Coordination Boundary Identification with Similarity and Replaceability

Hiroki Teranishi **Hiroyuki Shindo** **Yuji Matsumoto**
Nara Institute of Science and Technology
{teranishi.hiroki.sw5, shindo, matsu}@is.naist.jp

Abstract

We propose a neural network model for coordination boundary detection. Our method relies on two common properties — similarity and replaceability in conjuncts — in order to detect both similar and dissimilar pairs of conjuncts. The model improves the identification of clause-level coordination using bidirectional recurrent neural networks incorporating two properties as features. We show that our model outperforms existing state-of-the-art methods for the coordination annotated Penn Treebank and Genia corpus without any syntactic information from parsers.

1 Introduction

Coordination is a common structure and one of major ambiguities in human languages. Although coordination gives a large amount of syntactic or semantic information of coordinated phrases, disambiguating coordination still remains one of the difficult problems that state-of-the-art parsers cannot cope with.

Given a coordinator word, how can we find conjuncts? Coordinate structures are characterized by two properties: (1) similar structures often appear in conjuncts, and (2) one conjunct can be replaced with another conjunct without losing sentence consistency in syntax or semantics. However, many previous studies of coordination disambiguation rely only on the similarities between conjuncts, despite the fact that similarities are not always helpful (Shimbo and Hara, 2007; Hara et al., 2009; Hanamoto, 2012). For example, the sentence *"[at least two commercial versions have been put on the U.S. market], and [an estimated 500 have been sold]."* does not have similar phrases around the coordinating conjunction *"and."* Thus, existing methods sometimes fail to capture coordination. In addition to the case where there is a lack of similarities, many similarity-based methods use handcrafted features, heuristic rules, or external resources such as thesauri.

To overcome these problems, Ficler and Goldberg (2016) proposed a neural network model with the replaceability feature as well as the similarity feature. Their model produces candidate pairs of conjuncts using probabilities assigned by the Berkeley Parser. All candidate pairs are scored on the basis of the similarity, replaceability and parser-derived features, and then the best scored pair is picked. Their approach outperforms existing constituent parsers for the Penn Treebank and similarity-based coordination disambiguation methods such as those by Shimbo and Hara (2007) and Hara et al. (2009) for the Genia treebank. Although Ficler and Goldberg's (2016) method improves performance significantly, it heavily depends on the syntactic parser. They use the outputs from the parser not only for candidates generation and the feature for scoring, but also for computation of the similarities. The problems of propagated errors from the parser and dependencies on external resources still remain in their work.

In this work, we propose a neural network model for coordination disambiguation that does not require any external syntactic parser. Our model exploits both the similarity and replaceability properties to avoid suffering from an absence of these properties (Section 2). We use bidirectional recurrent neural networks (RNNs) to obtain the contextual information of candidate conjuncts and then compute similarity and replaceability features without syntactic information (Section 3). We show that our model performs well for both types of coordination: NP coordination (whose conjuncts tend to be similar) and S coordination

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 264–272,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

(whose conjuncts make sense individually) and outperforms the methods by Ficler and Goldberg (2016) and Hara et al. (2009) in Section 4.

The contributions of our work include the following:

(i) Our model can capture dissimilar conjuncts as well as similar ones using the similarity and replaceability features.

(ii) Our model performs better than others without any thesauri, feature engineering, or syntactic parsers to extract conjunct features.

2 Coordinate Structure Analysis

2.1 Task Description

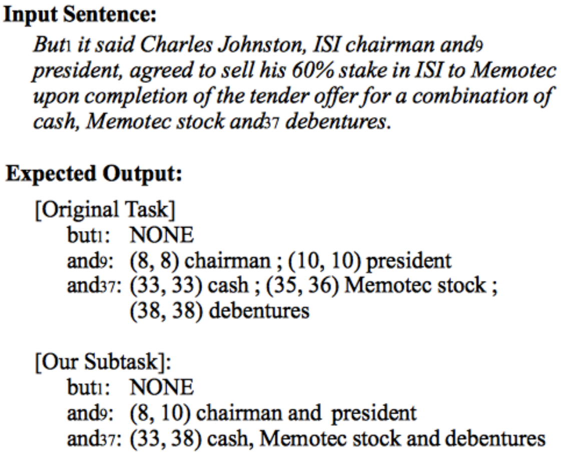
Coordination is a frequently occurring syntactic structure along with several phrases, known as conjuncts. The task of coordination disambiguation is identifying the boundaries of each conjunct with a single coordinator word as one coordinate structure instance. Given a coordinator word (e.g., “and,” “or,” or “but”), a system must return each conjunct span if the word actually plays the role of a coordinator; otherwise, NONE is output for the absence of coordination. The task sounds simple, yet is difficult because two complex phenomena appear in coordination.

