CoRec: An Easy Approach for Coordination Recognition

Anonymous ACL submission

Abstract

In this paper, we observe and address the challenges of the coordination recognition task. Most existing methods rely on syntactic parsers to identify the coordinators 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. To better solve the problems, we propose a pipeline model COordination RECognizer (CoRec). It composes of two components: coordinator identifier and conjunct boundary detector. The experimental results on datasets from various domains demonstrate the effectiveness and efficiency of the proposed method. Further experiments show that CoRec positively impacts downstream tasks, improving the yield of state-of-the-art Open IE models.

1 Introduction

Coordination is a common syntactic phenomenon in various corpora. Based on our counting, 39.4% of the sentences in OntoNotes Release 5.0 (Weischedel et al., 2013) contain at least one coordination. The frequently appeared conjunctive sentences bring many challenges to various NLP tasks, including Natural Language Inference (NLI) (Saha et al., 2020), Named Entity Recognition (NER) (Dai et al., 2020), and text simplification (Xu et al., 2015). Specifically, in Open Information Extraction (Open IE) tasks, researchers find that ineffective processing of conjunctive sentences will result in substantial yield lost (Corro and Gemulla, 2013; Saha and Mausam, 2018; Kolluru et al., 2020), where yield is essential since Open IE tasks aim to obtain a comprehensive set of structured information. Thus processing conjunctive sentences is important to improve the performance of Open IE models.

It is a common practice to apply constituency parsers or dependency parsers to identify the coordination structures of a sentence. However, there are several drawbacks. First, the state-of-the-art syntactic parsers confront an increase of errors when processing conjunctive sentences, especially when the input sentence contains complex coordination structures. Second, applying parsers can be slow, which will make the identification of coordination 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 recognition problem without using syntactic parsers and propose a simple yet effective pipeline model COordination RECognizer (CoRec). CoRec composes of two steps: coordinator identification and conjunct boundary detection. For coordinator identification, we consider three types of coordinator spans: contiguous span coordinators (e.g. 'or' and 'as well as'), paired span coordinators (e.g. 'either...or...'), and coordination with 'respectively'. Given each identified coordinator span, we formulate the conjunct boundary detection task as a sequence labeling task and design a position-aware BIOC labeling schema based on the unique characteristics of this task. We also present a simple trick called coordinator markers that can greatly improve the model performance.

Despite CoRec's simplicity, we find it to be both effective and efficient in the empirical studies: CoRec consistently outperforms state-of-the-art models on benchmark datasets from both general domain and biomedical domain. Further experiments demonstrate that processing the conjunctive sentences with CoRec can enhance the yield of Open IE models.

In summary, our main contributions are:

- We propose a pipeline model CoRec, a specialized coordination recognizer without using syntactic parsers.
- We formulate the conjunct boundary detec-

tion task as a sequence labeling task with position-aware labeling schema.

• Empirical studies on three benchmark datasets from various domains demonstrate the efficiency and effectiveness of CoRec, and its impact on yield of Open IE models.

2 Related Work

For the tasks of coordination boundary detection and disambiguation, earlier heuristic, non-learning-based approaches design different types of features and principles based on syntactic and lexical analysis (Hogan, 2007; Shimbo and Hara, 2007; Hara et al., 2009; Hanamoto et al., 2012; Corro and Gemulla, 2013). Ficler and Goldberg (2016) are 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. Teranishi et al. (2017, 2019) design similarity and replaceability feature vectors and train scoring models to evaluate the possible boundary pairs of the conjuncts. Since these methods are designed to work on conjunct pairs, they have natural shortcomings to handle more than two conjuncts in one coordination.

Researchers in the Open Information Extraction domain also consider coordination analysis to be important to improve model performance. CALM, proposed by Saha and Mausam (2018), improves upon the conjuncts identified from dependency parsers. It ranks conjunct spans based on the 'replaceability' principle and uses various linguistic constraints to additionally restrict the search space. OpenIE6 (Kolluru et al., 2020) also has a coordination analyzer called IGL-CA, which utilizes a novel iterative labeling-based architecture. However, its labels only focus on the boundaries of the whole coordination and do not utilize the position information of the specific conjuncts.

3 Methodology

3.1 Task Formulation

Given a sentence $S = \{x_1, x_2, ..., x_n\}$, we decompose the coordination recognition task into two sub-tasks, coordinator identification and conjunct boundary detection. The coordinator identifier aims to detect all potential target coordinator spans from S. The conjunct boundary detector takes the positions of all the potential target coordinator spans as additional input and detects the conjuncts coordinated by each target coordinator span.

