proposed and achieved considerable success. However, structural information, which has been proven useful in coreference resolution, has been largely ignored in previous neural approaches. In this paper, we focus on effectively incorporating structural information to neural coreference resolution from three aspects. Firstly, nodes in the parse trees are employed as a constraint to filter out impossible text spans (i.e., mention candidates) in reducing the computational complexity. Secondly, contextual information is encoded in the traversal node sequence instead of the word sequence to better capture hierarchical information for text span representation. Lastly, additional structural features (e.g., the path, siblings, degrees, category of the current node) are encoded to enhance the mention representation. Experimentation on the data-set of the CoNLL 2012 Shared Task shows the effectiveness of our proposed approach in incorporating structural information into neural coreference resolution.

1 Introduction

Coreference resolution aims to identify mentions in a text that refer to the same real world entity. It has been one of the key areas in NLP for two decades [Soon *et al.*, 2001; Ng and Cardie, 2002; Yang *et al.*, 2004; Lee *et al.*, 2013; Kummerfeld and Klein, 2013; Wiseman *et al.*, 2015; Clark and Manning, 2016b; Zhang *et al.*, 2018].

Recently, Lee et al. [2017] proposed the first state-of-the-art end-to-end neural coreference resolution system. They consider all text spans as potential mentions and therefore eliminate the need of carefully hand-engineered mention detection systems. The experiment results show that their system significantly outperforms all previous work in English language. However, there still exist two issues to be addressed. On the one hand, whether the scheme of taking all text spans as potential mentions is better than traditional rule-based mention extraction schemes, especially for other languages? On the other hand, syntactic information has been proven useful in many natural language understanding tasks including coreference resolution. Whether introducing syntactic information can further improve the performance of neural coreference resolution?

For the first issue, after duplicating the neural coreference resolution system proposed by Lee et al. [2017] using pytorch 1.0[1], we replace the neural span pruning based mention extraction model with the traditional rule-based mention extraction model which extracts mentions directly from syntactic parse trees[2]. Experimentation with the same setting in English shows that the overall performance reduces from 65.89% to 63.50%, i.e., with the performance gap of about 2.39% in F1. While for Chinese language, the performance increases from 58.41% to 60.06%. Further experimentation shows that considering all text spans as potential mentions can ensure the recall, however may bring a lot of noise as well as increase the computational intensity. This indicates that, the neural span pruning scheme achieves more effective mention collections for English language, while for Chinese, the traditional rule-based mention extraction model performs better than the span pruning scheme.

For the second issue, although structural information has been largely ignored in neural coreference resolution, structural embedding approaches have been explored in various NLP tasks and achieved considerable improvements [Liu *et al.*, 2017; Tymoshenko *et al.*, 2017; Zhu *et al.*, 2018]. Intuitively, neural coreference resolution should benefit from effective structural information.

In this paper, we incorporate structural information from three aspects.

- Firstly, we take the nodes of the parse trees as a constraint of mention candidates, and filter out those text spans without any exactly-matched node to reduce the computational complexity.
- Secondly, we propose a novel contextual encoding approach based on the traversal node sequence instead of the word sequence. By traversing the parse tree, we can achieve a node sequence implying the hierarchical information. Furthermore, we use the representation of child nodes to refine the parent node representation. In this

[1]https://pytorch.org/
[2]https://nlp.stanford.edu/software/dcoref.html

way, structural information can be well incorporated in text span representation.

- Finally, we encode additional structural features (e.g., the path, siblings, degrees, category of the current node) to improve the mention representation.

Experimentation on the data-set of the CoNLL 2012 Shared Task shows the effectiveness of our proposed approach in incorporating structural information into neural coreference resolution.

2 Related Work

Previous studies have shown that structural information played an important role in coreference resolution. For example, [Hobbs, 1978] proposed a parse tree-based pronoun resolution algorithm. By traversing the corresponding parse tree in the breadth-first way, the algorithm selected the appropriate noun phrase as the antecedent of a given pronoun from the resulting node sequence according to the dominance and government binding relationship in the grammatical structure. [Lappin and Leass, 1994] proposed the RAP algorithm for both third person pronoun and reflexive pronoun resolution. The algorithm first calculated the prominence of the candidate antecedent by manually weighting various linguistic features, and then determined the antecedent using various filtering rules. [Kong and Zhou, 2012] proposed a tree-kernel based approach to pronoun resolution for both English and Chinese languages. After dynamically expanding and pruning the parse tree based on the centering theory, competitor information and predicate-argument information, the algorithm used a convolution tree kernel function to directly calculate the similarity between the resulting structures and determine the coreferential relationship.