1. A coordinator does not always connect two conjuncts. Sometimes, a coordinate structure consists of three or more consecutive conjuncts. For example[1],

   “It was not an unpleasant evening, certainly, thanks to [the high level of performance], [the compositional talents of Mr. Douglas], and$_{25}$ [the obvious sincerity with which Mr. Stoltzman chooses his selection].”

2. Two or more coordinate structures can be found in the same sentence. In addition, one coordinate structure can be nested inside another. For example,

   “Aside from [the Soviet economic plight] and$_{7}$ [talks on cutting (strategic) and$_{12}$ (chemical) arms], one other issue the Soviets are likely to want to raise is naval force reduction.”



Input Sentence:

But$_{1}$ it said Charles Johnston, ISI chairman and$_{9}$ president, agreed to sell his 60% stake in ISI to Memotec upon completion of the tender offer for a combination of cash, Memotec stock and$_{37}$ debentures.

Expected Output:

[Original Task]
but$_{1}$: NONE
and$_{9}$: (8, 8) chairman ; (10, 10) president
and$_{37}$: (33, 33) cash ; (35, 36) Memotec stock ; (38, 38) debentures

[Our Subtask]:
but$_{1}$: NONE
and$_{9}$: (8, 10) chairman and president
and$_{37}$: (33, 38) cash, Memotec stock and debentures

Figure 1: The coordination identification task and our subtask.

In this work, we solve this task by focusing on identifying the beginning and end of an entire coordinate structure. Figure 1 shows our task. We attempt to identify two conjuncts to the left and right sides of a conjunction. We call these conjuncts the *preconjunct* and *post-conjunct*, respectively[2]. In addition, we assume that the end of the preconjunct and the beginning of the post-conjunct adjoin a coordinator word; thus it appears that we work on the subtask of coordinate structure span identification. After identifying a coordination span, we recover individual conjuncts within the span.

2.2 Conjunct Properties

Coordination has many unique traits other than its structure. We focus on the key properties between conjuncts that can be helpful to disambiguate a coordination boundary.

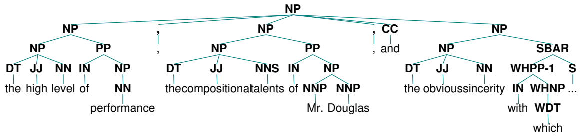
(a) **Similarity**: Conjuncts in a coordination have a similar structure or meaning.

(b) **Replaceability**: A conjunct can be replaced with another conjunct in the same coordination.

Conjuncts tend to have similar semantic or syntactic constituents. For example, the three conjuncts *“the high level of performance,”* *“the compositional talents of MR. Douglas,”* and *“the obvious sincerity with which Mr. Stolzman chooses his selection”* have part-of-speech (POS) tag sequences starting with *“DT JJ NN(S) IN NN(P)*

[1] We write coordinator words with their position in a sentence in the form of $word_{\text{position}}$ to distinguish them.

[2] If two or more conjuncts appear before a conjunction, we regard them as one conjunct.


NP
NP , NP , CC NP
NP PP , NP PP , and NP SBAR
DT JJ NN IN NP DT JJ NNS IN NP DT JJ NN WHPP-1 S
the high level of NN the compositional talents of NNP NNP the obvious sincerity IN WHNP ...
performance Mr. Douglas with WDT
which

(a) Similar structures between conjuncts

1. Aside from [the Soviet economic plight], one other . . .
2. Aside from [talks on cutting (strategic) arms], one other . . .
3. Aside from [talks on cutting (chemical) arms], one other . . .

(b) Replaceability

Figure 2: Characteristic of conjuncts

. . . ” in common. At a phrase level, they all are categorized as NP and have identical tree structures (Figure 2 (a)). Many previous works exploit this characteristic to detect conjuncts (Shimbo and Hara, 2007; Hara et al., 2009).

The replaceability of conjuncts is also often observed. A sentence is still consistent even if one conjunct is replaced with another one. For example, the coordination *“Aside from [the Soviet economic plight] and [talks on cutting (strategic) and (chemical) arms]”* can be transformed into *“Aside from [talks on cutting (chemical) and (strategic) arms] and [the Soviet economic plight]”* by exchanging conjuncts. Using this property, we can expand a coordinate structure as one sentence by one conjunct (Figure 2 (b)). Replaceability has recently been used to capture conjuncts (Ficler and Goldberg, 2016).