3.2 Label Formulation

Since the coordinator spans are usually short, we adopt simple binary labels for the coordinator identification sub-task. Tokens inside coordinator spans are labeled as 'C' and all other tokens are labeled as 'O'.

For the conjunct boundary detection sub-task, conjuncts can be long and more complicated. Thus we formulate this sub-task as a sequence labeling task. Specifically, inspired by the BIO (Beginning-Inside-Outside) (Ramshaw and Marcus, 1995) labeling schema of the NER task, we also design a position-aware labeling schema, as previous researches have shown that using a more expressive labeling schema can improve model performance (Ratinov and Roth, 2009; Dai et al., 2015).

The proposed labeling schema contains both position information for each conjunct and position information for each coordination. For each conjunct, we use 'B' to label the beginning token and 'I' to label the following tokens. For each coordination structure, we further append 'before' and 'after' tags to indicate the relative positions to the target coordinator token(s), which is/are labeled as 'C'. More details can be found in Appendix A.

3.3 Coordinator Identifier

As mentioned above, the coordinator identification sub-task is formulated as a binary classification problem. Our coordinator identifier uses a BERT (Devlin et al., 2019) encoder to encode a sentence $S = \{x_1, x_2, ..., x_n\}$, and the output is:

$$[\boldsymbol{h}_1^c, ..., \boldsymbol{h}_n^c] = Enc_1([x_1, ..., x_n]). \quad (1)$$

A linear projection layer is then added. We denote coordinator spans detected by the coordinator identifier as $P = \{p_1, p_2, ..., p_k\}$.

3.4 Conjunct Boundary Detector

The conjunct boundary detector then processes each target coordinator span $p_t \in P$ independently to find all coordinated conjuncts in sentence S.

To inject the target coordinator span information into the encoder, we insert coordinator markers, '[C]' token, before and after the target coordinator span, respectively. The resulting sequence is $S_m = \{x_1, ..., [C], p_t, [C]..., x_n\}$. For simplicity we denote $S_m = \{w_1, ..., w_m\}$.

The marked sequence S_m is fed into a BERT encoder:

$$[\boldsymbol{h}_1^{cbd}, ..., \boldsymbol{h}_m^{cbd}] = Enc_2([w_1, ..., w_m]). \quad (2)$$

The position information of all the coordinators found by the coordinator identifier can help the model to understand the sentence structure. Thus we encode such information into a vector $\boldsymbol{b_i}$ to indicate if w_i is part of a detected coordinator span. Given $w_i \in S_m$, we concatenate its encoder output and coordinator position encoding as $\boldsymbol{h}_i^o = [\boldsymbol{h}_i^{cbd}; \boldsymbol{b}_i]$.

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

3.5 Training & Inference

The coordinator identifier and the conjunct boundary detector are trained using task-specific losses. For both, we fine-tune the two pre-trained BERT$_{base}$ encoders. Specifically, we use cross-entropy loss:

$$\mathcal{L}_c = -\sum_{x_i \in S} logP_c(t_i^*|x_i), \quad (3)$$

$$\mathcal{L}_{cbd} = -\sum_{w_i \in S_m} logP_{cbd}(z_i^*|w_i), \quad (4)$$

where t_i^*, z_i^* represent the ground truth labels. During inference, we first apply the coordinator identifier and obtain:

$$y_c(x_i) = \underset{t_i \in T}{\text{argmax}}\, P_c(t_i|x_i). \quad (5)$$

Then we use its prediction $y_c(x_i)$ with the original sentence as input to the conjunct boundary detector and obtain:

$$\boldsymbol{y} = \underset{[z_1,...,z_m], z_i \in Z}{\text{argmax}}\, P_{cbd}([z_1, ..., z_m]|[w_1, ..., w_m]), \quad (6)$$

where T and Z represent the set of possible labels of each model respectively.

3.6 Data Augmentation

We further automatically augment the training data. The new sentences are generated following the the symmetry rule, by switching the first and last conjuncts of each original training sentence. Since all sentences are augmented once, the new data distribution only slightly differs from the original one, which will not lead to a deterioration in performance (Xie et al., 2020).

4 Experiments

Training Setup The proposed CoRec is trained on the training set (WSJ 0-18) of Penn Treebank[1] (Marcus et al., 1993) following the most common split, and WSJ 19-21 are used for validation and WSJ 22-24 for testing. The ground truth constituency parse trees containing coordination structures are pre-processed to generate labels for the two sub-tasks as follows. If a constituent is tagged with 'CC' or 'CONJP', then it is considered a coordinator span. For each coordinator span, we first extract the constituents which are siblings to the coordinator span, and each constituent is regarded as a conjunct coordinated by that coordinator span. We automatically generate labels as described in Section 3.2. We also manually check and correct labels for complicated cases.

