CONJR: Conjunctive Sentence Splitter without Parsing

Anonymous ACL submission

Abstract

In this paper, we observe and address the challenges of splitting conjunctive sentences around each group of conjuncts. Most existing methods rely on parsers to identify the conjuncts in a sentence and detect the coordination boundaries. However, state-of-the-art syntactic parsers are slow and suffer from errors, especially for long and complicated sentences. In order to better solve the problems, we formulate coordination boundary detection as a sequence tagging task and propose a specialized model CONJR without using syntactic parsers. We introduce both semantic and syntactic features and a specially designed attention mechanism to capture the symmetry among the potential conjuncts. The experimental results on datasets from various domains demonstrate the effectiveness of our proposed methods.

1 Introduction

Conjunction is a common syntactic phenomenon in various Natural Language Processing (NLP) corpora. Based on our counting, 39.4% of the sentences in OntoNotes Release 5.0 (Weischedel et al., 2013) contain at least one conjunctions. The frequently appeared conjunctive sentences bring many NLP tasks challenges.

It is a common practice to apply constituency parsers or dependency parsers to identify the conjunctions of a sentence and then split this conjunctive sentence around each group of the conjuncts. However, there are two drawbacks. First, the state-of-the-art syntactic parsers confront an increase of errors when processing sentences with conjunctions, especially when the input sentence contains multiple conjunctions. Second, training and applying parsers can be slow, which will make the identification of conjunctions less efficient. Existing coordination boundary detection methods rely on the results of syntactic parsers (Ficler and Goldberg, 2016, 2017; Saha and Mausam, 2018) and thus still face similar drawbacks.

In this work, we approach the coordination boundary detection problem without using syntactic parsers. We innovatively formulate coordination boundary detection as a sequence tagging task. Inspired by researches in NER tasks, we modify the BIO (Beginning-Inside-Outside) schema (Ramshaw and Marcus, 1995) based on the task characteristics of coordination boundary detection. The proposed method, CONJR (CONJunctive sentence splitteR) can detect the boundary of conjunction with more than two conjuncts, as well as handle multiple conjunctions in one sentence. We design input features with BERT contextualized token encoding, Part-of-Speech embeddings, suffix, and character embeddings, and further design a special attention mechanism to better capture the semantic and syntactic symmetry among the potential conjuncts. Empirically, we test CONJR on three datasets from both general domain and biomedical domain. The results show that the proposed CONJR consistently outperforms state-of-the-art models.

In summary, our main contributions are:

- We observe and address the challenges of splitting conjunctive sentences in the field of NLP.
- We design the coordination boundary detection task as a sequence tagging task, and propose CONJR, a specialized coordination boundary detection model without using syntactic parsers.
- We propose both semantic and syntactic features and a special attention mechanism to capture the symmetry among the potential conjuncts.
- Empirical studies on three datasets from various domains demonstrate the effectiveness of the proposed method.

2 Related Work

For the tasks of coordination boundary detection and disambiguation, earlier work designs different types of features and principles (Hogan, 2007; Shimbo and Hara, 2007; Hara et al., 2009;

Hanamoto et al., 2012; Del Corro and Gemulla, 2013). (Ficler and Goldberg, 2016) is the first to propose a neural-network-based model for coordination boundary detection. This model operates on top of the constituency parse trees, and decomposes the trees to capture the syntactic context of each word. Later, CALM is proposed by (Saha and Mausam, 2018) to improve upon the conjuncts identified from dependency parsers. CALM ranks conjunct spans based on the 'replaceability' principle and uses various linguistic constraints to additionally restrict the search space. (Teranishi et al., 2017, 2019) design similarity and replaceability feature vectors and train scoring models to evaluate the possible boundary pairs of the conjuncts. These state-of-the-art models build on syntactic parsers, and thus may inherit some of the parsers' shortcomings, such as low efficiency and suffering from errors. IGL-CA, a coordination analyzer in OpenIE6 (Kolluru et al., 2020), utilizes a novel iterative labeling-based architecture designed for OpenIE and improves the performance of coordination boundary detection task.