The two properties described above are essential clues to identify conjunct spans; however, they are not always available. Coordination sometimes has different types of conjuncts or an ellipsis in conjuncts. For similarity, when conjuncts belong to the S type or are different types of syntactic categories, their semantic and syntactic structures can be apart from each other (e.g., *“[We turned the trading system on]*$_{\mathrm{S}}$*, and [it did whatever it was programmed to do]*$_{\mathrm{S}}$*.” ; “Bill is [in trouble]*$_{\mathrm{PP}}$ *and [trying to come up with an excuse]*$_{\mathrm{VP}}$*.”*). For replaceability, when words are omitted in a latter conjunct, we cannot replace one conjunct with another unless we supplement omitted words (e.g., *“[Honeywell’s contract totaled $69.7 million], and [IBM’s $68.8 million].”*). To cope with the case where there is a lack of similarity or replaceability, our proposed method incorporates both features.

3 Proposed Method

Our proposed model calculates the scores of all possible preconjunct and post-conjunct pairs. Given a sentence $x = \{x_1, x_2, x_3, \ldots, x_N\}$ and coordinator word x_k, the preconjunct and post-conjunct can be written as $s_1 = \{x_i, \ldots, x_{k-1}\}$ $(1 \leq i \leq k-1)$ and $s_2 = \{x_{k+1}, \ldots, x_j\}$ $(k+1 \leq j \leq N)$, respectively. As we mentioned in Section 2, we fix the end of the preconjunct at $k-1$ and the beginning of the post-conjunct at $k+1$. Thus, our model learns and predicts a set of spans (i, j), which indicate the two positions of the beginning and end of a coordination. We identify preconjuncts and post-conjuncts by picking the highest scoring pairs as predicted conjunct spans.

Figure 3 shows an overview of our neural network architecture. This model consists of four components.

Input Layer: Map a sequence of one-hot words and POS tags onto their representations from embeddings.

RNN Layer: Produce a sequence of sentence-level representations based on contexts using a bidirectional RNN.

Feature Extractor: Generate the conjunct phrase representations and feature vectors of possible pairs of conjuncts.

Output Layer: Calculate the scores of pairs of conjuncts using MLP.

In the following subsections, we describe these components in detail.

3.1 Input Layer

The first step of our neural network model is to represent a sequence of words and POS tags in distributed vectors, known as embeddings (Bengio et al., 2003). Our model receives a sequence of one-hot encoded words and POS tags $\{x_n^{word}\}_{n=1}^{N}$, $\{x_n^{tag}\}_{n=1}^{N}$ and then looks them up in the matrices $E^{word} \in \mathbb{R}^{d_{word} \times |v_{word}|}$, $E^{tag} \in \mathbb{R}^{d_{tag} \times |v_{tag}|}$, resulting in a sequence of real-valued vectors $\mathbf{h}_n^{word} \in \mathbb{R}^d$, $\mathbf{h}_n^{tag} \in \mathbb{R}^d$, respectively. These real-valued vectors are concatenated as the input of the

