Structured Tree Alignment for Evaluation of (Speech) Constituency Parsing

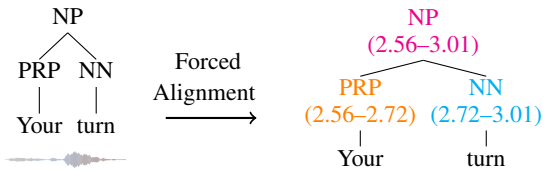
Freda Shi **Kevin Gimpel** **Karen Livescu**
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave, Chicago, IL, USA, 60637
{freda,kgimpel,klivescu}@ttic.edu

Abstract

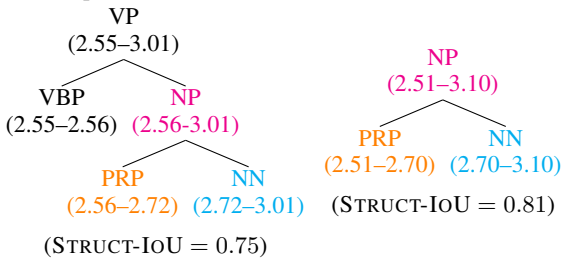
We present the structured average intersection-over-union ratio (STRUCT-IOU), a similarity metric between constituency parse trees motivated by the problem of evaluating speech parsers. STRUCT-IOU enables comparison between a constituency parse tree (over automatically recognized spoken word boundaries) with the ground-truth parse (over written words). To compute the metric, we project the ground-truth parse tree to the speech domain by forced alignment, align the projected ground-truth constituents with the predicted ones under certain structured constraints, and calculate the average IOU score across all aligned constituent pairs. STRUCT-IOU takes word boundaries into account and overcomes the challenge that the predicted words and ground truth may not have perfect one-to-one correspondence. Extending to the evaluation of text constituency parsing, we demonstrate that STRUCT-IOU shows higher tolerance to syntactically plausible parses than PARSEVAL (Black et al., 1991).[1]

1 Introduction

Automatic constituency parsing of written text (Marcus et al., 1993, *inter alia*) and speech transcriptions (Godfrey and Holliman, 1993, *inter alia*), as representative tasks of automatic syntactic analysis, have been widely explored in the past few decades. Appropriate evaluation metrics have facilitated the comparison and benchmarking of different approaches: the PARSEVAL F_1 score (Black et al., 1991; Sekine and Collins, 1997) has served as a reliable measure of text parsing across various scenarios; for speech transcription parsing, SPARSEVAL (Roark et al., 2006), which extends PARSEVAL and accounts for speech recognition errors by allowing word-level editing with a cost, has been commonly used.



(a) Ground-truth speech parse tree (right), obtained by forced alignment between the ground-truth text parse tree (left, top) and the spoken utterance (left, bottom).



(b) Predicted tree with good word boundaries and an errorful tree structure (left), or that with errorful word boundaries and a perfect tree structure (right).

Figure 1: Illustration of how STRUCT-IOU (§§ 4.1 and 4.2) evaluates textless speech constituency parsing. Best viewed in color, where nodes with the same color are aligned. Numbers in parentheses are the starting and ending times of the corresponding spans (in seconds).

Recent work (Lai et al., 2023; Tseng et al., 2023) has proposed the related task of textless speech constituency parsing. In contrast to earlier work that parses manually labeled (Charniak and Johnson, 2001, *inter alia*) or automatic (Kahn and Ostendorf, 2012, *inter alia*) speech transcriptions, these models construct constituency parse trees over automatically recognized spoken word boundaries, where each word is represented with a time range of the spoken utterance, without using any form of text. To evaluate these textless models, we need a metric that compares the predicted tree (over spoken word boundaries) with the manually labeled ground-truth tree (over written words) and faithfully reflects the parsing quality. Since the automatically recognized word boundaries may be imperfect, the metric should also reflect the changes

[1]We open-source the code of STRUCT-IOU at https://github.com/ExplorerFreda/struct-iou.

in parsing quality due to word boundary errors. To the best of our knowledge, none of the existing metrics meets these requirements, as they are all designed to compare parse trees over discrete word sequences, instead of continuous time ranges.

Motivated by the need for textless speech parsing evaluation, in this paper, we introduce the structured average intersection-over-union ratio (STRUCT-IOU; Figure 1), a metric that compares two parse trees over time ranges. We relax the definition of segment trees (Bentley, 1977) to represent speech constituency parse trees, where each node is associated with an interval that represents the time range of the corresponding spoken word or constituent. To obtain the "ground-truth" speech parse trees, we use the forced alignment algorithm (McAuliffe et al., 2017), a supervised and highly accurate method that aligns written words to time ranges of the corresponding spoken utterance, to project the ground-truth text parses onto the time domain. STRUCT-IOU is calculated by aligning the same-label nodes in the predicted and ground-truth parse trees, following structured constraints that preserve parent-child relations. The calculation of STRUCT-IOU can be formulated as an optimization problem (§4.1) with a polynomial-time solution (§4.2) in terms of the number of tree nodes.

Although STRUCT-IOU is designed to evaluate speech parsing, it is also applicable to text parsing evaluation. We analyze STRUCT-IOU for both purposes: in speech parsing evaluation, STRUCT-IOU robustly takes into account both the structure information and word boundaries; in text parsing evaluation, while maintaining a high correlation with the PARSEVAL F_1 score, STRUCT-IOU shows a higher tolerance to potential syntactic ambiguity.

2 Related Work

Text constituency parsing and evaluation. In the past decades, there has been much effort in building and improving constituency parsing models (Collins and Koo, 2005; Charniak and Johnson, 2005; McClosky et al., 2006; Durrett and Klein, 2015; Cross and Huang, 2016; Dyer et al., 2016; Choe and Charniak, 2016; Stern et al., 2017; Kitaev and Klein, 2018, *inter alia*). PARSEVAL (Black et al., 1991) has been the standard evaluation metric for constituency parsing in most scenarios, which takes the ground truth and predicted trees and calculates the harmonic mean of precision and recall of labeled spans. For morphologically rich languages, TEDEVAL (Tsarfaty et al., 2012) extends PARSEVAL to accept multiple morphological analyses over sequences of words. All of these metrics are designed to evaluate parses over discrete word sequences, and cannot be easily extended to evaluate speech parses over continuous time ranges. Although our metric, STRUCT-IOU, is designed to evaluate speech constituency parsing, it can be easily extended for text parsing evaluation, reflecting a different aspect from existing metrics (§5.2).

Speech constituency parsing and its evaluation. Work on conversational speech parsing has focused on addressing the unique challenges posed by speech, including speech recognition errors (Kahn and Ostendorf, 2012; Marin and Ostendorf, 2014), unclear sentence boundaries (Kahn et al., 2004), disfluencies (Jamshid Lou and Johnson, 2020; Kahn et al., 2005; Lease and Johnson, 2006), as well as integrating prosodic features into the parsing systems (Tran et al., 2018; Tran and Ostendorf, 2021). On the evaluation side, the closest work to ours is SPARSEVAL (Roark et al., 2006), which extends PARSEVAL to account for speech recognition errors by allowing for word-level insertion, deletion and substitution. In contrast, our metric STRUCT-IOU applies to the cases where no speech recognizer is applied or available.

