CPTAM: Constituency Parse Tree Aggregation Method

Adithya Kulkarni* Nasim Sabetpour* Alexey Markin Oliver Eulenstein Qi Li
{aditkulk, nasim, amarkin, oeulenst, qli}@iastate.edu
Iowa State University, Ames, Iowa, USA

Abstract

Diverse Natural Language Processing tasks employ constituency parsing to understand the syntactic structure of a sentence according to a phrase structure grammar. Many state-of-the-art constituency parsers are proposed, but they may provide different results for the same sentences, especially for corpora outside their training domains. This paper adopts the truth discovery idea to aggregate constituency parse trees from different parsers by estimating their reliability in the absence of ground truth. Our goal is to consistently obtain high-quality aggregated constituency parse trees. We formulate the constituency parse tree aggregation problem in two steps, structure aggregation and constituent label aggregation. Specifically, we propose the first truth discovery solution for tree structures by minimizing the weighted sum of Robinson-Foulds (RF) distances, a classic symmetric distance metric between two trees. Extensive experiments are conducted on benchmark datasets in different languages and domains. The experimental results show that our method, CPTAM, outperforms the state-of-the-art aggregation baselines. We also demonstrate that the weights estimated by CPTAM can adequately evaluate constituency parsers in the absence of ground truth.

Keywords- Constituency parse tree, Truth discovery, Optimization

1 Introduction

The constituency parse trees (CPTs) display the syntactic structure of a sentence using context-free grammar. CPTs divide the input sentence into phrase structures that belong to a specific grammar category. The available state-of-the-art constituency parsers use different parsing techniques. They are leveraged in various NLP applications like Question Answering, Information Extraction, and word-processing systems. However, due to multiple limitations, the state-of-the-art constituency parsers may make errors, and different constituency parsers may give different results for the same sentence. The conflicts among parsers can confuse users on the parser to use for the downstream tasks, as the performance of different parsers can vary significantly on different domains and languages. No parser can consistently achieve the best results on all datasets, and it is costly and impractical for users to obtain ground truth parsing results. Table 1 shows the percentage of agreement among the structure of the parsers' outputs on six benchmark datasets, including Penn Treebank-3 [46], OntoNotes (English and Chinese) [32], Genia [31], French Treebank [1], and TIGER Corpus [6]. We execute four parsers including Berkeley [19], CoreNLP [27], AllenNLP [14], and Hanlp [17], for the English datasets, and three parsers, namely Berkeley, CoreNLP, and Hanlp, for the non-English datasets. On the Penn Treebank-3 dataset, it can be observed that all the parsers agree only on 1.32% of the sentences. A similar observation can be made for other datasets.

To overcome these challenges, we aim to construct a CPT that performs consistently well to represent the constituency grammar of the sentence in the absence of ground truth. Intuitively, such CPTs can be constructed by aggregating the parsing results from the state-of-the-art parsers to keep the common structure among the parsers' output and resolve their conflicts. Therefore, we propose to aggregate CPTs through the truth discovery idea.

Truth discovery has been proposed to conduct weighted aggregation for various applications, where the weights reflect the source reliabilities and are inferred from the data itself without the knowledge of the ground truth [23]. Truth discovery algorithms [22, 36, 37, 48] witness successful applications on the aggregation of categorical and numerical data types. However, the aggregation of the tree data type has never been investigated in this domain.

There are tree aggregation methods proposed in the phylogenetic domain [2, 28, 43, 15] and ensemble methods for parsers [44]. The issue with studies in the phylogenetic domain is that many assumptions are not applicable for CPTs, and none of them consider the constituent label aggregation. The ensemble methods use ground truth to evaluate the quality of the weak learners, whereas, for our task, the aggregation needs to be conducted in the absence of ground truth.

In this paper, we adopt the truth discovery framework to aggregate CPTs from different input parsers by estimating the parsers' reliability without ground truth. We formulate the constituency parse tree aggregation problem in two steps, structure aggregation and constituent label aggregation. In the

[*] The first two authors contributed equally to this work.

	Penn Treebank-3	OntoNotes (English)	OntoNotes (Chinese)	French Treebank	TIGER Corpus	Genia
Agreement of all parsers	1.32	6.65	23.45	0.91	3.23	0.21
Agreement of two parsers	45.33	55.14	31.73	7.69	2.94	16.25
No agreement	53.35	38.21	44.82	91.40	93.83	83.54

Table 1: Percentage of the sentences that different parsers agree on the tree structure

structure aggregation step, the key challenges are measuring the distance between trees and constructing the aggregated tree that can minimize that distance. We adopt the Robinson-Foulds (RF) distance, a symmetric difference metric for trees [35], to measure the distance between the aggregated tree and input CPTs. In practice, we propose an efficient algorithm that can construct the optimal aggregated tree in near-linear time and provide theoretical proofs. We adopt the same truth discovery framework in the constituent label aggregation step.

Extensive empirical studies demonstrate that the proposed Constituency Parse Tree Aggregation Model (CPTAM) can consistently obtain high-quality aggregated CPTs across different languages and domains. Specifically, we apply the most widely used constituency parsers as the input parsers on six corpora from English, Chinese, French and German languages, and from general domains and bio-medical domains. Our experimental results validate that there is no single parser that can achieve the best results across the corpora. CPTAM can consistently obtain high-quality results and significantly outperforms the aggregation baselines. We further examine the estimated weights for the parsers and illustrate that the weight estimation can correctly reflect the quality of each parser in the absence of ground truth.