Due to the outstanding performance of neural networks in various NLP tasks during the last few years, researchers began to apply various neural network models to coreference resolution. For example, [Wiseman *et al.*, 2015] used neural networks to pre-train two separate subtasks, i.e., anaphoricity determination and antecedent ranking, in learning various feature representations for coreference resolution. [Wiseman *et al.*, 2016] used recurrent neural networks to learn global representations of entity clusters. [Clark and Manning, 2016a] employed reinforcement learning to directly optimize a neural mention-ranking model for coreference evaluation metrics, and achieved the best known performance on Chinese portion of the CoNLL 2012 Shared Task. [Lee *et al.*, 2017] proposed the first end-to-end neural coreference resolution system. They used a bi-directional LSTM and a Head-finding attention mechanism to represent the mentions, and exploited a mention-ranking model to determine coreference clusters. [Zhang *et al.*, 2018] proposed to use a biaffine attention model to get antecedent scores and jointly optimized the two sub-tasks to improve the performance of neural coreference resolution. [Lee *et al.*, 2018] proposed an approximation of higher-order inference using the span-ranking architecture from Lee et al. [2017] in an iterative manner.

From the above we can find that, although structural information has been proven very useful to coreference resolution, it has been largely ignored in neural coreference resolution systems. In this paper, we focus on effectively incorporating structural information into neural coreference resolution.

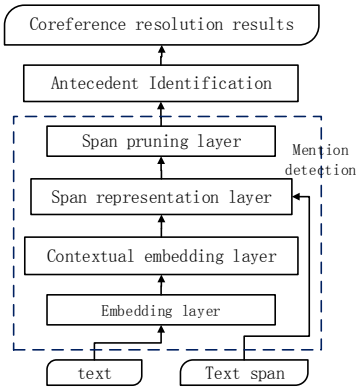


Figure 1: Our neural mention-ranking model.

3 Baseline: Neural Coreference Resolution

Following the work of [Lee *et al.*, 2017], our neural coreference resolution model adopts the mention-ranking approach. Such a model scores pairs of mentions for their coreferential likelihood rather than comparing partial coreference clusters. Hence it operates in a simple setting where coreference decisions are made independently. Figure 1 shows the framework of our baseline system, which consists of two components, i.e., mention detection and antecedent identification.

3.1 Mention Detection

In the mention detection stage, we first combine character-level, word-level and context-level information with an attention mechanism to represent each possible mention candidate (text span). Then, we use some scoring mechanism to rank the prominence of the text spans. Finally, we select a certain proportion of text spans as the set of mentions to be resolved. The detailed procedure of mention detection is shown as the part enclosed by the dotted line in Figure 1.

For a given text $D = \{w_1, w_2, \ldots, w_{N_D}\}$, the mention detector extracts all possible mentions $S = \{s_1, s_2, \ldots, s_n\}$ where N_D means the number of words in the document, $s_i = \{w_{b_i}, w_{b_i+1}, \ldots, w_{e_i}\}$ means a mention, b_i and e_i are the start and end index of mention s_i, $1 \leq b_i \leq e_i \leq N_D$.

- In the embedding layer, for $\forall w_i \in D$, by combining character and word embeddings, we achieve $\boldsymbol{x}_i = [\boldsymbol{w}_i, \boldsymbol{c}_i] \in \mathbb{R}^{d_x}$, where $d_x = d_w + d_c$, $\boldsymbol{w}_i \in \mathbb{R}^{d_w}$ means the word embedding of the word w_i, and $\boldsymbol{c}_i \in \mathbb{R}^{d_c}$ means the character embedding vector, which can be achieved using a Character CNN or a Character LSTM. In this way, we can get the embedding representation of the document D

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_D}] \in \mathbb{R}^{N_D \times d_x} \quad (1)$$

- In the contextual embedding layer, we use bidirectional LSTM as the contextual encoder to model the dependen-

cies between words. For the input $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$, we can get the contextual representation $\boldsymbol{h}_i \in \mathbb{R}^{d_h}$.