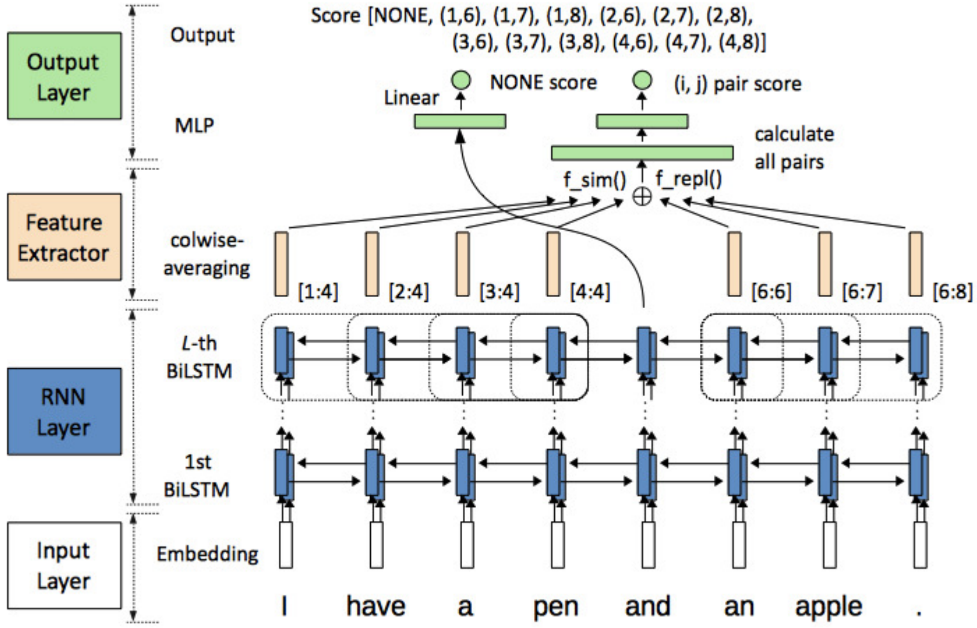


Figure 3: Overview of the architecture for coordination analysis.

next layer.

$$\begin{aligned}\mathbf{h}_t^{word} &= W^{word} x_t^{word} \\ \mathbf{h}_t^{tag} &= W^{tag} x_t^{tag} \\ \mathbf{h}_t^{(0)} &= [\mathbf{h}_t^{word}; \mathbf{h}_t^{tag}] \\ \mathbf{h}^{(0)} &= \{\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_N^{(0)}\}\end{aligned} \qquad (1)$$

3.2 RNN Layer

A sequence of distributed vectors is transformed into hidden state vectors using stacked bidirectional RNNs. Bidirectional RNNs process a time series of inputs from the past to a future direction and from the future to a past direction. We can make use of left-to-right (forward) and right-to-left (backward) contexts using these networks. The output of the ℓ-th layer of stacked bidirectional RNNs at a time step t in the forward direction, which is denoted as $\mathbf{h}_{\ell,t}^{f}$, is computed as

$$\mathbf{h}_{\ell,t}^{f} = f(\mathbf{h}_{\ell,t-1}^{f}, \mathbf{h}_{\ell-1,t}) \qquad (2)$$

where $\mathbf{h}_{\ell,t-1}^{f}$ is the hidden state vector of the same layer at the previous time step $t-1$ in the same direction and $\mathbf{h}_{\ell-1,t}$ is the hidden state vector of the previous bidirectional layer at the same time step t. The hidden vector of the ℓ-th layer of stacked bidirectional RNNs at a time step t in the backward direction is also computed in the same way. The stacked bidirectional RNNs that we use in this work output hidden state vectors by concatenating the vectors $\{\mathbf{h}_{\ell,t}^{f}\}_{t=1}^{T}$ from the forward direction and $\{\mathbf{h}_{\ell,t}^{b}\}_{t=1}^{T}$ from the backward direction at each time step t in every layer.

In general, an RNN has a function $f(\cdot)$ expressed as

$$f(\mathbf{x}_t, \mathbf{h}_{t-1}) = g(W\mathbf{x}_t + U\mathbf{h}_{t-1})$$

where $g(\cdot)$ is an arbitrary nonlinear function such as the hyperbolic tangent $tanh(\cdot)$ or rectified linear unit (ReLU). We use the long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) as the function $f(\cdot)$ to prevent backpropagated errors from vanishing or exploding, which arise in RNNs (Pascanu et al., 2013).

3.3 Feature Extractor

This component produces a feature vector based on a representation of a preconjunct and post-conjunct and a series of vectors $\{\mathbf{h}_t\}_{t=1}^{T}$ from bidirectional RNNs. We compute the preconjunct representation $\mathbf{v}_i^{pre}$ and post-conjunct $\mathbf{v}_j^{post}$ using the function $g(\cdot)$. In this work, we define element-wise averaging as the function $g(\cdot)$.

$$g(\mathbf{h}_{l:m}) = \text{average}\big(\mathbf{h}_l, \mathbf{h}_{l+1}, \ldots, \mathbf{h}_{m-1}, \mathbf{h}_m\big) \qquad (3)$$

Thus, $\mathbf{v}_i^{pre}$ and $\mathbf{v}_j^{post}$ are expressed as

$$\begin{aligned}\mathbf{v}_i^{pre} &= g(\mathbf{h}_{i:k-1}) \;\; (1 \le i \le k-1) \\ \mathbf{v}_j^{post} &= g(\mathbf{h}_{k+1:j}) \;\; (k+1 \le j \le N)\end{aligned} \qquad (4)$$

Then $\mathbf{v}_i^{pre}$ and $\mathbf{v}_j^{post}$ are fed into the following two feature extraction functions.

