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.

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; Ficler and Goldberg, 2016, 2017; Saha and Mausam, 2018) and thus still face similar drawbacks.
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 conjunctions 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 conjunction 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 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 split into subwords \( [t_1, t_2, ..., t_k] \). Then its token encoding is:

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

**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

(1) I like eating fruits, dancing, [CW] cooking

\[
O \quad O \quad B-b \quad I-b \quad I-b \quad B-b \quad I-b \quad CONJ \quad B-a
\]

and my sister likes running.

\[
O \quad O \quad O \quad O \quad O \quad O \quad O
\]

(2) I like eating fruits, dancing, and cooking

\[
B-b \quad I-b \quad I-b \quad I-b \quad L-b \quad I-b \quad L-b
\]

[CW] my sister likes running.

\[
CONJ \quad B-a \quad I-a \quad I-a \quad I-a \quad O
\]
each sentence. Then the sequence of POS tags are used to train a GloVe (Pennington et al., 2014) embedding as the POS embedding \( POS = \{v(\text{pos}_1), v(\text{pos}_2), \ldots, v(\text{pos}_N)\} \). It can capture the statistics of POS tag co-occurrences in the corpus and carry more than syntactic information compared 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 \( \text{SUF} = \{\text{suf}_1, \text{suf}_2, \ldots, \text{suf}_N\} \).

**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 \( C = \{c_1, c_2, \ldots, c_N\} \).

**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 \( b_i \) 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:

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

The output of Bi-LSTM is the concatenation of its forward and backward context representations, \( h = [\bar{h}; \bar{h}] \).

**Attention.** We set queries, keys and values to be \( Q = K = V = h \) and calculate the attention as:

\[
\text{Attn}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (3)
\]

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

\[
F(Z) = \text{ReLU}(ZW_1 + b_1)W_2 + b_2 \quad (4)
\]

**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., 2021).

\^1https://catalog.ldc.upenn.edu/LDC99T42
\^2https://catalog.ldc.upenn.edu/LDC2013T19
Table 1: Performance Comparison.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Time</th>
<th>OntoNotes</th>
<th></th>
<th>CORD-19</th>
<th></th>
<th>Penn</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AllenNLP</td>
<td>hours</td>
<td>76.71</td>
<td>70.99</td>
<td>73.74</td>
<td>70.72</td>
<td>70.02</td>
<td>70.36</td>
</tr>
<tr>
<td>Stanford</td>
<td>hours</td>
<td>64.32</td>
<td>60.75</td>
<td>62.48</td>
<td>63.73</td>
<td>60.41</td>
<td>62.02</td>
</tr>
<tr>
<td>Teranishi+19</td>
<td>3000-3500s</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IGL-CA</td>
<td>4800-5500s</td>
<td>59.03</td>
<td>52.39</td>
<td>55.51</td>
<td>57.98</td>
<td>55.00</td>
<td>56.45</td>
</tr>
<tr>
<td>CONJR (our)</td>
<td>1600-2000s</td>
<td>75.65</td>
<td>75.20</td>
<td>75.42</td>
<td>72.81</td>
<td>72.89</td>
<td>72.85</td>
</tr>
</tbody>
</table>

Table 2: Ablation Study

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>86.19</td>
<td>85.62</td>
<td>85.90</td>
</tr>
<tr>
<td>+POS</td>
<td>86.58</td>
<td>85.77</td>
<td>86.18</td>
</tr>
<tr>
<td>+suffix</td>
<td>87.06</td>
<td>86.26</td>
<td>86.66</td>
</tr>
<tr>
<td>+char</td>
<td>87.18</td>
<td>86.24</td>
<td>86.71</td>
</tr>
<tr>
<td>+attention</td>
<td>87.83</td>
<td>87.38</td>
<td>87.60</td>
</tr>
</tbody>
</table>

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.

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


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