- In the span representation layer, we represent a given text span $s_i = \{w_{b_i}, w_{b_i+1}, \ldots, w_{e_i}\}$ as

$$\boldsymbol{s}_i = [\boldsymbol{h}_{b_i}, \boldsymbol{h}_{e_i}, \hat{\boldsymbol{x}}_i, \boldsymbol{f}_i] \tag{2}$$

where, $\boldsymbol{s}_i \in \mathbb{R}_{d_s}$, $\boldsymbol{f}_i$ is the additional feature vector to encode the span width, and $\hat{\boldsymbol{x}}_i$ is the weighted sum of the representations of all the words contained in s_i, which are learned using the Head-finding attention mechanism.

$$\alpha_t = \boldsymbol{v}_\alpha^\top FFNN_\alpha(\boldsymbol{h}_t) \tag{3}$$

$$a_{i,t} = \frac{exp(\alpha_t)}{\sum_{k=b_i}^{e_i} exp(\alpha_k)} \tag{4}$$

$$\hat{\boldsymbol{x}}_t = \sum_{t=b_i}^{e_i} a_{i,t} \cdot \boldsymbol{x}_t \tag{5}$$

- In the span pruning layer, we first use a multi-layer feedforward neural network to score the spans according to their representation. Then, we sort the spans by their scores, and select a certain part of spans with higher scores as the mention candidates.

$$score_i^m = FFNN_m(\boldsymbol{s}_i) \tag{6}$$

3.2 Antecedent Identification

In the antecedent identification stage, the basic idea is to find the best antecedent in a search space. Similar to traditional Mention-Pair models, we do mention pairing from the back to the front in a certain search space according to the mention appearing order. For each mention-pair instance, we calculate its score using a feedforward neural network.

$$score_{i,j}^a = FFNN_a([\boldsymbol{s}_i, \boldsymbol{s}_j, \boldsymbol{s}_i \odot \boldsymbol{s}_j, \boldsymbol{f}_{i,j}]) \tag{7}$$

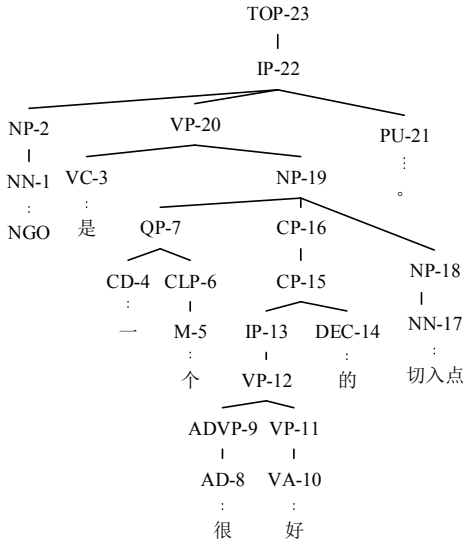
where $\boldsymbol{f}_{i,j}$ encodes speaker,genre and distance features. In the coreference scoring layer, we combine the scores of two mentions s_i and s_j, and the score of the mention-pair instance (s_i, s_j) to achieve the final coreferential score.

$$score_{i,j} = \begin{cases} 0, & s_j = \varepsilon \\ score_i^m + score_j^m + score_{i,j}^a, & s_j \neq \varepsilon \end{cases} \tag{8}$$

where, $j = \varepsilon$ means the case that the ith mention has no antecedent (i.e., non-anaphoric). At this point, the coreferential score is 0. For each mention, we choose the mention with the highest score in the search space as its antecedent.

4 Incorporating Structural Information

In this paper, we incorporate structural information from three aspects, i.e., taking nodes in the parse trees to filter out impossible text spans, encoding contextual information with the traversal node sequence, and encoding additional structural features (e.g., the path, siblings, degrees, category of the current node) to enhance the text span representation.