Similarity feature vector

In order to capture the similarity between the pre-conjunct and the post-conjunct, the feature vector is computed as follows:

$$f_{sim}(\mathbf{v}_i^{pre}, \mathbf{v}_j^{post}) = \left[|\mathbf{v}_i^{pre} - \mathbf{v}_j^{post}|; \mathbf{v}_i^{pre} \odot \mathbf{v}_j^{post}\right] \tag{5}$$

where $|\mathbf{v}_i^{pre} - \mathbf{v}_j^{post}|$ is the absolute value of element-wise subtraction, and $\mathbf{v}_i^{pre} \odot \mathbf{v}_j^{post}$ is element-wise multiplication. These subtraction and multiplication operations are intended to model the semantic distance and relatedness (Ji and Eisenstein, 2013; Tai et al., 2015; Hashimoto et al., 2016).

Replaceability feature vector

We define a feature vector based on the conjunct replaceability as follows.

$$\begin{aligned} f_{repl}(\mathbf{h}_{1:N}, i, j, k) = \\ \big[|\mathbf{h}_{i-1} \odot \mathbf{h}_i - \mathbf{h}_{i-1} \odot \mathbf{h}_{k+1}|; \\ |\mathbf{h}_j \odot \mathbf{h}_{j+1} - \mathbf{h}_{k-1} \odot \mathbf{h}_{j+1}|\big] \end{aligned} \tag{6}$$

where $\mathbf{h}_{i-1}$ is the context vector that is linked to the heads of conjuncts and $\mathbf{h}_{j+1}$ is the context vector that is linked to the tails of conjuncts. The first subtraction $|\mathbf{h}_{i-1} \odot \mathbf{h}_i - \mathbf{h}_{i-1} \odot \mathbf{h}_{k+1}|$ is the difference between two context-conjunct connections at the beginning of coordination. The second subtraction $|\mathbf{h}_j \odot \mathbf{h}_{j+1} - \mathbf{h}_{k-1} \odot \mathbf{h}_{j+1}|$ is the difference between two context-conjunct connections at the end of coordination. These distance measures can be interpreted as difficulty in replacing conjuncts. Note that the function $f_{repl}(\mathbf{h}_{1:N}, i, j, k)$ returns a zero vector when $i = 0$ or $j = N$.

3.4 Output Layer

This layer computes the scores of pairs of conjuncts based on the similarity feature vectors and the replaceability feature vectors. The network is a multilayered perceptron (MLP) that consists of multiple layers of computational units interconnected in a feed-forward way. The score of a pre-conjunct $(i, k-1)$ and post-conjunct $(k+1, j)$ candidate pair is calculated as

$$\begin{aligned} \mathrm{Score}(i, j) = \\ \mathrm{MLP}\big(\big[f_{sim}(\mathbf{v}_i^{pre}, \mathbf{v}_j^{post}); \\ f_{repl}(\mathbf{h}_{1:N}, i, j, k)\big]\big) \end{aligned} \tag{7}$$

To cope with the absence of coordination against a coordinator, we also calculate the score for a candidate of NONE. The score NONE is simply computed as the product of a weight vector and the sentence-level representation of the coordinator from the RNN layer.

$$\mathrm{Score}(\textsc{None}) = w \cdot \mathbf{h}_k + b \tag{8}$$

Using these scoring functions, we assign scores to all possible pairs of conjuncts. Thus, when the length of a sentence is N and a coordinator appears as the k-th word, we obtain $(k-1) \times (N-k) + 1$ candidates and choose the pair with the best score as the predicted conjuncts with the softmax function.

$$\begin{aligned} \mathbf{s} &= [\mathrm{Score}(\textsc{None}); \mathrm{Score}(1, \mathrm{k}+1); \ldots; \\ &\quad \mathrm{Score}(1, \mathrm{N}); \ldots; \mathrm{Score}(k-1, N)] \\ \hat{p_\theta}(y|x) &= \mathrm{softmax}(\mathbf{s}) \\ \hat{y} &= \arg\max_y \big(\hat{p_\theta}(y|x)\big) \end{aligned} \tag{9}$$

3.5 Learning

The loss function is the negative log-likelihood of the true pair of conjuncts $y^{(k)}$:

$$J(\theta) = -\sum_{d=1}^{D} \log \hat{p_\theta}(y^{(d)}|x^{(d)}) + \frac{\lambda}{2}\|\theta\|^2 \tag{10}$$

where D is the number of occurrences of coordinator words in a training dataset, θ is a set of model parameters, and the hyperparameter λ adjusts the regularization strength. The model parameters are optimized by minimizing the loss using the stochastic gradient descent (SGD).