Other structured evaluation metrics for parsing. There have been evaluation metrics of abstract meaning representations (AMRs; Cai and Knight, 2013), where two AMR graphs are matched by solving an NP-complete integer linear programming problem. While our work shares the spirit with theirs, we focus on the evaluation of speech constituency parsing over continuous word boundaries. There also exists a polynomial-time exact solution to our optimization problem.

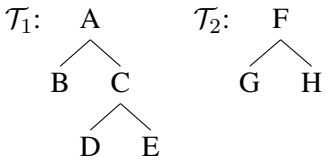
3 Preliminaries

We use real-valued open intervals to represent speech spans for simplicity, although most of the following definitions and conclusions can be easily extended to closed intervals and half-open intervals. Proof of each corollary and proposition can be found in Appendix A.

3.1 Open Intereval Operations

Definition 1. The **length** of a real-valued open interval $I = (a, b)$, where $a < b$, is $|I| = b - a$.

Definition 2. The **intersection size** of open inter-



A $\leftrightarrow$ G and E $\leftrightarrow$ H: conflicted (violating rule 10.1)
C $\leftrightarrow$ F and A $\leftrightarrow$ H: conflicted (violating rules 10.2 and 10.3)
B $\leftrightarrow$ H and C $\leftrightarrow$ F: conflicted (violating rule 10.4)
A $\leftrightarrow$ F and C $\leftrightarrow$ H: not conflicted

Figure 2: Examples of conflicted and non-conflicted node matchings (Definition 10).

vals I_1 and I_2 is

$$\mathcal{I}(I_1, I_2) = \begin{cases} 0 & \text{if } I_1 \cap I_2 = \emptyset \\ |I_1 \cap I_2| & \text{otherwise.} \end{cases}$$

Definition 3. The **union size** of open intervals I_1 and I_2 is $\mathcal{U}(I_1, I_2) = |I_1| + |I_2| - \mathcal{I}(I_1, I_2)$.

Definition 4. The **intersection over union** (IoU) ratio between open intervals I_1 and I_2 is

$$\text{IoU}(I_1, I_2) = \frac{\mathcal{I}(I_1, I_2)}{\mathcal{U}(I_1, I_2)}.$$

Throughout this paper, we will use IoU as the similarity metric between two intervals.

3.2 Relaxed Segment Trees

We relax the definition of a segment tree (Bentley, 1977) as follows to represent parse trees.

Definition 5. A **node** $\boldsymbol{n}$ of a relaxed segment tree is a triple $\boldsymbol{n} = \langle I_{\boldsymbol{n}}, C_{\boldsymbol{n}}, \ell_{\boldsymbol{n}} \rangle$, where

1. $I_{\boldsymbol{n}} = (s_{\boldsymbol{n}}, e_{\boldsymbol{n}})$ is an open interval (i.e., segment) associated with the node $\boldsymbol{n}$, where $s_{\boldsymbol{n}} < e_{\boldsymbol{n}}$;
2. $C_{\boldsymbol{n}}$ is a finite set of disjoint children nodes of $\boldsymbol{n}$: for any $\boldsymbol{p}, \boldsymbol{q} \in C_{\boldsymbol{n}} (\boldsymbol{p} \neq \boldsymbol{q})$, $I_{\boldsymbol{p}} \cap I_{\boldsymbol{q}} = \emptyset$. $C_{\boldsymbol{n}} = \emptyset$ if and only if $\boldsymbol{n}$ is a terminal node;
3. For a nonterminal node $\boldsymbol{n}$, $s_{\boldsymbol{n}} = \min_{\boldsymbol{p} \in C_{\boldsymbol{n}}} s_{\boldsymbol{p}}$, and $e_{\boldsymbol{n}} = \max_{\boldsymbol{p} \in C_{\boldsymbol{n}}} e_{\boldsymbol{p}}$.

Corollary 5.1. *For nodes $\boldsymbol{p}, \boldsymbol{n}$, if $\boldsymbol{p} \in C_{\boldsymbol{n}}$, then $I_{\boldsymbol{p}} \subseteq I_{\boldsymbol{n}}$.*

Definition 6. Node $\boldsymbol{p}$ is an **ancestor** of node $\boldsymbol{q}$ if there exists a sequence of nodes $\boldsymbol{n}_0, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_k (k \geq 1)$ such that (i.) $\boldsymbol{n}_0 = \boldsymbol{p}$, (ii.) $\boldsymbol{n}_k = \boldsymbol{q}$, and (iii.) or any $i \in [k]$,[2] $\boldsymbol{n}_i \in C_{\boldsymbol{n}_{i-1}}$.

Corollary 6.1. *If node $\boldsymbol{p}$ is an ancestor of node $\boldsymbol{q}$, then $I_{\boldsymbol{p}} \supseteq I_{\boldsymbol{q}}$.*

Definition 7. Node $\boldsymbol{p}$ is a **descendant** of node $\boldsymbol{q}$ if $\boldsymbol{q}$ is an ancestor of $\boldsymbol{p}$.

Definition 8. An **relaxed segment tree** $\mathcal{T} = \langle \boldsymbol{r}_{\mathcal{T}}, N_{\mathcal{T}} \rangle$ is a tuple, where

1. $\boldsymbol{r}_{\mathcal{T}}$ is the root node of $\mathcal{T}$;
2. $N_{\mathcal{T}} = \{\boldsymbol{r}_{\mathcal{T}}\} \cup \{\boldsymbol{n} : \boldsymbol{n} \text{ is a descendant of } \boldsymbol{r}_{\mathcal{T}}\}$ is a finite set of all nodes in $\mathcal{T}$.

Example 8.1. A constituency parse tree over spoken word time ranges (Figure 1a) can be represented by a relaxed segment tree.

Corollary 8.1. *A relaxed segment tree can be uniquely characterized by its root node.*

In the following content, we use $\mathcal{T}(\boldsymbol{n})$ to denote the relaxed segment tree rooted at $\boldsymbol{n}$.