Testing Setup We use three testing datasets to evaluate the performance of the proposed CoRec model. The first dataset, ontoNotes, contains 1,000 randomly selected conjunctive sentences from the English portion of OntoNotes Release 5.0[2] (Weischedel et al., 2013). The second dataset, Genia, contains 802 conjunctive sentences from the testing set of GENIA[3] (Ohta et al., 2002), a benchmark dataset from biomedical domain. The third dataset, Penn, contains 1,625 conjunctive sentences from Penn Treebank testing set (WSJ 22-24). These three datasets contain the gold standard constituency parsing annotations. We convert them into the OC and BIOC labels in the same way as described in 4.

Baseline Methods We compare the proposed CoRec with two categories of baseline methods: parsing-based and learning-based methods. Parsing-based methods convert the constituency parsing results and regard constituents at the same depth with the target coordinator spans as coordinated conjuncts. 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.

[1] https://catalog.ldc.upenn.edu/LDC99T42
[2] https://catalog.ldc.upenn.edu/LDC2013T19
[3] http://www.geniaproject.org/genia-corpus/treebank

	ontoNotes (Simple)				**Genia** (Simple)				**Penn** (Simple)			
Model	**P**	**R**	**F1**	**Time**	**P**	**R**	**F1**	**Time**	**P**	**R**	**F1**	**Time**
AllenNLP	74.2	68.4	71.2	452s	79.7	47.7	59.7	1059s	**88.7**	67.7	76.8	823s
Stanford	56.9	53.4	55.1	763s	73.8	72.2	73.0	1722s	81.8	79.3	80.6	1387s
Teranishi+19	68.3	60.8	64.7	167s	76.4	65.2	70.3	136 s	74.2	75.5	75.4	217s
IGL-CA	**77.6**	59.7	67.5	**17s**	78.0	64.3	71.0	27s	87.9	86.9	87.4	**17s**
CoRec (our)	72.4	**75.8**	**74.1**	32s	**82.0**	**81.2**	**81.6**	**15s**	88.3	**89.2**	**88.8**	57s
	ontoNotes (Complex)				**Genia** (Complex)				**Penn** (Complex)			
AllenNLP	**84.6**	49.4	62.4	105s	**82.3**	25.7	39.2	370s	90.2	61.7	73.2	363 s
Stanford	62.4	34.5	44.4	248 s	64.3	32.9	43.5	831s	86.1	69.7	77.1	530 s
CoRec (our)	73.1	**79.3**	**76.0**	**4s**	67.7	**56.7**	**61.7**	**9s**	**91.5**	**89.5**	**90.5**	**10s**

Table 1: Performance Comparison (average over 5 runs)

Evaluation Metrics We evaluate both the effectiveness and efficiency of different methods. We evaluate effectiveness using span level precision, recall, and F1 scores. A predicted span is correct only if it is an exact match of the corresponding span in the ground truth.

For efficiency evaluation, we report the inference 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.

4.1 Main Results

The results are shown in Table 1. Note that each test set is further split into a simple set and a complex set. Simple set contains instances with 'and', 'but', 'or' as target coordinators only. The remaining instances go into the complex set. The learning-based baseline methods cannot handle the complex set. In terms of effectiveness, CoRec's recall and F1 scores are higher than all baseline methods on all datasets, and the improvement on F1 scores is 2.9, 8.6, and 1.4 for ontoNotes, Genia, and Penn compared to the best baseline methods, respectively. Although CoRec is not trained on a biomedical corpus, it still demonstrates superior performance. The inference time of CoRec is also competitive.

4.2 Impact of CoRec on Open IE Tasks

To show the effect of CoRec on downstream tasks, we implement a sentence splitter that generates simple, non-conjunctive sub-sentences based on CoRec's output. Then we apply two state-of-the-art Open IE models, Stanford OpenIE (Angeli et al., 2015) and OpenIE6 (Kolluru et al., 2020), to extract unique relation triples on the Penn dataset before and after our sentence splitting. The results are shown in Table 2. The yield of unique extractions has a significant increase for both models after sentence splitting. Though OpenIE6 implements the coordination boundary detection method IGL-CA, the coordination structure still negatively impacts the Open IE yield.