3 Methodology

3.1 Task Formulation and Labeling Schema

A sentence may contain multiple conjunctions. For example, one sentence may have more than two "and". Previous research (Saha and Mausam, 2018) shows that for each pair of conjunctions in a sentence, they are either non-overlapping, or one is fully contained in the other (i.e., nested). Thus in this paper we focus on one conjunction at a time. Our goal of coordination boundary detection is to find the boundary of each conjunct given a target conjunctive word in a sentence. The original multiple-conjunction sentence can be transformed into multiple input sentences, with each input sentence having exactly one conjunctive word replaced by the '[CW]' token indicating which specific conjunction is to be processed. A illustrative example can be found in Figure 1.

It can also be observed that there can be more than two conjuncts coordinated by the same conjunctive word. Therefore, the model needs to be designed to detect the coordination boundary for each conjunct. Inspired by the BIO (Beginning-Inside-Outside) labeling schema of the NER task, where entity boundaries are to detected, we use 'B' to label the beginning word and 'I' to label the inside words for each conjunct, and 'O' to label

(1)

I	like	eating	fruits	,	dancing	,	[CW]	cooking	and	my	sister	likes	running	.
O	O	B-b	I-b	I-b	B-b	I-b	CONJ	B-a	O	O	O	O	O	O

(2)

I	like	eating	fruits	,	dancing	,	and	cooking	[CW]	my	sister	likes	running	.
B-b	I-b	I-b	I-b	I-b	I-b	I-b	I-b	I-b	CONJ	B-a	I-a	I-a	I-a	O

Figure 1: Illustrative examples of input sentences and their corresponding labels

words outside the current conjunction. We add a special label for the '[CW]' token as 'CONJ'.

With the BIO and 'CONJ' label schema, we need to further incorporate the following constraints. 'B' or 'I' before the special 'CONJ' label cannot be followed by 'O', but after the 'CONJ' label they can be followed by 'O' but cannot be followed by another 'CONJ'. Therefore, to preserve the different sequential rules of labels before and after the '[CW]' token, we use 'B-before' and 'I-before' for conjuncts before the '[CW]' token, and use 'B-after' and 'I-after' for the conjunct after the '[CW]' token. The designed BIOC labeling schema is illustrated in Figure 1.

3.2 Input Features

Given a conjunctive sentence with word tokens $\{w_1, w_2, ..., w_N\}$, the input features consist of both semantic and syntactic features, including BERT contextualized token encoding, Part-of-Speech (POS) embeddings, suffix, and character embeddings, to capture the symmetry among the potential conjuncts and enhance the model performance.

BERT Contextualized Token Encoding. Coordinated conjuncts tend to have related semantic meanings. Therefore we adopt BERT (Devlin et al., 2018) token encoding to capture the semantics of the input tokens. Specifically, we use the output of the last hidden layer of BERT_{base} model to generate the token encoding. During BERT's tokenization, a word w_i may be splitted into subwords $[t_1, t_2, ..., t_k]$. Then its token encoding is:

$$\boldsymbol{ENC}(w_i) = \frac{1}{k}\sum_{j=1}^{k} enc(t_j) \tag{1}$$

Part-of-Speech Embedding. Syntactic information is another important feature of coordinated conjuncts. To capture this feature, we propose to add POS embeddings. Specifically, we run a POS tagger and get $\{pos_1, pos_2, ..., pos_N\}$ for each sentence. Then the sequence of POS tags are used to train a GloVe (Pennington et al., 2014) embedding as the POS embedding $\boldsymbol{POS} = \{v(pos_1), v(pos_2), ..., v(pos_N)\}$. It can capture the statistics of POS tag co-occurrences in the corpus and carry more than syntactic information comparing to original POS tags.

Suffix. In some of the conjunctions, the head words of the coordinated conjuncts have a similar form (Ficler and Goldberg, 2017). Thus the length of the common suffix can be a signal of symmetry, and we also implement it as an input feature of the CONJR model, represented as $\boldsymbol{SUF} = \{suf_1, suf_2, ..., suf_N\}$.

Character Embedding. Character-level compositions of the words can reflect the symmetry aspect of the coordinated conjuncts as well. Thus we use Bi-LSTM (Lample et al., 2016) to generate character-level embeddings for each token and obtain $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_N\}$.