Proposition 9. *For a relaxed segment tree $\mathcal{T}$ and $\boldsymbol{p}, \boldsymbol{q} \in N_{\mathcal{T}}$, $\boldsymbol{p}$ is neither an ancestor nor a descendant of $\boldsymbol{q} \Leftrightarrow I_{\boldsymbol{p}} \cap I_{\boldsymbol{q}} = \emptyset$.*

4 The STRUCT-IoU Metric

4.1 Problem Formulation

Given relaxed segment trees $\mathcal{T}_1$ and $\mathcal{T}_2$ with node sets $N_{\mathcal{T}_1} = \{\boldsymbol{n}_{1,i}\}_{i=1}^{|N_{\mathcal{T}_1}|}$ and $N_{\mathcal{T}_2} = \{\boldsymbol{n}_{2,j}\}_{j=1}^{|N_{\mathcal{T}_2}|}$, we can align the trees by matching their same-label nodes. Let $\boldsymbol{n}_{1,i} \leftrightarrow \boldsymbol{n}_{2,j}$ denote the matching between the nodes $\boldsymbol{n}_{1,i}$ and $\boldsymbol{n}_{2,j}$.

Definition 10 (conflicted node matchings; Figure 2)**.** The matchings $\boldsymbol{n}_{1,i} \leftrightarrow \boldsymbol{n}_{2,j}$ and $\boldsymbol{n}_{1,k} \leftrightarrow \boldsymbol{n}_{2,\ell}$ are *conflicted* if any of the following conditions holds:

1. $\boldsymbol{n}_{1,i}$ is an ancestor of $\boldsymbol{n}_{1,k}$, and $\boldsymbol{n}_{2,j}$ is not an ancestor of $\boldsymbol{n}_{2,\ell}$;
2. $\boldsymbol{n}_{1,i}$ is not an ancestor of $\boldsymbol{n}_{1,k}$, and $\boldsymbol{n}_{2,j}$ is an ancestor of $\boldsymbol{n}_{2,\ell}$;
3. $\boldsymbol{n}_{1,i}$ is a descendant of $\boldsymbol{n}_{1,k}$, and $\boldsymbol{n}_{2,j}$ is not a descendant of $\boldsymbol{n}_{2,\ell}$;
4. $\boldsymbol{n}_{1,i}$ is not a descendant of $\boldsymbol{n}_{1,k}$, and $\boldsymbol{n}_{2,j}$ is a descendant of $\boldsymbol{n}_{2,\ell}$.

Intuitively, we would like the alignment to be consistent with the ancestor-descendant relationship between nodes.

The optimal (i.e., maximally IoU-weighted) structured alignment between $\mathcal{T}_1$ and $\mathcal{T}_2$ is given by the solution to the following problem:

Problem 11 (maximally IoU-weighted alignment)**.**

[2] $[k] = \{1, 2, \ldots, k\}$, where $k \in \mathbb{N}$.

$$\boldsymbol{A}^* = \arg\max_{\boldsymbol{A}} \sum_{i=1}^{|N_{\mathcal{T}_1}|} \sum_{j=1}^{|N_{\mathcal{T}_2}|} a_{i,j} \operatorname{IoU}\left(I_{\boldsymbol{n}_{1,i}}, I_{\boldsymbol{n}_{2,j}}\right)$$

$$\text{s.t.} \sum_j a_{i,j} \leq 1 (\forall i \in [|N_{\mathcal{T}_1}|]), \tag{1}$$

$$\sum_i a_{i,j} \leq 1 (\forall j \in [|N_{\mathcal{T}_2}|]), \tag{2}$$

$$a_{i,j} + a_{k,\ell} \leq 1$$
if $\boldsymbol{n}_{1,i} \leftrightarrow \boldsymbol{n}_{2,j}$ and $\boldsymbol{n}_{1,k} \leftrightarrow \boldsymbol{n}_{2,\ell}$ are conflicted.

$\boldsymbol{A} \in \{0,1\}^{|N_{\mathcal{T}_1}| \times |N_{\mathcal{T}_2}|}$ denotes an alignment matrix: $a_{i,j} = 1$ indicates that the matching $\boldsymbol{n}_{1,i} \leftrightarrow \boldsymbol{n}_{2,j}$ is selected, otherwise $a_{i,j} = 0$. The last constraint of Problem 11 ensures that there are no conflicted matchings selected. Equations (1) and (2) imply one-to-one matching between nodes; that is, in a valid tree alignment, each node in $\mathcal{T}_1$ can be matched with at most one node in $\mathcal{T}_2$, and vice versa. The solution to Problem 11 gives the maximal possible sum of IoU over aligned node pairs.

Definition 12. The **structured average IoU** (STRUCT-IOU) between $\mathcal{T}_1$ and $\mathcal{T}_2$ is given by

$$\overline{\operatorname{IoU}}(\mathcal{T}_1, \mathcal{T}_2) = \frac{1}{|N_{\mathcal{T}_1}| + |N_{\mathcal{T}_2}|} \sum_{i=1}^{|N_{\mathcal{T}_1}|} \sum_{j=1}^{|N_{\mathcal{T}_2}|} a^*_{i,j} \operatorname{IoU}\left(I_{\boldsymbol{n}_{1,i}}, I_{\boldsymbol{n}_{2,j}}\right),$$

where $\boldsymbol{A}^* = \{a^*_{i,j}\}$ is the solution to Problem 11.

4.2 Solution

We present a polynomial-time algorithm for the exact solution to Problem 11, by breaking it down into structured subproblems and solving them recursively with dynamic programming.

We define the subproblem as follows: given relaxed segment trees $\mathcal{T}_1$ and $\mathcal{T}_2$, we would like to find the maximum IoU weighted alignment of $\mathcal{T}_1$ and $\mathcal{T}_2$, where the roots of $\mathcal{T}_1$ and $\mathcal{T}_2$ are aligned. Without loss of generality, we assume that the root nodes of $\mathcal{T}_1$ and $\mathcal{T}_2$ are both indexed by 1. Formally,

Problem 13 (maximum IoU weighted alignment, with root nodes aligned)**.**

$$f_{\mathcal{T}_1,\mathcal{T}_2} = \max_{\boldsymbol{A}} \sum_{i=1}^{|N_{\mathcal{T}_1}|} \sum_{j=1}^{|N_{\mathcal{T}_2}|} a_{i,j} \operatorname{IoU}\left(I_{\boldsymbol{n}_{1,i}}, I_{\boldsymbol{n}_{2,j}}\right)$$

$$\text{s.t. } a_{1,1} = 1;$$
$$\sum_j a_{i,j} \leq 1 (\forall i \in [|N_{\mathcal{T}_1}|]),$$
$$\sum_i a_{i,j} \leq 1 (\forall j \in [|N_{\mathcal{T}_2}|]),$$
$$a_{i,j} + a_{k,\ell} \leq 1$$
if $\boldsymbol{n}_{1,i} \leftrightarrow \boldsymbol{n}_{2,j}$ and$\boldsymbol{n}_{1,k} \leftrightarrow \boldsymbol{n}_{2,\ell}$ are conflicted,

where $\boldsymbol{A} \in \{0,1\}^{|\mathcal{T}_1| \times |\mathcal{T}_2|}$ is the alignment matrix.

While Problems 11 and 13 are not equivalent in principle, Problem 11 can be reduced to Problem 13 within $\mathcal{O}(1)$ time, by adding a dummy root node to each tree that associates with segments covering all the segments in both trees. We now present a polynomial-time solution to Problem 13.