In summary, our main contributions are:

- We identify the pitfalls and challenges in data with tree structures for the task of truth discovery.
- We adopt Robinson-Foulds (RF) distance to measure the differences among data with tree structures.
- We construct the best aggregation trees by solving an optimization problem and derive the theoretical proofs for the correctness and the efficiency of the algorithm.
- We test the proposed algorithm on real-world datasets, and the results clearly demonstrate the advantages of the approach in finding the accurate tree structures from the multi-sourced input.

2 Related Works

We summarize the related works in three categories as below.

2.1 Truth Discovery Truth discovery aims to resolve the conflicts from multiple sources [23]. One line of work applies probabilistic methods to model the workers' labeling behavior in crowdsourcing platforms [12, 26, 21]. Another line of work formulates optimization frameworks that seek to minimize the weighted distance between the source and aggregated results and estimate the source reliability [22, 49]. Recent truth discovery methods consider different applications such as aggregation of sequential labels [37, 42, 29] and aggregation of time series data [47, 24, 41].

Most of the available truth discovery methods mainly focus on the numerical and categorical data [23], and none of them consider tree structure. Furthermore, the distance measurements introduced in previous works do not support the tree structure. However, the problem of how to aggregate information from trees into one representative tree has been of great importance for various applications [34].

2.2 Phylogenetic Tree Aggregation Problem The tree aggregation problem has been studied in the *phylogenetic* domain, where trees are branching diagrams showing the evolutionary relationships among biological species or other taxa [13]. The taxa can be described through different types of data (e.g., morphological or biomolecular). Since the inference of phylogenetic trees is an immensely complex problem, practitioners often perform many tree estimation runs with the same or different phylogenetic inference methods. The estimated trees are aggregated using *consensus tree* techniques [5, 8].

A variety of methods have been developed for phylogenetic tree aggregation [2, 8]. Some methods conduct aggregation through simple heuristics when the aggregated tree only contains branches with a certain percentage of agreement, such as the majority rule consensus [28], strict consensus [7], semi-strict consensus [15], and the greedy consensus [8]. Further, *supertree* and *median tree* approaches have been extensively explored to compute fully binary aggregated trees [4]. Such methods typically seek an output tree that minimizes the overall distance to the input trees. Since the mentioned methods are introduced in *phylogenetic* domain, they do not consider the characteristics of parse trees.

2.3 Ensemble Trees Tree ensemble methods such as Random Forest [33] or Boosted trees [10] are not suitable for our needs since these methods ensemble on the classification decisions instead of constructing an aggregation tree.

There are multiple ensemble models for the parsing of syntactic dependencies in the literature, aiming to construct aggregation trees [44, 20]. These parsing tree ensemble methods are commonly categorized into two groups. The first group aggregates the base parsers at training time [30, 3, 44]. The second group aggregates the independently trained models at the prediction time [38, 16, 20]. One of the common approaches in these ensemble methods is to find the maximum spanning tree (MST) for the directed weighted graph to obtain the optimal dependency structure. Unlike our proposed task, all these methods rely on the ground truth to estimate the parsers' reliability.

3 Preliminaries

This section briefly overviews the optimization-based problem in truth discovery that we adopt for CPT aggregation. The basic idea is that the inferred truth is likely to be correct if a reliable source provides it. Therefore, the goal is to minimize the overall distance of the aggregated truth to a reliable source [22]. Based on this principle, the optimization framework is defined as follows:

$$\min_{\mathcal{X}^*,\mathcal{W}} f(\mathcal{X}^*,\mathcal{W}) = \sum_{k=1}^{K} w_k \sum_{i=1}^{N} \sum_{m=1}^{M} d_m(v_{im}^*, v_{im}^k)$$
$$s.t.\ \delta(\mathcal{W}) = 1, \mathcal{W} \in \mathcal{S}, \tag{3.1}$$

where $\mathcal{X}^*$ and $\mathcal{W}$ correspond to the set of truths and the source weight, respectively, and w_k refers to the reliability degree of the k-th source. The function $d_m(\cdot,\cdot)$ measures the distance between the sources' observations v_{im}^k and the aggregated truths v_{im}^*. The regularization function $\delta(\mathcal{W})$ is defined to guarantee the weights are always non-zero and positive.

To optimize the objective function Eq. (3.1), the block coordinate descent algorithm is applied by iteratively updating the aggregated truths and source weights, conducting the following two steps.

Source Weight Update. To update the source weight in the model, the values for the truths are considered fixed, and the source weights are computed, which jointly minimizes the objective function as shown in Eq. (3.2).

$$\mathcal{W} \leftarrow \operatorname*{argmin}_{\mathcal{W}} f(\mathcal{X}^*,\mathcal{W})\ s.t.\ \delta(\mathcal{W}) = \sum_{k=1}^{K} exp(-w_k). \tag{3.2}$$

This function regularizes the value of w_k by constraining the sum of $exp(-w_k)$.

Truth Update. At this step, the weight of each source w_k is fixed, and the truth is updated for each entry to minimize the difference between the truth and the sources' observations, where sources are weighted by their reliability degrees.

$$v_{im}^{(*)} \leftarrow \operatorname*{argmin}_{v} \sum_{k=1}^{K} w_k \cdot d_m(v, v_{im}^k). \tag{3.3}$$

By deriving the truth using Eq. (3.3) for every instance, the collection of truths $\mathcal{X}^*$ which minimizes $f(\mathcal{X}^*,\mathcal{W})$ with fixed $\mathcal{W}$ is obtained.