4 Experiments

We evaluate our proposed model using the coordination annotated Penn Treebank (Ficler, 2016) and the Genia treebank beta (Kim et al., 2003). We present the number of occurrences of coordinator words and the number of sentences with coordination in Table 1[3].

[3] We consider "and," "or," "but," "nor," and "and/or" in the PTB and "and," "or," and "but" in the Genia as coordinator words following Ficler and Goldberg (2016) and Hara et al. (2009).

	# Coordinators	# Sentences
Penn Treebank	27903 (24450)	21314 (19095)
Training	22670 (17893)	17282 (13932)
Development	953 (848)	742 (673)
Testing	1282 (1099)	985 (873)
Genia	3598 (3598)	2508 (2508)

Table 1: The number of coordinators in the datasets. (#count) indicates the number of actual presences of coordination.

Parameter	Value
Dimension of the LSTM hidden vector	600
MLP units in the hidden layer	2400
Dropout ratio (all)	0.50
Regularization term λ	0.0001

Table 2: The final hyperparameters in the experiment for the Penn Treebank.

4.1 Evaluation Using the Penn Treebank

4.1.1 Experimental Setup

We use the coordination annotated Penn Treebank and divide it into wsj 2-21 as the training set, wsj 22 as the development set, and wsj 23 as the testing set. We use pretrained 200-dimensional word embeddings from the New York Times section in English Gigaword (fifth edition) (Parker et al., 2011) using Word2Vec[4] with its default parameter. For the POS tags, we use 10-way jackknifing using the Stanford POS Tagger (Toutanova et al., 2003) and initialize the 50-dimensional embeddings with the uniform distribution within $[-1, 1]$. We use three-layer bidirectional LSTMs as an RNN layer. The dimensionality of the LSTM hidden vectors in each direction is selected from {400, 600}. Our MLP consists of one hidden layer with ReLU activation, and an output layer. The number of the hidden layer units is selected from {1200, 2400}. The model parameters are optimized by the minibatched SGD with a batch size of 20. The learning rate is automatically tuned by Adam (Kingma and Ba, 2014). When training, we apply dropout (Srivastava et al., 2014) to the embeddings, input vectors of each LSTM in bidirectional LSTMs (except the first layer), and the hidden layer of the MLP. Dropout ratio is selected from {0.33, 0.50}. We choose the regularization strength λ from {0.0001, 0.0005, 0.001}. We train our model for 50 iterations and choose the model that achieves the best F1 score[5] on the development set and evaluate it with the testing set. We present the final hyperparameters choice in Table 2.

4.1.2 Evaluation Metrics

We evaluate our model on the basis of the ability to predict the beginning and end of each coordination (*whole*) with the precision, recall, and F1 measures. In another setup, we focus on NP coordination[6]. To compare the performance with Ficler and Goldberg (2016), we also evaluate our model with two conjunct spans that are adjacent to the coordinator (*inner*), the first and last conjuncts (*outer*), and all complete conjuncts (*exact*). Furthermore, in order to investigate the effectiveness of our proposed features, we perform the experiment with a simple baseline model that uses two averaged vectors as features (Eq. 3) and feeds them into the MLP instead of the similarity and replaceability features (Eq. 7).

Note that our proposed model learns and predicts the coordinate structure boundaries and not each conjunct; thus, when evaluating the inner, outer, and exact metrics, we simply divide the preconjuncts into subconjuncts using the character "," as the divider.

4.1.3 Results

We present the results in Table 3. For all metrics, the recall values are low compared with the precision values. Our model is likely to produce NONE for some coordinators by mistake. The proposed model suffers from a worse outer metric than the inner metric. Intuitively, this is because the preconjunct for the inner prediction is placed next to a coordinator and it is easier to identify its span, while outer conjuncts occur apart from the coordinators.

Table 4 summarizes the performance of different uses of features. The similarity and replaceability features work better than the baseline independently. However, the joint model performs the best by exploiting both features.

Table 5 presents a comparison with existing methods. For all coordination, our proposed method outperforms the state-of-the-art models with a test set F1 score of 72.81 (0.11 better than

[4] https://code.google.com/archive/p/word2vec/

[5] This F1 score is measured for the *whole* criterion, which is mentioned later.

[6] We consider that NP and NX are NP coordination as in Ficler and Goldberg (2016).

	All			NP		
	P	R	F	P	R	F
whole	75.92	72.87	74.36	77.90	75.05	76.45
outer	72.48	69.57	70.99	76.24	73.45	74.82
inner	74.07	71.10	72.56	77.43	74.59	75.99
exact	72.11	69.22	70.63	75.77	72.99	74.35

Table 3: Performance difference by the metrics for the PTB development set.