Definition 14. Given a node $\boldsymbol{n}$ of a relaxed segment tree, $\mathbf{D} = (\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_k)$ is an **ordered disjoint descendant sequence** of $\boldsymbol{n}$ if

1. (ordered) for any $i, j \in [k]$ and $i < j$, $s_{\boldsymbol{n}_i} < s_{\boldsymbol{n}_j}$, where $s_{\boldsymbol{n}_i}$ and $s_{\boldsymbol{n}_j}$ are left endpoint of the associated intervals;
2. (disjoint) for any $i, j \in [k]$ and $i \neq j$, $I_{n_i} \cap I_{n_j} = \emptyset$;
3. (descendant) for any $i \in [k]$, $\boldsymbol{n}_i$ is a descendant of $\boldsymbol{n}$.

Corollary 14.1. *In an ordered disjoint descendant sequence* $\mathbf{D} = (\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_k)$ *of* $\boldsymbol{n}$, $e_{\boldsymbol{n}_i} \leq s_{\boldsymbol{n}_{i+1}}$ *for any* $i \in [k-1]$.

The solution to Problem 13 is given by the following recursion:

$$f_{\mathcal{T}_1,\mathcal{T}_2} = \operatorname{IoU}\left(I_{\boldsymbol{r}_{\mathcal{T}_1}}, I_{\boldsymbol{r}_{\mathcal{T}_2}}\right) + \max_{|\mathbf{D}_1|=|\mathbf{D}_2|} \sum_{i=1}^{|\mathbf{D}_1|} f_{\mathcal{T}(\boldsymbol{d}_{1,i}),\mathcal{T}(\boldsymbol{d}_{2,i})}, \tag{3}$$

where $\boldsymbol{r}_{\mathcal{T}_1}$ and $\boldsymbol{r}_{\mathcal{T}_2}$ denote the root nodes of $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively; $|\cdot|$ denotes the length of a sequence; $\mathbf{D}_1 = (\boldsymbol{d}_{1,1}, \boldsymbol{d}_{1,2}, \ldots, \boldsymbol{d}_{1,|\mathbf{D}_1|})$ and $\mathbf{D}_2 = (\boldsymbol{d}_{2,1}, \boldsymbol{d}_{2,2}, \ldots, \boldsymbol{d}_{2,|\mathbf{D}_2|})$ are same-length ordered disjoint descendant sequences of $\boldsymbol{r}_{\mathcal{T}_1}$ and $\boldsymbol{r}_{\mathcal{T}_2}$ respectively. Equation (3) can be computed within polynomial time, by solving a knapsack-style problem with dynamic programming. Specifically, let

$$g[\mathcal{T}_1, \mathcal{T}_2, e_1, e_2] = \max_{|\mathbf{D}_1^{e_1}|=|\mathbf{D}_2^{e_2}|} \sum_{j=1}^{|\mathbf{D}_1^{e_1}|} f_{\mathcal{T}(\boldsymbol{d}_{1,j}^{e_1}),\mathcal{T}(\boldsymbol{d}_{2,j}^{e_2})},$$

Algorithm 1 Polynomial time solution to Equation (3)

Input: $\mathcal{T}_1, \mathcal{T}_2$
  $g[\mathcal{T}_1, \mathcal{T}_2, x, y] \leftarrow 0, \forall x, y$
  $g'[\mathcal{T}_1, \mathcal{T}_2, x, y] := \max_{x'<x, y'<y} g[\mathcal{T}_1, \mathcal{T}_2, x', y']$
  $\boldsymbol{d}_1 \leftarrow$ the sequence of all descendants of $\boldsymbol{r}_{\mathcal{T}_1}$, sorted in increasing order of right endpoint
  $\boldsymbol{d}_2 \leftarrow$ the sequence of all descendants of $\boldsymbol{r}_{\mathcal{T}_2}$, sorted in increasing order of right endpoint
  for $i \leftarrow 1 \ldots, |\boldsymbol{d}_1|$ **do**
    for $j \leftarrow 1 \ldots, |\boldsymbol{d}_2|$ **do**
      $g[\mathcal{T}_1, \mathcal{T}_2, e_{\boldsymbol{d}_{1,i}}, e_{\boldsymbol{d}_{2,j}}] \leftarrow \max(g[\mathcal{T}_1, \mathcal{T}_2, e_{\boldsymbol{d}_{1,i}}, e_{\boldsymbol{d}_{2,j}}], f_{\mathcal{T}(\boldsymbol{d}_{1,i}), \mathcal{T}(\boldsymbol{d}_{2,j})} + g'[\mathcal{T}_1, \mathcal{T}_2, s_{\boldsymbol{d}_{1,i}}, s_{\boldsymbol{d}_{2,j}}])$
      update g' accordingly within $\mathcal{O}(1)$ time
    end for
  end for
Output: Equation (3) $= g[\mathcal{T}_1, \mathcal{T}_2, e_{\boldsymbol{r}_{\mathcal{T}_1}}, e_{\boldsymbol{r}_{\mathcal{T}_2}}]$

where e_1 and e_2 are arbitrary scalars denoting the constraints of endpoints; $\mathbf{D}_1^{e_1} = (\boldsymbol{d}_{1,1}^{e_1}, \ldots, \boldsymbol{d}_{1,|\mathbf{D}_1^{e_1}|}^{e_1})$ is an ordered disjoint descendant sequence of $\boldsymbol{r}_{\mathcal{T}_1}$, where for any $j \in [|\mathbf{D}_1^{e_1}|]$, the right endpoint of the corresponding node $e_{\boldsymbol{d}_{1,j}^{e_1}} \leq e_1$; similarly, $\mathbf{D}_2^{e_2} = (\boldsymbol{d}_{2,1}^{e_2}, \boldsymbol{d}_{2,2}^{e_2}, \ldots, \boldsymbol{d}_{2,|\mathbf{D}_2^{e_2}|}^{e_2})$ is a disjoint descendant sequence of $\boldsymbol{r}_{\mathcal{T}_2}$ of which the right endpoint of each node does not exceed e_2. Algorithm 1 computes g and Equation (3) within polynomial time, and therefore leads to a polynomial-time solution to Problem 13.

Complexity analysis. Suppose $|\mathcal{T}_1| = n$ and $|\mathcal{T}_2| = m$. To compute $f_{\mathcal{T}_1, \mathcal{T}_2}$, all we need to compute is $g[\mathcal{T}_1', \mathcal{T}_2', e_1', e_2']$ for all $\mathcal{T}_1', \mathcal{T}_2'$, e_1' and e_2'. Here, $\mathcal{T}_1'$ and $\mathcal{T}_2'$ enumerate over all subtrees of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, and e_1' and e_2' enumerate over the endpoints of all nodes in both trees, respectively. The update process requires $\mathcal{O}(1)$ time for each $\mathcal{T}_1, \mathcal{T}_2, e_1, e_2$. The edge cases, i.e., g values of terminal nodes, can be directly computed in $\mathcal{O}(1)$ time, and therefore, the overall time complexity to solve Problem 13 is $\mathcal{O}(n^2 m^2)$.