Notation	Definition
n	number of sentences indexed by i
p	number of parsers indexed by k
S_i	the i-th sentence in the dataset
$\mathcal{W}$	set of input CPTs' weights
w_k^S	the weight of the k-th parser w.r.t. the clusters
w_k^l	the weight of the k-th parser w.r.t. the labels
T_{ik}	the k-th input CPT for the i-th sentence
C_i	set of all unique clusters from input trees for the i-th sentence
$T_i^{S^*}$	aggregated tree for the i-th sentence w.r.t. the tree structure
T_i^*	aggregated tree for the i-th sentence w.r.t. the labels
$\mathcal{L}_{Clu(T)}$	clusters' labels in tree T

Table 2: Summary of Notations

4 Constituency Parse Tree Aggregation Model (CPTAM)

In this section, we first formally define the problem. Then, we propose our solution in two steps. In the first step, we focus on tree structure aggregation to resolve the conflict between input trees and obtain the aggregated tree structure $T_i^{S^*}$ of the CPTs. In the second step, the corresponding POS tags and constituent labels are obtained through the label aggregation. It is worth mentioning that both tree structures and tree labels are essential for adequately parsing a sentence.

4.1 Problem Definition

We define the CPT aggregation problem using the notations summarized in Table 2. Suppose there is a corpus that consists of n sentences indexed by i ($i \in [1, n]$), and p different parsers indexed by k ($k \in [1, p]$) produce CPTs for each sentence in the corpus. We use T_{ik} to denote the k-th input CPT for the i-th sentence (S_i). Each input constituency parser has two weight parameters w_k^S and w_k^l to reflect the parser's reliability with respect to structure and constituent labels, respectively. We use different weight parameters for structure and constituent labels to account for the scenarios where a parser can successfully identify the phrase structure but assign incorrect labels. A higher weight implies that the parser is of higher reliability. The CPT aggregation problem seeks an aggregated tree for a sentence (T_i^*) given the input CPTs (T_{ik}), and estimates the qualities of parsers in the absence of ground truth.

4.2 Tree Structure Aggregation

We formulate our framework utilizing the truth discovery framework presented in Eq. (3.1). In the tree structure aggregation step, our goal is to minimize the overall weighted distance of the aggregated CPT ($T_i^{S^*}$) to the reliable input CPT (T_{ik}^S) considering the structure only. Various distance measurements can be plugged in the optimization function shown in Eq. (3.1). We adopt RF distance defined in Eq. (4.4).

Robinson-Foulds (RF) distance is a symmetric difference metric to calculate the distance between leaf-labeled trees [35] in terms of clusters, where a *cluster* refers to a maximal set of leaves with a common ancestor in a rooted tree (T) [40]. For any two trees T_1 and T_2 that share the same leaf set, the RF distance is defined in Eq. (4.4):

$$RF(T_1, T_2) = |Clu(T_1) \Delta Clu(T_2)|, \tag{4.4}$$

where the operation Δ computes the symmetric difference between two sets (i.e., $A\Delta B = (A\backslash B)\bigcup(B\backslash A)$), function $|\cdot|$ computes the cardinality of the set, and $Clu(T)$ refers to the cluster set of tree T. Different from Tree Edit Distance (TED) [39], which takes $O(y^3)$ time [11] to calculate, where y refers to the number of tokens in the sentence, RF distance can be calculated in $O(|Clu(T_1)| + |Clu(T_2)|)$ time [9].

Applying the truth discovery framework (Eq. (3.1)), we formulate the CPT aggregation problem with respect to the tree structure as shown in Eq. (4.5). Each parser has a weight parameter w_k^S to reflect the reliability of that parser in terms of the structure, and $\mathcal{W}^S = \{w_1^S, w_2^S, ..., w_p^S\}$ refers to the set of all parsers' weights in terms of the structure. The higher the weight, the more reliable the parser. The aggregated tree $\mathcal{T}^{S^*}$ is the one that can minimize the overall weighted RF distances.

$$\min_{\mathcal{T}^{S^*}, \mathcal{W}^S} f(\mathcal{T}^{S^*}, \mathcal{W}^S) = \sum_{k=1}^{p} w_k^S \sum_{i=1}^{n} RF(T_i^{S^*}, T_{ik}^S). \qquad (4.5)$$

We follow the block coordinate descent method introduced in Section 3. To update the weights of the input constituency parsers in the objective function Eq. (4.5), $T_i^{S^*}$ is fixed, and w_k^S is updated as follows:

$$w_k^S = -log(\frac{\sum_i RF(T_i^{S^*}, T_{ik}^S)}{max_k \sum_i RF(T_i^{S^*}, T_{ik}^S)}). \qquad (4.6)$$

This means that the weight of a parser is inversely proportional to the maximum sum of the distance between its input trees (we use T_{ik}^S to refer to the input CPT with respect to the structure) and the aggregated trees. Next, we update the aggregated parse tree for each sentence to minimize the difference between the aggregated parse tree and the input CPTs by treating the weights as fixed. The aggregated tree is updated following Eq. (3.3) as shown in Eq. (4.7):

$$T_i^{S^*} \leftarrow \underset{T_i^{S^*}}{\operatorname{argmin}} \sum_{k=1}^{p} w_k^S \sum_{i=1}^{n} RF(T_i^{S^*}, T_{ik}^S). \qquad (4.7)$$

We propose an optimal solution for Eq. (4.7).