	All			NP		
	P	R	F	P	R	F
Baseline	70.83	68.75	69.77	74.27	72.87	73.57
f_{sim}	71.79	69.92	70.84	74.76	73.22	73.98
f_{repl}	74.29	71.58	72.91	76.12	73.68	74.88
Both	75.92	72.87	74.36	77.90	75.05	76.45

Table 4: Performance of different sets of features for the PTB development set for the outer metric. "f_{sim}," "f_{repl}," and "Both" indicate the use of similarity feature vectors, replaceability feature vectors, and both feature vectors, respectively.

	Dev			Test		
	P	R	F	P	R	F
	All Coordination					
Berkeley	70.14	70.72	70.42	68.52	69.33	68.92
Zpar	72.21	72.72	72.46	68.24	69.42	68.82
Ficler16	72.34	72.25	72.29	72.81	72.61	72.7
Ours	74.07	71.10	**72.56**	73.46	72.16	**72.81**
	NP Coordination					
Berkeley	67.53	70.93	69.18	69.51	72.61	71.02
Zpar	69.14	72.31	70.68	69.81	72.92	71.33
Ficler16	75.17	74.82	74.99	76.91	75.31	**76.1**
Ours	77.43	74.59	**75.99**	75.87	74.76	75.31

Table 5: Performance of inner-conjunct prediction on all coordination and on NP coordination for the PTB. The results for the three methods other than our method are reported in Ficler16 : (Ficler and Goldberg, 2016).

COOD	#	Ours	Ficler16	Hara09
Overall	3598	**65.98**	64.14	61.5
NP	2317	**66.59**	65.08	64.2
VP	465	63.87	**71.82**	54.2
ADJP	321	78.50	74.76	**80.4**
S	188	**52.65**	17.02	22.9
PP	167	53.89	56.28	**59.9**
UCP	60	50.00	**51.66**	36.7
SBAR	56	78.57	**91.07**	51.8
ADVP	21	**85.71**	80.95	85.7
Others	3	33.33	33.33	**66.7**

Table 6: Recall with Genia treebank beta. The numbers in the columns "Ficler16" and "Hara09" are taken from their papers; Ficler16 : (Ficler and Goldberg, 2016) ; Hara09 : (Hara et al., 2009).

the previously reported result). For NP coordination, our model achieves competitive results, despite the rough extraction of conjuncts from pre-conjuncts, even for inner-conjunct prediction.

4.2 Evaluation Using Genia

4.2.1 Experimental Setup

We also evaluate our model with the Genia treebank beta to compare with the previous work of Hara et al. (2009) and Ficler and Goldberg (2016). The settings of this experiment are based on those presented in Section 4.1.1, except for the following hyperparameters: Word embeddings are initialized by the pretrained 200-dimensional representation that BioASQ (Tsatsaronis et al., 2012) provides. These embeddings are trained from biomedical abstracts by using Word2Vec. We use gold POS as in Hara et al. (2009), and the dimension of the POS embeddings is 50. For regularization, we set $\lambda = 0.0005$ and train our model for 20 iterations.

4.2.2 Evaluation Metrics

As in Hara et al. (2009), we measure the recall values of coordinate structure boundary prediction, disregarding individual conjunct spans[7]. Thus, we do not decode conjuncts because our model can be compared directly. Coordination phrases in the Genia treebank are explicitly annotated with a special label (COOD). Making use of this label, we also measure the performance for each type of coordination, as reported in previous work. We evaluate our model by five-fold cross-validation, as in Hara et al. (2009).

4.2.3 Results

We present the results in Table 6. For all coordination, our model outperforms the scores reported by Hara et al. (2009) and Ficler and Goldberg (2016). In the evaluation of each type, our method greatly improves the performance for VP, SBAR, and especially the S type of coordination compared with the similarity-based method of Hara et al. (2009). Regarding the S type, our results are considerably better than those of Ficler and Goldberg (2016). As presented in Table 4, our proposed replaceability feature significantly contributes to the detection of this type of coordination, where only the similarity feature does not work because of a collapse

[7] In the Genia corpus, all coordinator words are associated with conjuncts; thus, there is no absence of coordination, as described in Table 1.

of similarity between conjuncts. The results for NP coordination, which accounts for nearly 65% of all coordination, are fairly good for the Genia corpus; however, the model proposed by Ficler and Goldberg (2016) exhibits better performance than ours for the PTB for the inner metric.