NGO/是/一/个/很/好/的/切入点/。
NGO is a good entry point.

Figure 2: An example of a parse tree.

4.1 Taking Nodes as a Filter Constraint

Same as [Lee *et al.*, 2017], our baseline system takes all text spans within a certain length limit as potential mentions. That is, the only limit is the length of the mention (i.e., the number of words making up a mention). Taking the sentence shown in figure 2 as an example, when the length limit is no more than 10, our baseline system can extract 45 mention candidates (e.g., NGO, NGO是, NGO是一, NGO是一个, NGO是一个很, NGO是一个很好, ...). Although all possible mentions can be covered, a large amount of noise is introduced due to the neglect of lexical and syntactic information. This greatly increases the computational burden of the baseline system. To address this issue, we propose to take node as a filter constraint. The motivation behind is that each valid mention naturally corresponds to a node in the parse tree. This is also different from the traditional rule-based mention extraction schemes, which extract possible mentions directly from the parse trees ignoring the length limit. After listing all text spans within the length limit, we look for the corresponding node for each text span and only keep the text spans having the node in the parse tree. For the example shown in Figure 2, for the extracted 45 mention candidates, we can get 15 mentions (e.g., 9 words corresponding to leaf nodes, "NGO是一个很好的切入点。","是一个很好的切入点", "一个", "一个很好的切入点", "很好", "很好的") after filtering with the node constraint.

4.2 Using the Traversal Node Sequence to Encode Context

Just as noted in 3.1, our baseline system uses a bidirectional LSTM as the contextual encoder to model the dependencies

between words. That is to say, the context information is represented as the dependencies between linear words. In fact, structural information between constituent words in a mention and between multiple mentions in a sentence is very important to coreference resolution. In this paper, we propose a node-based contextual encoder to better concentrate on the presentation of such information.

Our node-based contextual encoder is performed in three steps, i.e., contextualizing leaf nodes, updating non-terminal node representations, and re-contextualizing all nodes.

Contextualizing Leaf Nodes

By traversing the parse tree, we can achieve the node sequence[3]. In particular, we traverse the given parse tree t in post-order to get the node sequence $O^{(t)} = [o_1^{(t)}, o_2^{(t)}, \ldots, o_{n_t}^{(t)}]$, where n_t means the number of the nodes in parse tree t without the leaf nodes. In order to get the representation of the vector $O^{(t)}$, we first initialize it with zero vectors.

$$\boldsymbol{O}^{(t)} = [\mathbf{0}_1^{(t)}, \mathbf{0}_2^{(t)}, \ldots, \mathbf{0}_{n_t}^{(t)}] \in \mathbb{R}^{n_t \times d_x} \tag{9}$$

Then, we replace the representations of the leaf nodes with the corresponding word embedding representation.

$$\boldsymbol{o}_i^{(t)} = \begin{cases} \boldsymbol{x}_j & \text{if } o_i^{(t)} \text{ is a leaf and } t.leaves[j] = o_i^{(t)} \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{10}$$

We take the result representation vector $o^{(t)}$ as the input of the contextual embedding layer instead of the original word embedding matrix. In this way, the contextual embedding layer can learn both linear information and hierarchical structure representation implied in node sequence. So far, we can achieve the context representations $\boldsymbol{H}^{(t)}$.

$$\boldsymbol{H}^{(t)} = [\boldsymbol{h}_1^{(t)}, \boldsymbol{h}_2^{(t)}, \ldots, \boldsymbol{h}_{n_t}^{(t)}] \in \mathbb{R}^{n_t \times d_h} \tag{11}$$

Updating Non-terminal Node Representations

For non-terminal nodes, we exploit the representations of theirs child nodes to refine their representations. The basic strategy is to form a new representation for each non-terminal node by using an attention mechanism to transform the vector representation of their children into a fixed dimension vector, and a gating mechanism to combine the representation of child nodes with the original representation. The shaded part in Figure 3 shows the details.