4.2.1 The Optimal Solution

We present an *optimal* solution to obtain an aggregated tree by solving the optimization problem in Eq. (4.7). Our proposed approach constructs the aggregated tree by adding clusters with weighted support greater than or equal to 50%, where support refers to the aggregated weight of CPTs containing that cluster. To establish the solution, we first demonstrate some properties of an optimal aggregated tree.

LEMMA 4.1. *The cluster set ($Clu(T_i^{S^*})$) in Eq. (4.7) satisfies the constraint $Clu(T_i^{S^*}) \subseteq C_i$ ($C_i = \cup_{k=1}^p Clu(T_{ik}^S)$).*

Proof. We can prove this lemma by contradiction. Suppose $Clu(T_i^{S^*})$ is the optimal solution to Eq. (4.7) and there exists a cluster $c \neq \emptyset$ such that $c \in Clu(T_i^{S^*})$ but $c \notin C_i$. Therefore, $c \notin T_{ik}^S, \forall k$. Let $Clu(T_i'^{S^*}) = Clu(T_i{}^{S^*}) - c$. Then based on the definition of RF distance, we have $\sum_{k=1}^p w_k^S \sum_{i=1}^n RF(T_i{}^{S^*}, T_{ik}) > \sum_{k=1}^p w_k^S \sum_{i=1}^n RF(T_i'^{S^*}, T_{ik}^S)$, which contradicts the assumption that $Clu(T_i^{S^*})$ is the optimal solution. □

This property suggests that the search space of the solution to Eq. (4.7) is C_i. That is, all clusters in the aggregated tree must be present in at least one of the input CPTs.

LEMMA 4.2. *For any cluster c, if $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) > 0.5 * \sum_{k=1}^p w_k^S$, then $c \in Clu(T_i^{S^*})$, and if $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) < 0.5 * \sum_{k=1}^p w_k^S$, then $c \notin Clu(T_i^{S*})$, where $\mathbb{1}(\cdot)$ is the indicator function.*

Proof. The proof is similar to the proof for Lemma 4.1. We can prove the two statements by contradiction. □

Therefore, for the optimal solution, the clusters that have more than 50% weighted support from all the input CPTs should be included in the aggregated tree.

LEMMA 4.3. *For any cluster c_1 and c_2, if $\sum_{k=1}^p w_k^S \mathbb{1}(c_1 \in T_{ik}^S) > 0.5 * \sum_{k=1}^p w_k^S$ and $\sum_{k=1}^p w_k^S \mathbb{1}(c_2 \in T_{ik}^S) > 0.5 * \sum_{k=1}^p w_k^S$, then c_1 and c_2 must be compatible.*

Proof. Note that for any constituency parse tree T_{ik}^S, its clusters must be compatible. Therefore, for a cluster c, all its non-compatible clusters can only occur in trees that c is not occurred. If $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) > 0.5 * \sum_{k=1}^p w_k^S$, then $\forall c'$ not compatible with c, $\sum_{k=1}^p w_k^S \mathbb{1}(c' \in T_{ik}^S) < 0.5 * \sum_{k=1}^p w_k^S$, and based on Lemma 4.2, $c' \notin Clu(T_i^{S^*})$. □

There is a special case when $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) = 0.5 * \sum_{k=1}^p w_k^S$. To consider this situation, we add the compatibility constraint as follows:

$$c_1 \cap c_2 = \emptyset, or\ c_1 \subset c_2,\ or\ c_2 \subset c_1, \forall c_1, c_2 \in Clu(T_i^{S^*}). \qquad (4.8)$$

This constraint ensures that the aggregated tree follows the syntactic structure requirement of constituency parsing. Therefore, all the clusters in the aggregated tree should be *compatible*, which means they should either be disjoint or a proper subset.

In the cases where $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) = 0.5 * \sum_{k=1}^p w_k^S$, we propose to find the maximum number of compatible clusters to add into the aggregated tree $Clu(T_i^{S^*})$. Although adding these clusters into the constructed aggregation tree does not affect the resulting total RF distance, we favor the aggregated trees with more compatible clusters since they contain as many details from the input trees. We conduct the following steps. First we form a set C_i' that includes all clusters such that $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) = 0.5 * \sum_{k=1}^p w_k^S$. Then we construct the incompatibility graph by treating the clusters as nodes and adding an edge if two clusters are not compatible. Finding the maximum number of compatible clusters is then equivalent to the maximum independent set problem [45]. This strong NP-hard problem can be addressed by the existing methods [25, 18].

Based on the properties of the optimal solution, we construct the aggregated tree $T_i^{S^*}$ as follows. We compute the weighted support for each cluster c in C_i. If $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) > 0.5 * \sum_{k=1}^p w_k^S$ then c is added to the aggregated tree $Clu(T_i^{S^*})$. If $\sum_{k=1}^p w_k^S \mathbb{1}(c \in T_{ik}^S) = 0.5 * \sum_{k=1}^p w_k^S$ then we find maximum number of compatible clusters C_i^m by solving the maximum independent set problem. We then add these clusters to the aggregated tree