Model	**Before**	**After**	**Yield**
Stanford	12210	21284	+74.3%
OpenIE6	8084	12085	+58.4%

Table 2: The impact of CoRec on Open IE Yield

Model	**Precision**	**Recall**	**F1**
BERT	78.73	85.46	81.96
+[C] mark	87.15	88.92	88.03
+CRF	88.36	89.35	88.85
+aug	**89.28**	**90.21**	**89.74**

Table 3: Ablation Study

4.3 Ablation Study

We conduct an ablation study to examine the contribution of different components in terms of performance gain. The base model only uses BERT encoder, then coordinator markers, CRF, and data augmentation are incrementally added. The testing results on Penn dataset are shown in Table 3. From the results, we can see that all of the components can improve the performance in terms of precision and recall.

5 Conclusions

In this paper, we develop CoRec, a simple yet effective coordination recognizer without using syntactic parsers. We approach the problem by formulating a pipeline of coordinator identification and conjunct boundary detection. CoRec can not only detect the boundaries of more than two coordinated conjuncts but also handle multiple coordination in one sentence. It can also deal with both simple and complex cases of coordination. Experimental results show that CoRec outperforms state-of-the-art models on datasets from various domains. Further experiments imply that CoRec can improve the performance of state-of-the-art Open IE models.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366. International World Wide Web Conferences Steering Committee / ACM.

Hong-Jie Dai, Po-Ting Lai, Yung-Chun Chang, and Richard Tzong-Han Tsai. 2015. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *J. Cheminformatics*, 7(S-1):S14.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cécile Paris. 2020. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5860–5870. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 23–32. The 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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 343–348. Association for Computational Linguistics.

Atsushi Hanamoto, Takuya Matsuzaki, and Jun'ichi Tsujii. 2012. Coordination structure analysis using dual decomposition. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*, pages 430–438. The Association for Computer Linguistics.

Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. 2009. Coordinate structure analysis with global structural constraints and alignment-based local features. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 967–975. The Association for Computer Linguistics.

Deirdre Hogan. 2007. Coordinate noun phrase disambiguation in a generative parsing model. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Vidur Joshi, Matthew E. Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1190–1199. Association for Computational Linguistics.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020. Openie6: Iterative grid labeling and coordination analysis for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3748–3761. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19(2):313–330.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 82–86, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Lance A. Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995*.

Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*, pages 147–155. ACL.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2288–2299. Association for Computational Linguistics.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. Conjnli: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8240–8252. Association for Computational Linguistics.

Masashi Shimbo and Kazuo Hara. 2007. A discriminative learning model for coordinate conjunctions. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 610–619. ACL.

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, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 264–272. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3394–3403. 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.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. 2020. Adversarial examples improve image recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 816–825. Computer Vision Foundation / IEEE.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Trans. Assoc. Comput. Linguistics*, 3:283–297.

A Labeling Details

CoRec handles three types of coordinator spans: contiguous span coordinators, paired span coordinators, and coordination with 'respectively'. In this section, we use simple examples to show the labeling details of each type.

Contiguous Span Coordinators Processing contiguous span coordinator is straightforward. Take the sentence "*My sister likes apples, pears, and grapes.*" as an example, following Section 3.2 we should generate one instance with labels as shown in Table 4.

Paired Span Coordinators Each paired span coordinator consists of two coordinator spans: the left coordinator span, which appears at the beginning of the coordination, and the right coordinator span, which appears in the middle. The right coordinator span stays more connected with the conjuncts due to relative position. Therefore, we choose to detect the conjuncts only when targeting the right coordinator span. Take the sentence "*She can have either green tea or hot chocolate.*" as an example, following Section 3.2 we should generate two instances with labels as shown in Table 5.

Coordination with 'Respectively' The conjunctive sentence with 'respectively' usually has the structure '...and...and...respectively...', where the first and the second coordination have the same number of conjuncts. In order not to confuse CoRec, we process the sentence as an ordinary contiguous span instance when targeting the first or second 'and'. When targeting 'respectively', we aim to detect conjunct boundaries of the first and second coordination together. Thus we can match the conjuncts one by one during sentence splitting. Take the sentence "*The dog and the cat were named Jack and Sam respectively.*" as an example, following Section 3.2 we should generate three instances with labels as shown in Table 6.

B Error Analysis

To better understand the bottleneck of CoRec, we conduct a case study to investigate the errors that CoRec makes. We randomly selected 50 wrong predictions and analyzed their reasons. We identify four major types of errors as follows (for detailed examples check Table 7):

Ambiguous Boundaries (38%) The majority of the errors occurred when the detected boundaries are ambiguous. In this case, although our prediction is different from the gold standard result, they can both be treated as true. We identify two common types of ambiguous boundaries: ambiguous shared heads (28%) and ambiguous shared tails (10%). The former is usually signaled by 'a/an/the' and shared modifiers. The latter is usually signaled by prepositional phrases.