For the node sequence $o^{(t)}$, we have got the contextual representations $\boldsymbol{H}^{(t)}$. For any non-terminal node $o_i^{(t)}$, the sequence of its child nodes is marked as $O_i^{(t)}.children$, and we can get the fixed dimension vector of the child nodes using Eq 12. For leaf nodes, we let $\boldsymbol{a}_i^{(t)} = \boldsymbol{h}_i^{(t)}$.

$$\boldsymbol{a}_i^{(t)} = \Gamma([\boldsymbol{h}_j^{(t)} \mid o_j^{(t)} \in o_i^{(t)}.children]) \tag{12}$$

Using a gated mechanism, we can update the representation of the node $o_i^{(t)}$.

$$score_i^g = \sigma(\mathbf{W}_g[\boldsymbol{h}_i^{(t)}, \boldsymbol{a}_i^{(t)}]) \tag{13}$$

[3]Word nodes are not considered

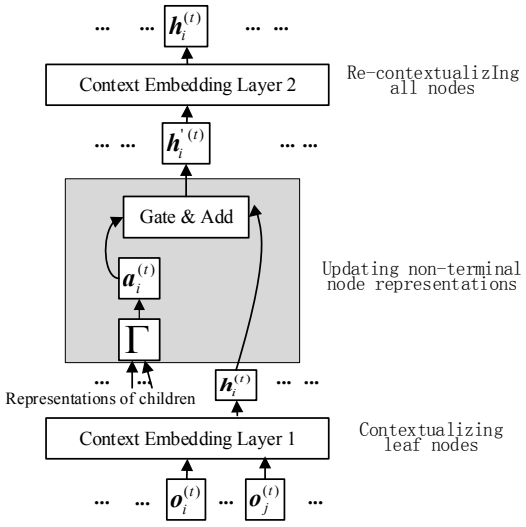


Figure 3: The node-based contextual encoder.

$$\boldsymbol{h}_i^{'(t)} = score_i^g * \boldsymbol{h}_i^{(t)} + (1 - score_i^g) * \boldsymbol{a}_i^{(t)} \tag{14}$$

Obviously, when $o_i^{(t)}.children = \emptyset$,

$$\begin{aligned} \boldsymbol{h}_i^{'(t)} &= score_i^g * \boldsymbol{h}_i^{(t)} + (1 - score_i^g) * \boldsymbol{a}_i^{(t)} \\ &= score_i^g * \boldsymbol{h}_i^{(t)} + (1 - score_i^g) * \boldsymbol{h}_i^{(t)} \\ &= \boldsymbol{h}_i^{(t)} \end{aligned} \tag{15}$$

In the meanwhile, we use a attention mechanism to implement the function Γ. Suppose that node $o_i^{(t)}$ has child node $o_j^{(t)}$, i.e., $o_j^{(t)} \in o_i^{(t)}.children$, and the contextual representations of these two nodes are $\boldsymbol{h}_i^{(t)}$ and $\boldsymbol{h}_j^{(t)}$, respectively. We aim to get $\boldsymbol{a}_i^{(t)}$ using the attention mechanism. In this paper, we use both dot attention and bilinear attention [Luong *et al.*, 2015] to compute the correlation between the representation of child nodes and the given node, with score functions defined as follows:

$$score(\boldsymbol{h}_j^{(t)}, \boldsymbol{h}_i^{(t)}) = \begin{cases} \boldsymbol{h}_j^{(t)\top} \boldsymbol{h}_i^{(t)} & dot \\ \boldsymbol{h}_j^{(t)\top} \boldsymbol{W}_a \boldsymbol{h}_i^{(t)} & bilinear \end{cases} \tag{16}$$

After using a softmax function to normalize $score(\boldsymbol{h}_j^{(t)}, \boldsymbol{h}_i^{(t)})$, we can get the final child nodes' representations of node o_i.

$$a_{i,j} = \frac{exp(score(\boldsymbol{h}_j^{(t)}, \boldsymbol{h}_i^{(t)}))}{\sum_{o_k \in o_i.children} exp(score(\boldsymbol{h}_k^{(t)}, \boldsymbol{h}_i^{(t)}))} \tag{17}$$

$$\boldsymbol{a}_i = \sum_{o_j \in o_i.children} a_{i,j} \cdot \boldsymbol{h}_j^{(t)} \tag{18}$$

Parameter	Description	Value
Degrees	the number of child nodes	3
Siblings	the number of left siblings and right siblings	[1,0]
Category	the syntactic category in the parse tree	NP
Path	the path from current node to the root of the parse tree	[NP,VP,IP,TOP]

Table 1: Additional features.