Algorithm 1 Optimal solution to Eq. (4.7)

```
Input: The set of unique clusters in all input CPTs for i-th sentence
  ($C_i$), weights ($w_k^S$).
  Aggregated CPT ($T_i^{S^*}$).
  $Clu(T_i^{S^*}) = \emptyset$ ;
  $C_i' = \emptyset$;
  for $c$ in $C_i$ do
      if $\sum_{k=1}^{p} w_k^S \mathbb{1}(c \in T_{ik}^S) > 0.5 * \sum_{k=1}^{p} w_k^S$ then
          $Clu(T_i^{S^*}) = Clu(T_i^{S^*}) \bigcup c$
      if $\sum_{k=1}^{p} w_k^S \mathbb{1}(c \in T_{ik}^S) = 0.5 * \sum_{k=1}^{p} w_k^S$ then
          $C_i' = C_i' \bigcup c$;
  Construct incompatibility graph $g$ for $C_i'$;
  $C_i^m$ = Maximum-Independent-Set($g$);
  $Clu(T_i^{S^*}) = Clu(T_i^{S^*}) \bigcup C_i^m$
  return $T_i^{S^*}$
```

$Clu(T_i^{S^*})$. Finally, we re-order the clusters in $Clu(T_i^{S^*})$ to form $T_i^{S^*}$. The pseudo-code of our proposed algorithm is given in Algorithm (1).

THEOREM 4.1. *The aggregated tree $T_i^{S^*}$ calculated by Algorithm (1) is the optimal solution to the following problem:*

$$T_i^{S^*} \leftarrow \underset{T_i^{S^*}}{\operatorname{argmin}} \sum_{k=1}^{p} w_k^S \sum_{i=1}^{n} RF(T_i^{S^*}, T_{ik}^S)$$

such that

$$c_1 \cap c_2 = \emptyset, \textit{or } c_1 \subset c_2, \textit{ or } c_2 \subset c_1, \forall c_1, c_2 \in Clu(T_i^{S^*}).$$

Proof. In the Algorithm (1), we consider all the clusters with $\sum_{k=1}^{p} w_k^S \mathbb{1}(c' \in T_{ik}^S) > 0.5 * \sum_{k=1}^{p} w_k^S$ and add them to $T_i^{S^*}$. From Lemma (4.3), we show that all of these clusters are compatible and from Lemma (4.2), we show that adding these clusters minimizes the RF distance. Adding all these clusters result in the minimum RF distance implying that the objective function will be minimized. Applying maximum independent set algorithm on the incompatibility graph provides us with the maximum number of compatible clusters for $\sum_{k=1}^{p} w_k^S \mathbb{1}(c' \in T_{ik}^S) = 0.5 * \sum_{k=1}^{p} w_k^S$. Adding all these clusters to $T_i^{S^*}$ results in obtaining the maximum set of compatible clusters. Thus the solution is optimal. □

4.2.2 Time Complexity

LEMMA 4.4. *The incompatibility graph constructed for clusters with weighted support equal to 50% is bipartite.*

Proof. Let $C_1, \ldots, C_k$ be the clusters with 50% support from the input constituency parse trees. The set of all constituency parse trees with respect to structure is denoted by $\mathcal{T}^S = \{T_{i1}^S, T_{i2}^S, T_{i3}^S, \ldots, T_{ik}^S\}$.

Assume that a cluster C_i is supported by trees $\mathcal{T}_i^S \subset \mathcal{T}^S$. If C_i is not compatible with C_j, then it implies that $\mathcal{T}_i^S = \mathcal{T}^S \setminus \mathcal{T}_j^S$. Otherwise, $\mathcal{T}_i^S$ would have a non-empty intersection with $\mathcal{T}_j^S$, which would imply that there is a tree $\mathcal{T}_s^S$ that supports both C_i and C_j, which contradicts with the assumption that C_i and C_j are incompatible.

We prove Lemma 4.4 by contradiction. Let's assume that the incompatibility graph G for clusters $C_1, \ldots, C_k$ is *not bipartite*. It means that G contains an odd-cycle. Without loss of generality assume that this cycle is $(C_1, C_2, \ldots, C_{2p+1})$. That is, C_2 is not compatible with C_1, C_3 is not compatible with C_2, and so on. Then, by our previous observation, C_2 must be supported by $\mathcal{T}^S \setminus \mathcal{T}_1^S$, C_3 must be supported by $\mathcal{T}_1^S$, and so on. That is, for odd $i \leq 2p+1$, C_i must be supported by $\mathcal{T}_1^S$. Then C_{2p+1} and C_1 are supported by the same set of trees, which means that C_1 and C_{2p+1} must be compatible. This is a contradiction (i.e., $(C_1, C_2, \ldots, C_{2p+1})$ could not be a cycle). □

The existing methods [18] solve the maximum independent set problem for a bipartite graph with time complexity of $O(z^{2.5} + (outputsize))$ where z refers to the number of nodes in the incompatibility graph. As the expected output is the list of compatible clusters, the output size is in the order of $O(z)$. The for loop that iterates over cluster set C_i runs in $O(|C_i|)$ time. Therefore, the overall run time of Algorithm (1) is $O(|C_i| + z^{2.5} + z)$. However, in practice, z is very small compared to $|C_i|$ because it only contains clusters with support equal to 50%. Thus, Algorithm (1) has, in practice, near-linear run time in $|C_i|$.