Positional Encoding. To better capture the symmetry among the conjuncts, we propose to add the attention mechanism to draw more attention to compare words before and after the target conjunctive word ('[CW]'). Since the regular self attention mechanism (Vaswani et al., 2017) contains no information about relative positions within the sequence, we include such information by adding a relative position vector $\boldsymbol{b}_i$ for each token. Specifically, there are five important relative positions to '[CW]' token: the '[CW]' token itself, the left and right tokens adjacent to the '[CW]' token, and all other left tokens and right tokens. We use one-hot vector to indicate the relative positions.

3.3 Model Architecture

We propose a Bi-LSTM-Attn-CRF architecture for the CONJR model to predict labels based on the BIOC labeling schema defined in Section 3.1.

Bidirectional LSTM. Bi-LSTM is robust and can take advantage of context on both sides of a word (Graves, 2013). Thus we use it as an encoder of our input features. The input of Bi-LSTM is:

$$\boldsymbol{X} = [\boldsymbol{ENC}; \boldsymbol{POS}; \boldsymbol{SUF}; \boldsymbol{B}; \boldsymbol{C}] \quad (2)$$

The output of Bi-LSTM is the concatenation of its forward and backward context representations, $\boldsymbol{h} = [\overrightarrow{\boldsymbol{h}}; \overleftarrow{\boldsymbol{h}}]$.

Attention. We set queries, keys and values to be $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{h}$ and calculate the attention as:

$$Attn(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^{\boldsymbol{T}}}{\sqrt{d_k}})\boldsymbol{V} \quad (3)$$

where d_k is the dimension of queries and keys. The concatenation of Bi-LSTM and attention output $\boldsymbol{Z} = [\boldsymbol{h}; \boldsymbol{attn}]$, is feed to two linear transformations with a ReLU activation in between to add nonlinearity:

$$F(\boldsymbol{Z}) = ReLU(\boldsymbol{Z}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \quad (4)$$

Conditional Random Fields. Finally, a CRF (Lafferty et al., 2001) layer is added to ensure the constraints on the sequential rules of labels and decode the best label path in all possible label paths.

4 Experiments

4.1 Experiment Setup

Training Setup The proposed model, CONJR, is trained on the training set (WSJ 0-18) of Penn Treebank[1] (Marcus et al., 1993), and we continue following the most common split to use WSJ 19-21 for validation and WSJ 22-24 for testing. The ground truth Penn Treebank constituency parsing trees containing coordination structures (e.g., have 'CC' tag) are pre-processed to generate our special BIOC labels as follows. For each target conjunctive word, we first extract the subtrees which are at the same depth as the conjunctive word, and each of these subtrees is regarded as a conjunct coordinated by that conjunctive word. Thus we obtain the boundaries of the conjuncts for each sentence and generate labels as described in Section 3.1.

Testing Setup We use three testing datasets to evaluate the performance of the proposed CONJR model. The first testing dataset contains 10,000 randomly selected conjunctive sentences from OntoNotes Release 5.0[2] (Weischedel et al., 2013). We convert the gold standard constituency parsing results into the BIOC labels in the same way as the Penn Treebank, and we call this portion of data 'OntoNotes Test Set'. The second dataset is our manually labeled CORD-19 Test Set, which contains 768 sentences randomly selected from COVID-19 Open Research Dataset (Wang et al.,

[1]https://catalog.ldc.upenn.edu/LDC99T42

[2]https://catalog.ldc.upenn.edu/LDC2013T19

		OntoNotes			CORD-19			Penn		
Model	Training Time	P	R	F1	P	R	F1	P	R	F1
AllenNLP	hours	**76.71**	70.99	73.74	70.72	70.02	70.36	87.39	69.28	77.29
Stanford	hours	64.32	60.75	62.48	63.73	60.41	62.02	79.03	74.84	76.88
Teranishi+19	3000-3500s	-	-	-	-	-	-	73.19	73.52	73.35
IGL-CA	4800-5500s	59.03	52.39	55.51	57.98	55.00	56.45	84.35	85.21	83.50
CONJR (our)	**1600-2000s**	75.65	**75.20**	**75.42**	**72.81**	**72.89**	**72.85**	**87.83**	**87.38**	**87.60**

Table 1: Performance Comparison.