5 Experiments

We present two example applications of STRUCT-IOU: speech constituency parsing evaluation (§5.1) and text constituency parsing evaluation (§5.2), where the former is our main focus. In each part, we show the connection between STRUCT-IOU and existing metrics in appropriate settings and present the unique features of STRUCT-IOU.

5.1 Speech Constituency Parsing Evaluation

5.1.1 Datasets and Setups

We use the NXT-Switchboard (NXT-SWBD; Calhoun et al., 2010) dataset to train and evaluate models, where the parser can access the forced alignment word boundaries in both training and testing stages. We train an off-the-shelf supervised constituency parsing model for speech transcriptions (Jamshid Lou and Johnson, 2020) on the training set of NXT-SWBD, do early-stopping using PARSEVAL F_1 on the development set, and perform all the analysis below on the development set. The model achieves $F_1 = 85.4$ and STRUCT-IOU (averaged across sentences)[3]$= 0.954$ on the standard development set.

5.1.2 Comparison to the PARSEVAL F_1 score

Since the forced alignment word boundaries are accessible by the models, the PARSEVAL F_1 metric can be directly calculated between the predicted speech constituency parse tree and the ground truth. We compare the values of STRUCT-IOU and PARSEVAL (Sekine and Collins (1997) implementation with default parameters) in the settings with forced-alignment word segmentation (Figure 3), and find a strong correlation between the two metrics.

5.1.3 Analysis: STRUCT-IOU with Perturbed Word Boundaries

In textless speech parsing (Lai et al., 2023; Tseng et al., 2023), the word boundaries are unknown, and the boundaries predicted by the parser are usually

[3]Unless otherwise specified, all STRUCT-IOU scores reported in the paper are computed by averaging across STRUCT-IOU scores of individual sentences. We compare and discuss sentence-level and corpus-level STRUCT-IOU in §5.1.4

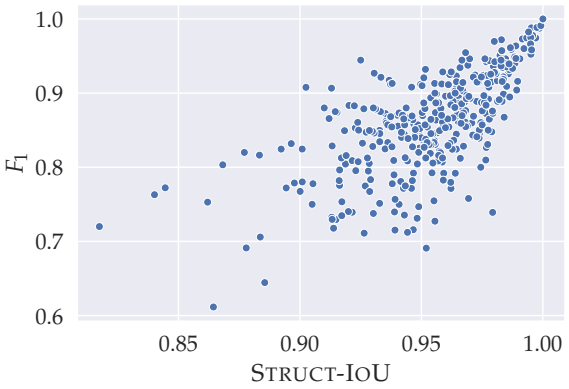
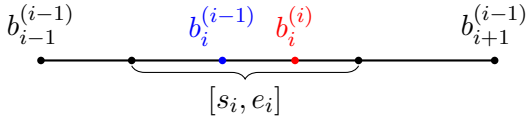
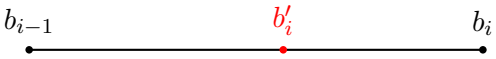


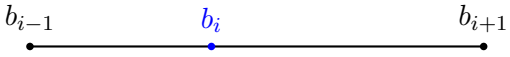
Figure 3: STRUCT-IOU vs. PARSEVAL F_1 on NXT-SWBD (Spearman's correlation $\rho = 0.689$, p-value=1.79×10^{-54}). Each dot represents the results of the base model (F_1=85.4 on the full development set) on 10 random examples from the development set.



(a) Noise-δ, where $s_i = b_i^{(i-1)} - \delta \cdot \left(b_i^{(i-1)} - b_{i-1}^{(i-1)}\right)$ and $e_i = b_i^{(i-1)} + \delta \cdot \left(b_{i+1}^{(i-1)} - b_i^{(i-1)}\right)$ denote the most left and right possible position of $b_i^{(i)}$.



(b) Insert-δ, with $r_i < \delta$; otherwise b'_i will not be inserted.



(c) Delete-δ, with $r_i < \delta$; otherwise b_i will not be deleted.

Figure 4: Examples of three types of perturbation: when applicable, the added boundaries are shown in red and the deleted boundaries are shown in blue. Best viewed in color.

imperfect. As a controlled simulation to such settings, we perturb the forced alignment word boundaries of the predicted parse tree (Figure 4), and calculate the STRUCT-IOU score between the perturbed parse tree and the ground truth over the original forced alignment word boundaries. Specifically, we suppose the word boundaries of a sentence with n words are $\mathcal{B} = b_0, b_1, \ldots, b_n$,[4] and consider the following types of perturbation with a hyperparameter $\delta \in [0, 1]$ controlling the perturbation level:

- **Noise**-δ. We start with $\mathcal{B}^{(0)} = \mathcal{B}$, and update the boundaries iteratively as follows. For each $i \in [n-1]$, we randomly draw a number r_i from the uniform distribution $U(-\delta, \delta)$, and let $b_i^{(i)} = b_i^{(i-1)} + |r_i| * \left(b_{i+\text{sgn}(r_i)}^{(i-1)} - b_i^{(i-1)}\right)$, where $\text{sgn}(\cdot) : \mathbb{R} \to \{1, -1\}$ denotes the sign function

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0; \\ -1 & \text{if } x < 0. \end{cases}$$

  For all $j \neq i$ and $j \in [n]$, we let $b_j^{(i)}$ remain the same as $b_j^{(i-1)}$. Finally, we take $\mathcal{B}^{(n-1)}$ as the perturbed word boundaries for the predicted tree.
- **Insert**-δ. We randomly draw a number r_i from the uniform distribution for each boundary index $i \in [n]$. If $r_i < \delta$, we insert a word boundary at the position b'_i, randomly drawn from the uniform distribution $U(b_{i-1}, b_i)$, breaking the i^{th} spoken word into two (i.e., $[b_{i-1}, b'_i]$ and $[b'_i, b_i]$).
- **Delete**-δ. Similarly to the insertion-based perturbation, we randomly draw a number r_i from the uniform distribution $U(0, 1)$ for each boundary index $i \in [n-1]$, and delete the boundary b_i if $r_i < \delta$. Since such boundary deletion may break the predicted tree structure, we use the base model (§5.1.1) to re-predict the parse tree with the new word boundaries, where words concatenated by space are taken as the textual input (Jamshid Lou and Johnson, 2020).

A larger δ means a higher level of perturbation is applied, and we therefore expect a lower STRUCT-IOU score; $\delta = 0$ means no perturbation is applied, and the STRUCT-IOU score is the same as that for the predicted parse trees with forced alignment word boundaries.