4.3 Constituent Label Aggregation

After obtaining the aggregated structures, we aggregate the corresponding labels provided by the parsers. In constituent label aggregation step, we aim to minimize the objective function Eq. (4.9) with respect to the $\mathcal{L}_{Clu(T_i^{S^*})}$ and $\mathcal{W}^l$, where $\mathcal{L}_{Clu(T_i^{S^*})}$ refers to the labels associated to the aggregated structure, and $\mathcal{W}^l = \{w_1^l, w_2^l, \ldots, w_p^l\}$ refers to the set of all parsers' weights with respect to the constituent labels, as follows:

$$\min_{\mathcal{T}^*, \mathcal{W}^l} f(\mathcal{T}^*, \mathcal{W}^l) = \sum_{k=1}^{p} w_k^l \sum_{i=1}^{n} d(\mathcal{L}_{Clu(T_i^{S^*})}, \mathcal{L}_{Clu(T_{ik}^S)}), \tag{4.9}$$

where $\mathcal{L}_{Clu(T_{ik}^S)}$ refers to the constituent labels provided by parsers for the obtained clusters in $T_i^{S^*}$. Accordingly, we show the weight update by taking differentiation with respect to $\mathcal{W}^l$ in Eq. (4.10):

$$w_k^l = -log\left(\frac{\sum_i d(\mathcal{L}_{Clu(T_i^{S^*})}, \mathcal{L}_{Clu(T_{ik}^S)})}{max_k \sum_i d(\mathcal{L}_{Clu(T_i^{S^*})}, \mathcal{L}_{Clu(T_{ik}^S)})}\right), \tag{4.10}$$

where d refers to the zero-one loss function. Similarly, the constituent label aggregation update is shown in Eq. (4.11):

$$T_i^* \leftarrow \underset{\mathcal{L}_{Clu(T_i^{S^*})}}{\operatorname{argmin}} \sum_{k=1}^{p} w_k^l \sum_{i=1}^{n} d(\mathcal{L}_{Clu(T_i^{S^*})}, \mathcal{L}_{Clu(T_{ik}^S)}). \tag{4.11}$$

5 Experiments

In this section, we conduct experiments on various datasets with different languages from different domains[1]. We start with the datasets in Section 5.1.The baseline methods and evaluations are discussed in Sections 5.2 and 5.3, respectively. We demonstrate the main experimental results in Section 5.4 and ablation studies in Section 5.5.

[1] Our implementation code is available at `https://github.com/kulkarniadithya/CPTAM`

Datasets	Language	Sentence	#token/sentence
Penn Treebank-3	English	49208	24.70
OntoNotes	English	143709	18.59
	Chinese	51230	17.64
Genia	English	18541	28.09
TIGER Corpus	German	40020	17.06
French Treebank	French	21550	24.80

Table 3: Statistics of Datasets

5.1 Datasets

We use six benchmark datasets from different domains and different languages for evaluation.

Penn Treebank-3[2] selected 2,499 stories from a three-year Wall Street Journal collection in English for syntactic annotation.

OntoNotes[3] consists of a large corpus comprising various genres of text (e.g., news, weblogs, Usenet newsgroups, broadcast, and talk shows) with structural information in three languages (English, Chinese, and Arabic). The Arabic portion of the dataset is not included in our experiments since the parsers' tokenization does not align with the ground truth.

Genia[4] is constructed from research abstracts in the molecular biology domain. Approximately 2500 abstracts are annotated from the MEDLINE database.

TIGER Corpus[5] consists of approximately 40,000 sentences from the German newspaper "Frankfurter Rundschau". The corpus was annotated with part-of-speech and syntactic structures in the project TIGER (DFG).

French Treebank[6] consists of approximately 22000 sentences from the articles of French newspaper "Le Monde".
Table 3 summarizes the statistics of the datasets.

5.2 Baselines

We compare CPTAM with two categories of baselines. The first category of baselines is the individual state-of-the-art input constituency parsers including **CoreNLP** [27], **Berkeley**[7] [19], **AllenNLP**[8] [14], and **HanLP** [17]. We have chosen these parsers as they are the most "stars" NLP libraries on GitHub, demonstrating their wide applications in industry and academia.

The second category of baselines is the tree aggregation methods[9] including

- **Majority Rule Consensus (MRC)** [28]. It constructs aggregation trees containing clusters with support greater than 50%.
- **Greedy Consensus (GC)** [8]. The aggregated trees are constructed progressively to have all the clusters whose support is above a threshold (30% for OntoNotes Chinese,TIGER Corpus, and French Treebank, and 20% for the other datasets) and compatible with the constructed tree. With these thresholds, this baseline essentially constructs aggregation trees with all compatible clusters from input trees.
- **Strict Consensus (SC)** [8]. It constructs aggregation trees containing clusters with support of 100%.

These methods only consider the aggregation of tree structures but not labels. Therefore, we apply **Majority Voting (MV)** to aggregate the labels after the tree aggregation step, where the label with the highest frequency is chosen for each cluster. We also compare with **CPTAM-W**, which is **CPTAM** without weight estimation. CPTAM-W considers clusters with support greater than or equal to 50%; thus, it is more aggressive compared to MRC, which considers clusters with support greater than 50% only, and more conservative compared to GC, which includes all compatible clusters.

5.3 Evaluation Measurements

The performance is evaluated by different standard metrics in the experiments. To evaluate the performance based on the real-life usage of constituency parsers, we also include the POS tags of individual tokens as part of the parsing results. Therefore, the following evaluation metric is stricter than *Evalb*, the standard metric for evaluating phrase structure. We report Precision, Recall, and F1 as follows:

$$Precision(P) = \frac{\#Correct\ Constituents}{\#Constituents\ in\ parser\ output} \tag{5.12}$$

$$Recall(R) = \frac{\#Correct\ Constituents}{\#Constituents\ in\ gold\ standard} \tag{5.13}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{5.14}$$

Accordingly, the same metrics Precision (P^S), Recall (R^S), and F1 ($F1^S$) are defined to evaluate the performance considering only the tree structure.