Errors without Obvious Reasons (32%) Many errors occurred without obvious reasons. However, we observe that CoRec makes more mistakes when the original sentences contain a large amount of certain symbols (e.g., '-', '.').

Wrong Number of Conjuncts (22%) Sometimes CoRec detects most conjuncts in the gold standard set but misses a few conjuncts. In some other cases, CoRec would detect additional conjuncts to the correct conjuncts.

Low-Quality Gold Labels (8%) We find there may also be some low-quality ground truth parse trees, thus generating incorrect gold labels. In this case CoRec may make a correct prediction that is different from the ground truth.

C Limitations

Language Limitation The proposed CoRec model works mostly for languages with limited morphology, like English. Our conclusions may not be generalized to all languages.

Label Quality Limitation We use ground truth constituency parse trees from Penn Treebank, GENIA, and OntoNotes Release 5.0 (Marcus et al., 1993; Weischedel et al., 2013; Ohta et al., 2002) to generate the labels. Since the parsing does not target for the coordination recognition task, we apply rules for the conversion. A single author inspected the labels for complicated cases but did not inspect all the labels. There could be erroneous labels in the training and testing data.

Comparison Limitation Comparison to the parsing-based methods may not be precise. Parsers are not specialized in the coordination recognition task. Our task and datasets may not be the best fit for their models.

My	sister	likes	apples	,	pears	,	[C]	and	[C]	grapes	.
O	O	O	B-before	I-before	B-before	I-before	C	C	C	B-after	O

Table 4: An example conjunctive sentence labeling with contiguous span coordinator 'and'

She	can	have	[C]	either	[C]	green	tea	or	hot	chocolate	.
O	O	O	C	C	C	O	O	O	O	O	O
She	can	have	either	green	tea	[C]	or	[C]	hot	chocolate	.
O	O	O	C	B-before	I-before	C	C	C	B-after	I-after	O

Table 5: An example conjunctive sentence labeling with paired span coordinator 'either...or...'

The	dog	[C]	and	[C]	the	cat	were	named	Jack	and	Sam	respectively	.
B-before	I-before	C	C	C	B-after	I-after	O	O	O	O	O	O	
The	dog	and	the	cat	were	named	Jack	[C]	and	[C]	Sam	respectively	.
O	O	O	O	O	O	O	B-before	C	C	C	B-after	O	
The	dog	and	the	cat	were	named	Jack	and	Sam	[C]	respectively	[C]	.
B-before	I-before	C	B-after	I-after	O	O	B-before	C	B-after	C	C	C	O

Table 6: An example conjunctive sentence labeling with 'respectively'

Category	**Ground Truth**	**CoRec**
Ambiguous Boundaries	I'm not going to worry about one day's decline, said Kenneth Olsen, digital equipment corp. president, who was leisurely strolling through the bright [orange] and [yellow] leaves of the mountains here after his company's shares plunged $5.75 to close at $86.5.	I'm not going to worry about one day's decline, said Kenneth Olsen, digital equipment corp. president, who was leisurely strolling through [the bright orange] and [yellow] leaves of the mountains here after his company's shares plunged $5.75 to close at $86.5.
Errors without Obvious Reasons	For example, the delay in selling people's heritage savings, Salina Kan, with $1.7 billion in assets, has forced the RTC to consider selling off the thrift [branch-by-branch,] instead of [as a whole institution].	For example, the delay in selling people's heritage savings, Salina Kan, with $1.7 billion in assets, has forced the RTC to consider [selling off the thrift branch-by-branch,] instead of [as a whole institution].
Wrong Number of Conjuncts	Sales of Pfizer's important drugs, [Feldene for treating arthritis,] and [Procardia, a heart medicine], have shrunk because of increased competition.	Sales of [Pfizer's important drugs,] [Feldene for treating arthritis,] and [Procardia, a heart medicine], have shrunk because of increased competition.
Low-Quality Gold Labels	The executives were remarkably unperturbed by the plunge even though it [lopped billions of dollars off the value of their companies] and [millions off their personal fortunes].	The executives were remarkably unperturbed by the plunge even though it lopped [billions of dollars off the value of their companies] and [millions off their personal fortunes].

Table 7: Case study of conjunct boundary detection results on the Penn dataset. For each case, the ground truth conjuncts are colored red and the CoRec detected conjuncts are colored blue.