For each $\delta \in \{0.1, 0.2, \ldots, 1.0\}$, starting from the base model (for deletion-based perturbation) or its predicted parse trees (for noise and insertion-based perturbation), we run the perturbation 5 times and report both the mean and the standard deviation of the STRUCT-IOU result after perturbation.

Results and discussion. We present how the STRUCT-IOU value changes with respect to δ for different types of perturbation (Figure 5). The standard deviation is nearly invisible in the figure, showing that our metric is stable under a specific setting. For all three types of perturbation, as desired, a larger δ leads to a lower STRUCT-IOU score. Among the perturbation types, STRUCT-IOU is the most sensitive to deletion, and the least sensitive to noise-based perturbation. Although the results are not comparable across perturbation

[4] We assume no silence between spoken words; if any inter-word silence exists, we remove it.

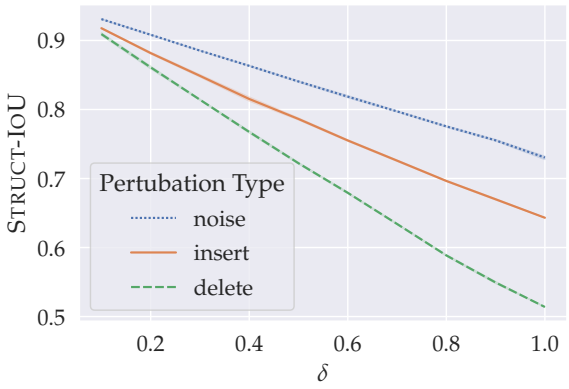


Figure 5: STRUCT-IOU scores with respect to δ for different types of perturbations.

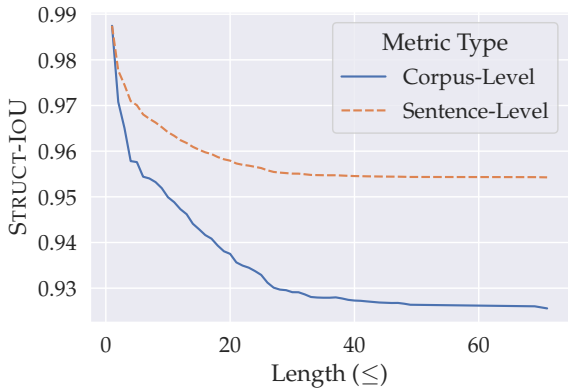


Figure 6: Corpus-level and sentence-level STRUCT-IOU scores of the predicted parse trees of the base model ($F_1 = 85.4$ on the development set), evaluated on development examples with less than or equal to a certain number of spoken words.

types in the most rigorous sense, this reflects the fact that STRUCT-IOU, to some extent, is more sensitive to structural change of the trees than simple word boundary changes.

Although both word boundary insertion and deletion change the predicted tree structures, the former has less impact on the STRUCT-IOU scores. This also aligns with our expectation: boundary insertion only splits some of the spoken words into two and keeps the longer constituents; however, deletion may change significantly the tree structure, especially when it happens at the boundary of two long constituents.

5.1.4 Corpus-Level vs. Sentence-Level Metric

Note that 39.7% utterances in the NXT-SWBD development set contain only one spoken word, and the STRUCT-IOU score of such sentences is always high—the metric degenerates to the IOU score between two intervals. Averaging the STRUCT-IOU scores across all sentence pairs in the dataset may therefore overly emphasize these short utterances. To address this, we introduce the corpus-level STRUCT-IOU score as an alternative, where Definition 12 is modified as follows:

Definition 15. The corpus-level STRUCT-IOU between two sets of parsed trees $\mathcal{D}_1 = \{\mathcal{T}_{1,k}\}$ and $\mathcal{D}_2 = \{\mathcal{T}_{2,k}\}$ is given by

$$\overline{\text{IOU}}(\mathcal{D}_1, \mathcal{D}_2) = \frac{\sum_{k=1}^{|\mathcal{D}_1|} (|\mathcal{T}_{1,k}| + |\mathcal{T}_{2,k}|)\, \overline{\text{IOU}}(\mathcal{T}_{1,k}, \mathcal{T}_{2,k})}{\sum_{k'=1}^{|\mathcal{D}_1|} |\mathcal{T}_{1,k'}| + |\mathcal{T}_{2,k'}|},$$

where $|\mathcal{D}_1| = |\mathcal{D}_2|$, and a pair of $\mathcal{T}_{1,k}$ and $\mathcal{T}_{2,k}$ denotes the parse trees of the k^{th} sentence in the corpus respectively.

We compare the corpus-level and sentence-level STRUCT-IOU scores (Figure 6). As desired, the corpus-level STRUCT-IOU score has lower absolute values than the sentence-level one, and the difference is more significant when longer sentences are considered. A similar phenomenon has been found in text constituency parsing (Kim et al., 2019) as well, where corpus-level PARSEVAL F_1 scores are lower than sentence-level ones.

5.2 Evaluation of Text Constituency Parsing

We extend our experiment to the evaluation of text constituency parsing. In this part, we suppose every written word corresponds to a segment of the same length—analogously, this can be considered as speech parsing with evenly distributed word boundaries, for both predicted and ground-truth trees.

5.2.1 Correlation with PARSEVAL F_1 Scores on the Penn Treebank

We use the Penn Treebank (PTB; Marcus et al., 1993) dataset to train and evaluate Benepar (Kitaev and Klein, 2018), a state-of-the-art text constituency parsing model, doing early-stopping using labeled PARSEVAL F_1 on the development set. The base model achieves PARSEVAL $F_1 = 94.4$ and STRUCT-IOU (averaged across sentences) $= 0.962$ on the standard development set.

We compare the STRUCT-IOU scores with the PARSEVAL F_1 scores on the development set (Figure 7). As in the speech parsing experiment, we find a strong correlation between the two metrics, showing that STRUCT-IOU is consistent with the

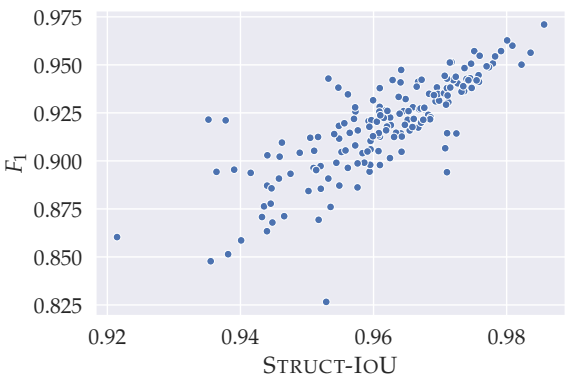


Figure 7: Comparison of STRUCT-IOU and PARSEVAL F_1 (Spearman's rank correlation $\rho = 0.821$, p-value=8.16×10^{-43}). Each dot represents the results of the base model on 10 random examples from the PTB development set.

existing metric in the text parsing domain.