Re-contextualizing All Nodes

After updating the representations of non-termination nodes, we feed the achieved representation vector to the contextual embedding layer again, and take the output $\boldsymbol{H}^{(t)}$ as the final contextual encoding results.

4.3 Encoding Additional Structural Features

Some additional wide-used structural features as shown in Table 1 are extracted to enhance the representation of span texts. The third column lists the values associated with the node "NP-19" in the example shown in Figure 2.

For the feature $path_i = [label_{root}, \ldots, label_i]$, we first use a method similar to Character LSTM to represent it as a vector, $\boldsymbol{path}_i = [\boldsymbol{f}_{label,root}, \ldots, \boldsymbol{f}_{label,i}] \in \mathbb{R}^{depth_i \times d_f}$. and then take it as the input of a Bi-LSTM. In this way, we can achieve $\boldsymbol{h}_{label,i} = Bi - LSTM(\boldsymbol{f}_{label,i}) = [\overrightarrow{\boldsymbol{h}}_{label,i}, \overleftarrow{\boldsymbol{h}}_{label,i}]$. and take Eq (19) as the final representation of path feature.

$$\boldsymbol{f}_{path} = [\overrightarrow{\boldsymbol{h}}_{label,root}, \overleftarrow{\boldsymbol{h}}_{label,i}] \tag{19}$$

5 Experimentation

In this section, we systematically evaluate our proposed approach to neural coreference resolution.

5.1 Experimental Settings

All experiments are conducted in the data from the CoNLL-2012 shared task [Pradhan *et al.*, 2012]. The English part of this corpus contains 2802 training documents, 343 development documents, and 348 test documents. Accordingly, the Chinese part contains 1810 training documents, 252 development documents, and 218 test documents.

Our English models reuse the hyperparameters from Lee et al. [2017] with the difference between Chinese and English languages as shown in Table 2. We report the precision, recall, and F1 for the standard MUC, B^3, and $CEAF_{\phi_4}$ metrics using the official CoNLL-2012 evaluation scripts, with the average F1 of the three metrics as main evaluation standard.

5.2 Experimental Results

Table 3 compares our model with several start-of-the-art systems for English and Chinese languages, such as [Clark and Manning, 2016b], which presented a neural cluster-ranking model for coreference resolution benefiting from entity-level information, [Clark and Manning, 2016a], which employed reinforcement learning to directly optimize a neural mention-ranking model for coreference evaluation metrics, [Lee *et al.*, 2017], which introduced the first end-to-end coreference resolution system with no external resources, and [Lee *et al.*, 2018], which refined the span-ranking architecture as proposed in Lee[2017] by modeling higher order interactions between spans in predicted clusters and achieved a significant improvement. [4]

For fair comparison, our baseline duplicate the neural coreferece resolution system proposed by [Lee *et al.*, 2017] using pytorch 1.0. Comparing the results of our baseline and Lee's[2017], there exists a performance gap of 1.3%in average F1. This is caused by much less fine-tuning of various layers in mention detection due to GPU-resource limitation and time efficiency considerations. On one hand, [Lee *et al.*, 2017] use the Dropout strategy proposed by [Gal and Ghahramani, 2016]. Instead, we use the official bidirectional LSTM provided in pytorch. On the other hand, we re-implement some time consuming processes, such as text span pruning, in batch form. This significantly reduce the training time of our baseline to only 1/6 as of Lee's[2017] on the same hardware environment.We can find that,

- In comparison with our baseline, our improved model with structural information significantly outperforms the baseline in all metrics for both Chinese and English languages. This indicates the effectiveness of our incorporated structural information.
- For Chinese, our improved model achieves competitive results with the two start-of-the-art systems. Especially, our model outperforms them in precision in all metrics. This is due to our incorporation of richer structural information representations.
- For English, our model outperforms [Clark and Manning, 2016b] and [Clark and Manning, 2016a] in all metrics. Comparing the results of our model with [Lee *et al.*, 2017]'s, we can find that the most significant gains come from the improvement in precision. This indicates the effectiveness of our introduced structural information. There is still a big performance gap between our proposed model with [Lee *et al.*, 2018]. This is largely due to the employment of ELMo[Peters *et al.*, 2018]. Moreover, in this paper, we focus on incorporating structural information. In the future, we will explore more strategies to improve our system.