2020) and each of them has at least one conjunctive word. These sentences contain many domain-specific terms, and are longer and more complicated in structure. The release of CORD-19 Test Set can be found in the Supplementary Materials. The third dataset is Penn Treebank Test Set (WSJ 22-24) as mentioned in 4.1.

Baseline Methods We compare the proposed CONJR with two categories of baseline methods: rule-based and learning-based methods. Rule-based methods convert the constituency parsing results and regard constituents at the same depth as the target conjunctive word as conjuncts coordinated by that conjunctive word. We adopt two state-of-the-art constituency parsers, AllenNLP (Joshi et al., 2018) and Stanford (Qi et al., 2019) parsers, for this category. For learning-based methods, we choose two state-of-the-art models for coordination boundary detection, Teranishi+19 (Teranishi et al., 2019), and IGL-CA (Kolluru et al., 2020). All results are obtained using their official released code.

Evaluation Metrics We evaluate both the effectiveness and efficiency of different methods. We use precision, recall, and F1 score compared to the ground truth coordination spans to evaluate the performance in terms of effectiveness. A span is correct only if it is an exact match of the corresponding span in the ground truth.

For efficiency evaluation, we report the training time of each method. All experiments are conducted on a computer with Intel(R) Core(TM) i7-11700k 3.60GHz CPU, NVIDIA(R) RTX(TM) 3070 GPU, and 40GB memory.

4.2 Main Results

The results are shown in Table 1. In terms of effectiveness, CONJR's recall and F1 score are higher than all the baseline methods on all datasets, and the improvement on F1 scores is 1.68, 2.49, and 4.10 for OntoNotes Test Set, CORD-19 Test Set, and Penn Treebank Test Set compared to the best baseline method, respectively. Although CONJR is not trained on a biomedical corpus, it still demonstrates superior performance. These results illustrate that the proposed task formulation is reasonable and the features used in CONJR are domain-independent. The training time of CONJR is also better than all the baseline methods.

Model	Precision	Recall	F1
BERT	86.19	85.62	85.90
+POS	86.58	85.77	86.18
+suffix	87.06	86.26	86.66
+char	87.18	86.24	86.71
+attention	**87.83**	**87.38**	**87.60**

Table 2: Ablation Study

4.3 Ablation Study

In order to study the model improvement of adding different features and components, an ablation study is conducted. The base model only uses BERT token encoding as the input feature, then POS embeddings, suffix, character embeddings, and attention are incrementally added. The testing results on Penn Treebank Test Set are shown in Table 2. From the results, we can see that all of the components can improve the performance in terms of F1 score.

5 Conclusions

In this paper, we develop CONJR, a specialized model for coordination boundary detection without using syntactic parsers. We approach the problem by (1) formulating coordination boundary detection as a sequence tagging task with a special BIOC labeling schema, and (2) designing conjunction-specific features and attention mechanism. CONJR can not only detect the boundaries of more than two conjuncts for a conjunction, but also handle multiple conjunctions in one sentence. It outperforms state-of-the-art models on datasets from both general and biomedical domains.

References

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 355–366, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jessica Ficler and Yoav Goldberg. 2016. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 23–32, Austin, Texas. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Improving a strong neural parser with conjunction-specific features. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 343–348, Valencia, Spain. Association for Computational Linguistics.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Atsushi Hanamoto, Takuya Matsuzaki, and Jun'ichi Tsujii. 2012. Coordination structure analysis using dual decomposition. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 430–438, Avignon, France. Association for Computational Linguistics.

Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. 2009. Coordinate structure analysis with global structural constraints and alignment-based local features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 967–975, Suntec, Singapore. Association for Computational Linguistics.

Deirdre Hogan. 2007. Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 680–687, Prague, Czech Republic. Association for Computational Linguistics.

V. Joshi, Matthew E. Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *ACL*.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020. Openie6: Iterative grid labeling and coordination analysis for open information extraction. *CoRR*, abs/2010.03147.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2019. Universal dependency parsing from scratch. *CoRR*, abs/1901.10457.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Masashi Shimbo and Kazuo Hara. 2007. A discriminative learning model for coordinate conjunctions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 610–619, Prague, Czech Republic. Association for Computational Linguistics.

Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Coordination boundary identification with similarity and replaceability. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 264–272, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Decomposed local models for coordinate structure parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3394–3403, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.