5.4 Experimental Results

The experimental results for CPT aggregation performance on Penn Treebank-3, OntoNotes (English and Chinese), and Genia are summarized in Table 4. Our experiments consider the scenario where the users only have access to the parsers but do not have any prior knowledge about their performance. Since there is no prior knowledge about parser performance on different datasets or languages, we consider the freely available state-of-the-art parsers to obtain initial parsing results. For French Treebank and TIGER Corpus datasets, as the ground truth labels are different from the labels provided by the parsers, we do not consider them for constituent label aggregation[10].

Among the input parsers, it is clear that none of them consistently perform the best on all datasets. Specifically, Hanlp performs poorly on English but the best on Chinese. This may be caused since Hanlp targets the Chinese language even though it is software for multiple languages. Allennlp performs the best on Penn Treebank-3 and OntoNotes (English) among the four parsers but does not perform well on the Genia dataset in the biomedical domain.

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

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

[4] https://github.com/allenai/genia-dependency-trees/tree/master/original_data

[5] https://www.ims.uni-stuttgart.de/documents/ressourcen/korpora/tiger-corpus/download/start.html

[6] http://ftb.llf-paris.fr/telecharger.php?langue=en

[7] We use the pretrained model provided by spaCy

[8] This parser can parse sentences in English only.

[9] We apply the implementations from https://evolution.genetics.washington.edu/phylip/getme-new1.html

[10] Taking CoreNLP output as an example in TIGER corpus, a chunk of the sentence is tagged as *(NUR (S (NOUN Konzernchefs) (VERB lehnen)* while the ground truth for the same span is *(NN-SB Konzernchefs) (VVFIN-HD lehnen)*

	Penn Treebank-3			OntoNotes (English)			OntoNotes (Chinese)			Genia		
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
CoreNLP	81.35	83.16	82.25	77.68	77.66	77.67	86.58	86.77	86.67	66.38	72.46	69.29
Berkeley	91.13	93.27	92.19	80.31	79.04	79.67	77.66	69.93	73.59	69.49	72.82	71.12
AllenNLP	93.08	**94.96**	**94.01**	80.99	80.47	80.73	-	-	-	68.00	69.54	68.76
Hanlp	18.42	19.21	18.81	21.61	21.90	21.75	**92.56**	**92.63**	**92.59**	28.65	29.19	28.92
MRC + MV	89.75	89.79	89.77	78.11	73.75	75.87	89.05	89.07	89.06	68.92	69.12	69.02
GC + MV	88.76	90.19	89.47	76.85	75.51	76.17	87.90	89.99	88.93	66.22	70.36	68.22
SC + MV	90.46	85.14	87.72	78.95	66.72	72.32	89.79	87.78	88.77	70.82	59.15	64.46
CPTAM-W	90.79	90.13	90.46	78.87	74.87	76.82	89.19	89.11	89.15	70.02	69.92	69.97
CPTAM	**93.58**	93.46	93.52	**81.94**	**80.72**	**81.33**	91.55	91.47	91.51	**71.43**	**72.43**	**71.93**

Table 4: CPT aggregation performance comparison on Penn Treebank-3, OntoNotes (English, Chinese), and Genia datasets.

	Penn Treebank-3				OntoNotes (English)				Genia				OntoNotes (Chinese)				French Treebank		TIGER Corpus	
	$F1^S$	w^S	Acc^l	w^l	$F1^S$	w^S	Acc^l	w^l	$F1^S$	w^S	Acc^l	w^l	$F1^S$	w^S	Acc^l	w^l	$F1^S$	w^S	$F1^S$	w^S
CoreNLP	3	3	3	3	3	3	3	3	3	3	1	1	1	1	2	2	1	1	2	2
Berkeley	2	2	2	2	2	2	2	2	1	1	2	2	3	3	3	3	2	2	1	1
Allennlp	1	1	1	1	1	1	1	1	2	2	3	3	-	-	-	-	-	-	-	-
HanLP	4	4	4	4	4	4	4	4	4	4	4	4	2	2	1	1	3	3	3	3

Table 5: The comparison between the rankings of parsers' performance with the rankings of estimated weights

The proposed CPTAM significantly outperforms all the state-of-the-art aggregation methods in terms of Precision, Recall, and F1 score, demonstrating the power of the proposed method in resolving the conflicts between input CPTs. Comparing CPTAM-W and CPTAM, CPTAM further improves in all metrics, indicating the necessity and effectiveness of the weight estimation in the truth discovery process.

Compared with individual parsers, CPTAM performs consistently well to represent the constituency grammar of the sentence in all datasets. CPTAM performs the best for two out of four datasets and remains competitive on the other two datasets. In contrast, AllenNLP and Hanlp are the best for one out of four datasets, and CoreNLP and Berkeley are not the best in any datasets. This shows that the proposed CPTAM can consistently obtain high-quality aggregated CPTs over different languages and domains.

Further, we study the accuracy of the weight estimations of CPTAM. We compare the rankings given by the estimated weights with the rankings of parsers' real quality, and the results are shown in Table 5. To evaluate the weights estimated for the structure aggregation, we compute the rank of parsers' quality by their structure F1 scores ($F1^S$) compared with the ground truth and by the weight estimation w_k^S computed in Eq. (4.6), where the numbers indicate the rank. Similarly, for the label aggregation, we compute the rank of parsers' quality by their label accuracy (Acc^l) and by the weight estimation w_k^l computed in Eq. (4.10).