To show the contribution of each structural information to our proposed model, we take the system without any structural information as the baseline, and incorporate structural

[4] It is worth noting that, [Luo and Glass, 2018] presented a word embedding model to learn cross-sentence dependency. Integrating this model into the coreference resolution system proposed by Lee[2017] improved the average F1 by 0.6% from 67.2 to 67.8. [Peters *et al.*, 2018] introduced a new type of deep contextualized word representation ELMo(Embeddings from Language Models). Integrating ELMo into the coreference resolution system proposed by Lee[2017] improved the average F1 by 3.2% from 67.2 to 70.4. However, these two studies mainly focus on the more effective word representation approaches rather than coreference resolution itself.

Language	Component	Value	Parameters
English	Word Embedding	300D Glove[2014] & 50D Turian[2010]	trainable=false
	Character Embedding	8D random embedding	trainable=true
	Character Encoder	Character CNN	window_sizes=[3,4,5], each consisting of 50 filters
Chinese	Word Embedding	300D Qiu[2018] & 64D Polyglot[2013]	trainable=false
	Character Embedding	64D Polyglot[2013]	trainable=false
	Character Encoder	Character LSTM	hidden_size=100

Table 2: Difference of Word and Character Embeddings between Chinese and English Languages.

	Systems	MUC			B^3			$CEAF_{\phi_4}$			CoNLL
		P(%)	R(%)	F	P(%)	R(%)	F	P(%)	R(%)	F	Avg.F1
Chinese	[Clark and Manning, 2016b]	73.85	65.42	69.38	67.53	56.41	61.47	62.84	57.62	60.12	63.66
	[Clark and Manning, 2016a]	73.64	65.62	69.40	67.48	56.94	61.76	62.46	58.60	60.47	63.88
	Our baseline	74.54	60.35	66.70	66.79	49.94	57.15	61.01	48.62	54.11	59.32
	Our model	76.95	64.58	70.21	70.58	54.68	61.60	64.92	55.36	59.75	63.85
English	[Clark and Manning, 2016b]	78.93	69.75	74.06	70.08	56.98	62.86	62.48	55.82	58.96	65.29
	[Clark and Manning, 2016a]	79.19	70.44	74.56	69.93	57.99	63.40	63.46	55.52	59.23	65.73
	[Lee *et al.*, 2017]	78.40	73.40	75.80	68.60	61.80	65.00	62.70	59.00	60.80	67.20
	[Lee *et al.*, 2018]	81.40	79.50	80.40	72.20	62.30	70.80	68.20	67.10	67.60	73.00
	Our baseline	78.80	70.76	74.57	69.87	58.44	63.65	62.16	56.97	59.45	65.89
	Our model	80.52	73.92	77.08	71.17	61.48	65.97	64.26	61.13	62.66	68.57

Table 3: Performance of the start-of-the-art systems on the test set from the CoNLL-2012 shared task.

	Chinese		English	
Systems	Avg.F1	Δ	Avg.F1	Δ
Baseline	60.82		66.01	
+S1	62.71	1.89	66.42	0.41
+S2	63.02	2.20	66.93	0.92
+S3	62.00	1.18	66.51	0.50
+S1+S2	63.67	2.85	67.11	1.1
+S1+S3	63.03	2.21	66.77	0.76
+S2+S3	63.53	2.71	67.04	1.03
+S1+S2+S3	64.00	3.18	67.27	1.26

Table 4: Performance of our model on the development set from the CoNLL-2012 shared task using automatic parse trees (S1 - node constraint, S2 - node-based contextual encoder, S3 - additional structural features).

information one by one. Table 4 shows the average F1 on the development data set using automatic parse trees[5] for both English and Chinese languages.