5.2.2 STRUCT-IOU vs. PARSEVAL F_1 on Syntactically Ambiguous Sentences

We consider a special setting of parsing syntactically ambiguous sentences, where the syntactically plausible parse tree of a sentence may not be unique (see examples in Figure 8). We simplify the case shown in Figure 8 and generate synthetic sentences with syntactic ambiguity with the template `N (P N){n}`, where `P` denotes a preposition and `N` denotes a noun, and n determines how many times the `P N` pattern is repeated. For `N (P N){2}`, the two potential parse trees are shown in Figure 9.

We compare the PARSEVAL and STRUCT-IOU in the following scenarios, choosing a random syntactically plausible parse tree as the ground truth:

- **Ground truth vs. random parse trees**, where the random parse trees are constructed by recursively combining random consecutive words (or word groups) into a binary tree. We construct 100 random parse trees and report the average.
- **Ground truth vs. syntactically plausible parse trees**, where we report the lowest possible score between the ground truth and other syntactically plausible trees.

As shown in Table 1, the lowest possible PARSEVAL F_1 score between the ground truth and another syntactically plausible tree is significantly lower than the score achieved by meaningless random trees; however, STRUCT-IOU consistently assigns higher scores to the syntactically plausible parses, showing more tolerance to syntactic ambiguity.

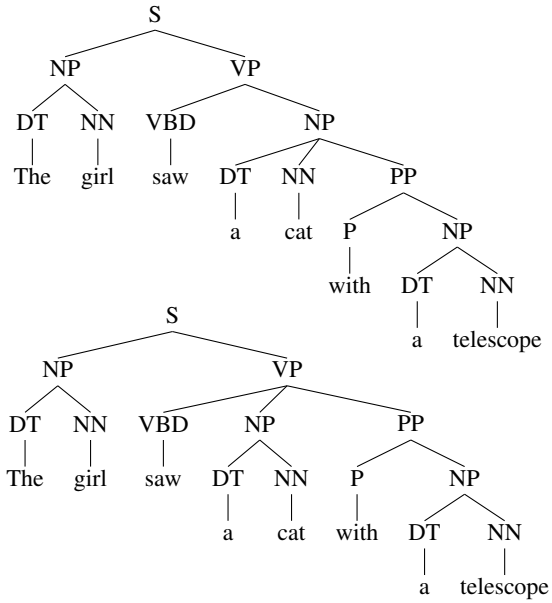


Figure 8: An example syntactically ambiguous sentence: *The girl saw a cat with a telescope*. Both parses are syntactically valid, but the first one implies that *a cat* was holding the telescope, whereas the second implies *the girl* was using the telescope.

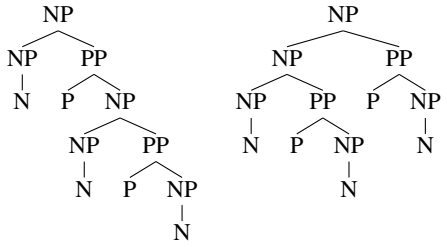


Figure 9: Two syntactically plausible parses of the `N (P N){2}`, where NP denotes a noun phrase, and PP denotes a prepositional phrase.

6 Conclusion and Discussion

In this paper, we present STRUCT-IOU, the first metric that computes the similarity between two parse trees over continuous spoken word boundaries. STRUCT-IOU enables the evaluation of textless speech parsing (Lai et al., 2023; Tseng et al., 2023), where no text or speech recognizer is used or available to parse spoken utterances.

In the canonical settings of text and speech parsing, STRUCT-IOU serves as a complement to the existing evaluation metrics (Black et al., 1991; Roark et al., 2006). In particular, STRUCT-IOU shows a higher tolerance to potential syntactic ambiguity under certain scenarios, providing an alternative interpretation of the parsing quality.

Metric	Ground-Truth vs. Random, Average
PARSEVAL F_1	27.3
STRUCT-IOU	61.9
	Ground-Truth vs. Plausible, Lowest
PARSEVAL F_1	12.5
STRUCT-IOU	63.6

Table 1: Average PARSEVAL F_1 and STRUCT-IOU scores between the ground truth and a random binary tree, and the lowest possible scores between the ground truth and another syntactically plausible tree. Experiments are done on the string "`N (P N){8}`". For simplicity, we report the unlabeled scores, where all nonterminals are treated as having the same label.

7 Limitations

STRUCT-IOU is designed to evaluate constituency parse trees over continuous spoken word boundaries, and is not directly applicable to evaluate other types of parses, such as dependency parse trees; however, it may be extended to evaluate other types of parse trees by modifying the alignment constraints. We leave the extension of STRUCT-IOU to other types of parses as future work. We do not foresee any risk of STRUCT-IOU being used in harmful ways.

References

Jon L. Bentley. 1977. Solutions to Klee's rectangle problems. *Technical Report, Carnegie Mellon University*.

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Sasha Calhoun, Jean Carletta, Jason M Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. 2010. The nxt-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language resources and evaluation*, 44:387–419.

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas. Association for Computational Linguistics.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2015. Neural CRF parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 302–312, Beijing, China. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

John J. Godfrey and Edward Holliman. 1993. Switchboard-1 release 2.

Paria Jamshid Lou and Mark Johnson. 2020. Improving disfluency detection by self-training a self-attentive model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3754–3763, Online. Association for Computational Linguistics.

Jeremy G. Kahn, Matthew Lease, Eugene Charniak, Mark Johnson, and Mari Ostendorf. 2005. Effective use of prosody in parsing conversational speech. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 233–240, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Jeremy G Kahn and Mari Ostendorf. 2012. Joint reranking of parsing and word recognition with automatic segmentation. *Computer Speech & Language*, 26(1):1–19.

Jeremy G. Kahn, Mari Ostendorf, and Ciprian Chelba. 2004. Parsing conversational speech using enhanced segmentation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 125–128, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Cheng-I Jeff Lai, Freda Shi, Puyuan Peng, Yoon Kim, Kevin Gimpel, Shiyu Chang, Yung-Sung Chuang, Saurabhchand Bhati, David Cox, David Harwath, et al. 2023. Audio-visual neural syntax acquisition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Matthew Lease and Mark Johnson. 2006. Early deletion of fillers in processing conversational speech. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 73–76, New York City, USA. Association for Computational Linguistics.

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.

Alex Marin and Mari Ostendorf. 2014. Domain adaptation for parsing in automatic speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6379–6383. IEEE.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Annual Conference of the International Speech Communication Association (Interspeech)*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA. Association for Computational Linguistics.

Brian Roark, Mary Harper, Eugene Charniak, Bonnie Dorr, Mark Johnson, Jeremy Kahn, Yang Liu, Mari Ostendorf, John Hale, Anna Krasnyanskaya, Matthew Lease, Izhak Shafran, Matthew Snover, Robin Stewart, and Lisa Yung. 2006. SParseval: Evaluation metrics for parsing speech. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Satoshi Sekine and Michael Collins. 1997. Evalb bracket scoring program.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Ke Tran and Mari Ostendorf. 2021. Assessing the use of prosody in constituency parsing of imperfect transcripts. In *Annual Conference of the International Speech Communication Association (Interspeech)*.