5 Related Works

Approaches using the similarity property between conjuncts have been developed in previous works. Regarding the task of coordination identification in Japanese, Kurohashi and Nagao (1994) used a chart to compute the similarity between conjuncts and identify conjunct spans with a dynamic programming technique. Shimbo and Hara (2007) proposed a sequence alignment model with dynamic programming to capture locally similar structures in two conjuncts on the basis of the set of features including word surfaces, POS tags, and morphological characteristics. The similarity score in their work is computed by a weighted linear combination (perceptron) of manually designed features assigned to edges and nodes in graphs, while the score in the work of Kurohashi and Nagao (1994) is calculated from a score function that uses some rules based on the observation of coordination. Although the method of Shimbo and Hara (2007) could not handle nested coordinate structures, Hara et al. (2009) extended their work to cope with nested coordination as well as three or more than consecutive conjuncts. Their proposed method defined several production rules to build consistent coordination trees with discriminative functions based on the similarity score. Hanamoto (2012) used dual decomposition to combine an HPSG parser with the model of Hara et al. (2009).

The method of use of the replaceability property has recently been adopted by Ficler and Goldberg (2016). They incorporated the replaceability property between conjuncts into the feature representations, as well as the similarity property. They made use of these properties to assign scores to candidate pairs of conjuncts. Their method consists of three components: a binary classifier to detect the presence of coordination, the parser extended from the Berkeley Parser (Petrov et al., 2006) to generate candidate pairs, and a discriminative neural network to identify conjuncts. As similarity features, they compute the Euclidean distance between the two representations of conjuncts, which are computed from syntactic trees generated by the parser, and this is more efficient with respect to the time complexity compared with the methods with graphs. The replaceability feature vectors are produced from bidirectional LSTMs by processing two sentences that are produced by leaving out one of two conjuncts. Their model then scores all candidate pairs of conjuncts from feature vectors including similarities, replaceabilities, and additional three values derived from the probabilities assigned by the parser. The best scored pair is selected as the most probable conjuncts. For the Genia corpus, their model outperformed the method of Hara et al. (2009) which only relied on the similarity property. Using neural networks, they overcame the problems of manually elaborated features and of access to external sources such as thesauri. However, their method heavily depends on their extension of the Berkeley Parser. Therefore, the problem of error propagation between components and the parser still remains.

Kawahara and Kurohashi (2008) tried to resolve coordination disambiguation without any similarities on the basis of the dependency relations and generative probabilities of phrases including conjuncts. Yoshimoto et al. (2015) extended the graph-based dependency parsing algorithm to handle coordinations.

6 Conclusions

We propose a neural network model to disambiguate coordinate structure boundaries. Our method relies on two properties: (i) conjuncts tend to have a similar structure in syntax or semantics and (ii) conjuncts can be replaced with each other, maintaining sentence consistency. On the basis of these observations, we compute two feature vectors from a sequence of vectors produced by bidirectional RNNs. Our model can capture the connections between conjuncts and other parts of sentences and sentence-level coordination. As a result, our model outperforms existing methods and achieves state-of-the-art performance. The biggest contribution of our work is resolving dependency on information from syntactic parsers.

We plan to improve our model to handle three or more conjuncts in future work. In addition, since our method treats nested coordinate structures individually, we expect to create constraints to build non-overlapping coordination spans.

Acknowledgments

This work was partly supported by JST CREST Grant Number JPMJCR1513, Japan. We are grateful to our colleagues in the NAIST Computational Linguistics Laboratory and the anonymous reviewers for their helpful insights and comments.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Jessica Ficler. 2016. Coordination Annotation Extension in the Penn Tree Bank. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 834–842.

Jessica Ficler and Yoav Goldberg. 2016. A neural network for coordination boundary prediction. *arXiv preprint arXiv:1610.03946*.

Atsushi Hanamoto. 2012. Coordination Structure Analysis using Dual Decomposition. pages 430–438.

Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. 2009. Coordinate Structure Analysis with Global Structural Constraints and Alignment-Based Local Features. 1(August):967–975.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *EMNLP*, pages 891–896.

Daisuke Kawahara and Sadao Kurohashi. 2008. Coordination disambiguation without any similarities. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 425–432. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. Genia corpusa semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. LDC2011T07.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Masashi Shimbo and Kazuo Hara. 2007. A Discriminative Learning Model for Coordinate Conjunctions. (June):610–619.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. 2012. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*.

Akifumi Yoshimoto, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. 2015. Coordination-aware dependency parsing (preliminary report). *IWPT 2015*, page 66.