From the results we can find that,

- The results on independent integration of the three schemes show that the second scheme (i.e.,the node-based contextual encoder) performs the best in both Chinese and English languages. It contributes about 2.20 and 0.92 F1 to the final results for Chinese and English, respectively. This indicates that the proposed node-based contextual embedding approach can well enhance the representation of the mentions and contribute much to coreference resolution.

[5]The automatic parse trees are provided by the CoNLL-2012 shared task.

- The combination of the three schemes achieves the best performance in both English and Chinese languages. This indicates the effectiveness of the incorporating structural information.

6 Conclusion

In this paper, we focus on effectively incorporating various kinds of structural information into neural coreference resolution. In particular, three schemes are exploited. First, we take nodes in the parse trees as a constraint to filter out impossible text spans in reducing the computational complexity. Second, we present a node-based contextual encoding scheme to capture hierarchical information in enriching the text span representation. Last, we encode additional structural features in enhancing the mention representation. Experimentation on the data-set of the CoNLL 2012 Shared Task shows the effectiveness of the structural information. Although our model much improves the performance of coreference resolution, there still exists much improvement room, such as the employment of ELMo. In the future work, we will pay more attention to antecedent identification.

Acknowledgements

This work is supported by Artificial Intelligence Emergency Project 61751206 under the National Natural Science Foundation of China, Project 61876118 under the National Natural Science Foundation of China and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[Al-Rfou *et al.*, 2013] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 183–192, 2013.

[Clark and Manning, 2016a] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of EMNLP-2016*, 2016.

[Clark and Manning, 2016b] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of ACL-2016*, August 2016.

[Gal and Ghahramani, 2016] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[Hobbs, 1978] J. R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

[Kong and Zhou, 2012] Fang Kong and Guodong Zhou. Pronoun resolution in english and chinese languages based on tree kernel. *Journal of Software*, 23(5):1085–1099, 2012.

[Kummerfeld and Klein, 2013] Jonathan K. Kummerfeld and Dan Klein. Error-driven analysis of challenges in coreference resolution. In *Proceedings of EMNLP-2013*, 2013.

[Lappin and Leass, 1994] Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4), 1994.

[Lee *et al.*, 2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4), 2013.

[Lee *et al.*, 2017] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of EMNLP-2017*, 2017.

[Lee *et al.*, 2018] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of NAACL-2018 Volume 2 (Short Papers)*, 2018.

[Liu *et al.*, 2017] Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. Structural embedding of syntactic trees for machine comprehension. In *Proceedings of EMNLP-2017*, 2017.

[Luo and Glass, 2018] Hongyin Luo and Jim Glass. Learning word representations with cross-sentence dependency for end-to-end co-reference resolution. In *Proceedings of EMNLP-2018*, 2018.

[Luong *et al.*, 2015] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP-2015*, pages 1412–1421, 2015.

[Ng and Cardie, 2002] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of ACL-2002*, 2002.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP-2014*, pages 1532–1543, 2014.

[Peters *et al.*, 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-2018*, 2018.

[Pradhan *et al.*, 2012] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *EMNLP-CoNLL-2012*, pages 1–40, 2012.

[Qiu *et al.*, 2018] Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 209–221. 2018.

[Soon *et al.*, 2001] Wee Meng Soon, Daniel Chung Yong Lim, and Hwee Tou Ng. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4), 2001.

[Turian *et al.*, 2010] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL-2010*, pages 384–394, 2010.

[Tymoshenko *et al.*, 2017] Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. Ranking kernels for structures and embeddings: A hybrid preference and classification model. In *Proceedings of EMNLP-2017*, 2017.

[Wiseman *et al.*, 2015] Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of ACL-IJCNLP-2015*, 2015.

[Wiseman *et al.*, 2016] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. In *Proceedings of NAACL-2016*, 2016.

[Yang *et al.*, 2004] Xiaofeng Yang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of ACL-2004*, 2004.

[Zhang *et al.*, 2018] Rui Zhang, Cicero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of ACL-2018 (Volume 2: Short Papers)*, 2018.

[Zhu *et al.*, 2018] Xunjie Zhu, Tingfeng Li, and Gerard de Melo. Exploring semantic properties of sentence embeddings. In *Proceedings of ACL-2018 (Volume 2: Short Papers)*, 2018.