Trang Tran, Shubham Toshniwal, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Mari Ostendorf. 2018. Parsing speech: a neural approach to integrating lexical and acoustic-prosodic information. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 69–81, New Orleans, Louisiana. Association for Computational Linguistics.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Joint evaluation of morphological segmentation and syntactic parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 6–10, Jeju Island, Korea. Association for Computational Linguistics.

Yuan Tseng, Cheng-I Jeff Lai, and Hung-yi Lee. 2023. Cascading and direct approaches to unsupervised constituency parsing on spoken sentences. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Appendix

A Proof of Corollaries and Propositions

We present the proof of the corollaries mentioned in the main content as follows.

Corollary 5.1 For nodes $\boldsymbol{p}, \boldsymbol{n}$, if $\boldsymbol{p} \in C_{\boldsymbol{n}}$, then $I_{\boldsymbol{p}} \subseteq I_{\boldsymbol{n}}$.

Proof. According to the definition of open intervals and Definition 5 (3),

$$\begin{aligned} & a_{\boldsymbol{n}} \leq a_{\boldsymbol{p}} < b_{\boldsymbol{p}} \leq b_{\boldsymbol{n}} \\ \Rightarrow & I_{\boldsymbol{p}} = (a_{\boldsymbol{p}}, b_{\boldsymbol{p}}) \subseteq (a_{\boldsymbol{n}}, b_{\boldsymbol{n}}) = I_{\boldsymbol{n}}. \end{aligned}$$

□

Corollary 6.1 If node $\boldsymbol{p}$ is an ancestor of node $\boldsymbol{q}$, then $I_{\boldsymbol{p}} \supseteq I_{\boldsymbol{q}}$.

Proof. According to Definition 6, there exists a sequence of nodes $\boldsymbol{n}_0, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_k (k \geq 1)$ such that (1) $\boldsymbol{n}_0 = \boldsymbol{p}$, (2) $\boldsymbol{n}_k = \boldsymbol{q}$ and (3) for any $i \in [k]$, $\boldsymbol{n}_i \in \mathbf{C}_{\boldsymbol{n}_{i-1}}$.

Corollary 5.1 implies that for any $i \in [k]$, $I_{\boldsymbol{n}_{i-1}} \supseteq I_{\boldsymbol{n}_i} \Rightarrow I_{\boldsymbol{n}_0} \supseteq I_{\boldsymbol{n}_k} \Rightarrow I_{\boldsymbol{p}} \supseteq I_{\boldsymbol{q}}$. □

Corollary 8.1 A relaxed segment tree can be uniquely characterized by its root node.

Proof. ($\Rightarrow$) Definition 8 implies that each relaxed segment tree has one root node.

($\Leftarrow$) Given a specific node $\boldsymbol{n}$, we have the unique set $\mathcal{N} = \{\boldsymbol{n}\} \cup \{\boldsymbol{n}' : \boldsymbol{n}' \text{ is a descendant of } \boldsymbol{n}\}$, and therefore extract the set of all nodes in the relaxed segment tree rooted at $\boldsymbol{n}$. □

Proposition 9 For a relaxed segment tree $\mathcal{T}$ and $\boldsymbol{p}, \boldsymbol{q} \in N_{\mathcal{T}}$, $\boldsymbol{p}$ is neither an ancestor nor a descendant of $\boldsymbol{q} \Leftrightarrow I_{\boldsymbol{p}} \cap I_{\boldsymbol{q}} = \emptyset$.

Proof. ($\Rightarrow$) Let $\boldsymbol{z}$ denote the least common ancestor of $\boldsymbol{p}$ and $\boldsymbol{q}$. There exists $\boldsymbol{p}', \boldsymbol{q}' \in C_{\boldsymbol{z}} (\boldsymbol{p}' \neq \boldsymbol{q}')$ such that $I_{\boldsymbol{p}'} \supseteq I_{\boldsymbol{p}}$ and $I_{\boldsymbol{q}'} \supseteq I_{\boldsymbol{q}}$; therefore

$$I_p \cap I_q \subseteq I_{\boldsymbol{p}'} \cap I_{\boldsymbol{q}'} \overset{\text{Definition 5 (2)}}{=} \emptyset \Rightarrow I_{\boldsymbol{p}} \cap I_{\boldsymbol{q}} = \emptyset.$$

($\Leftarrow$) If $I_{\boldsymbol{p}} \cap I_{\boldsymbol{q}} = \emptyset$, according to Definition 5 (3) and Definition 6, $\boldsymbol{p}$ is not an ancestor of $\boldsymbol{q}$ and vice versa. □

Corollary 14.1 Given an ordered disjoint descendant sequence $\mathbf{S} = (\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_k)$ of $\boldsymbol{n}$, for any $i \in [k-1]$, $b_{\boldsymbol{n}_i} \leq a_{\boldsymbol{n}_{i+1}}$.

Proof. If there exists $i \in [k-1]$ such that $b_{\boldsymbol{n}_i} > a_{\boldsymbol{n}_{i+1}}$, then

$$\begin{aligned} & I_{\boldsymbol{n}_i} \cap I_{\boldsymbol{n}_{i+1}} \\ =& (a_{\boldsymbol{n}_i}, b_{\boldsymbol{n}_i}) \cap (a_{\boldsymbol{n}_{i+1}}, b_{\boldsymbol{n}_{i+1}}) \\ =& \left\{x : \max(a_{\boldsymbol{n}_i}, a_{\boldsymbol{n}_{i+1}}) < x < \min(b_{\boldsymbol{n}_i}, b_{\boldsymbol{n}_{i+1}})\right\} \\ =& \left\{x : a_{\boldsymbol{n}_{i+1}} < x < \min(b_{\boldsymbol{n}_i}, b_{\boldsymbol{n}_{i+1}})\right\} \\ & \textit{(Definition 14 (1)).} \end{aligned}$$

Since $b_{\boldsymbol{n}_{i+1}} > a_{\boldsymbol{n}_{i+1}}$ (definition of open intervals),

$$a_{\boldsymbol{n}_{i+1}} < \min(b_{\boldsymbol{n}_i}, b_{\boldsymbol{n}_{i+1}}) \Rightarrow I_{\boldsymbol{n}_i} \cap I_{\boldsymbol{n}_{i+1}} \neq \emptyset.$$

This conflicts with Definition 14 (2). □

B Checklist Details

We discuss the additional checklist details in the following content.

B2: Licenses of Scientific Artifacts. EVALB (Sekine and Collins, 1997), an open-source implementation of PARSEVAL (Black et al., 1991), is free and unencumbered software released into the public domain. SPARSEVAL (Roark et al., 2006) is licensed under the Apache License, Version 2.0.