It is clear that the parsers' quality varies across different languages and domains. The ranks of parsers are exactly the same between their real quality and the estimated weights. It illustrates that the weight calculated by the proposed CPTAM properly estimates parsers' quality in the absence of ground truth. These experiments also suggest that parser users can first apply CPTAM on the sampled corpus to estimate the quality of individual parsers on the given corpus and then use the best parser to achieve high-quality parsing results and high efficiency.

5.5 Ablation Study

To gain insights into our framework, we investigate the effectiveness of the tree structure aggregation step as it is the foundation of CPTAM. To evaluate the performance on the structure, the RF distance ($RFdist.$) is calculated between the parser output and ground truth. We also calculate Precision (P^S), Recall (R^S), and F1 score ($F1^S$) considering the tree structure only. The ablation study results are shown in Table 6 and Table 7.

Table 6 and Table 7 illustrate a strong correlation between the RF distance and F1 score on all datasets. The lower the RF distance, the higher the F1 score. This correlation indicates that RF distance is a proper measurement for the quality of constituency parse trees. CPTAM outperforms all aggregation baseline approaches on all datasets. It consistently identifies proper clusters in the tree by correctly estimating the parsers' quality. As a result, CPTAM outperforms or stays competitive compared to the best parser on all datasets as well.

6 Conclusion

This paper adopts the truth discovery idea to aggregate CPTs from different parsers by estimating the parsers' reliability in the absence of ground truth. We aggregate the input CPTs in two steps: tree structure aggregation and constituent label aggregation. The block coordinate descent method is applied to obtain solutions through an iterative process, and an optimal solution is derived to construct aggregation trees that can minimize the weighted RF distance. We further provide theoretical analysis to show that the proposed approach gives the optimal solution. The proposed solution has near-linear run time in practice for the tree structure aggregation step. Our experimental results illustrate that the proposed solution CPTAM outperforms the state-of-the-art aggregation baselines

	Penn Treebank-3				OntoNotes (English)				Genia			
	$RFdist.$	P^S	R^S	$F1^S$	$RFdist.$	P^S	R^S	$F1^S$	$RFdist.$	P^S	R^S	$F1^S$
CoreNLP	436250	84.76	86.76	85.75	890401	87.58	89.03	88.30	341925	77.75	85.70	81.53
Berkeley	194282	94.40	96.16	95.27	790028	89.66	89.88	89.77	273796	82.84	**87.34**	85.03
AllenNLP	**154795**	95.41	**97.29**	**96.34**	696054	90.55	**91.75**	91.15	298548	82.23	83.90	83.06
Hanlp	1709148	59.26	60.69	59.97	3119630	64.71	66.24	65.47	521522	69.26	69.50	69.38
MRC	265967	93.23	92.79	93.01	1170233	87.98	83.18	85.51	307761	83.05	82.78	82.91
GC	266406	91.94	93.98	92.95	1028983	85.77	85.88	85.82	316758	78.93	84.16	81.46
SC	320215	93.77	87.53	90.54	1323468	88.39	75.33	81.34	370522	83.70	67.09	74.48
CPTAM-W	243413	93.12	93.94	93.53	1007384	87.85	84.63	86.21	293364	82.95	84.01	83.47
CPTAM	176187	**95.92**	95.73	95.82	**641058**	**91.28**	91.71	**91.49**	**264153**	**84.31**	85.96	**85.13**

Table 6: The tree structure aggregation performance comparison on Penn Treebank-3, OntoNotes (English), and Genia datasets

	OntoNotes (Chinese)				French Treebank				TIGER Corpus			
	$RFdist.$	P^S	R^S	$F1^S$	$RFdist.$	P^S	R^S	$F1^S$	$RFdist.$	P^S	R^S	$F1^S$
CoreNLP	**108817**	**96.47**	**96.72**	**96.59**	**212550**	**91.06**	**91.56**	**91.31**	740059	65.67	80.02	72.14
Berkeley	406708	92.90	84.16	88.31	344070	85.65	77.07	81.13	**220183**	**93.97**	**85.36**	**89.46**
HanLP	144618	95.73	95.78	95.75	1196428	44.90	37.82	41.06	818677	66.22	62.36	64.23
MRC	200008	95.41	94.04	94.72	231755	90.06	88.05	89.04	229139	92.95	82.79	87.58
GC	206438	93.93	95.06	94.49	308335	83.80	89.08	86.35	315613	87.66	84.38	85.99
SC	213173	95.83	92.84	94.31	314548	90.97	79.30	84.74	293991	93.07	80.23	86.17
CPTAM-W	198769	95.34	94.20	94.77	229344	90.8	88.35	89.56	228845	92.83	83.97	88.18
CPTAM	114733	96.39	96.49	96.44	212697	91.05	91.54	91.29	**220183**	**93.97**	**85.36**	**89.46**

Table 7: The tree structure aggregation performance comparison on OntoNotes (Chinese), French Treebank and TIGER Corpus datasets

and consistently obtains high-quality aggregated CPTs for various datasets in the absence of ground truth. We further illustrate that our adopted weight update correctly estimates parsers' quality. Empirically, the importance of the tree structure aggregation step is demonstrated in the ablation study. Overall, we present the effectiveness of the proposed CPTAM across different languages and domains.

7 Acknowledgement

The work was supported in part by the National Science Foundation under Grant NSF IIS-2007941. Any opinions, findings, and conclusions, or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation hereon.

References

[1] A Abeillé, L Clément, and L Liégeois. Un corpus arboré pour le français: Le french treebank. *Traitement Automatique des Langues*, 60(3):19–43, 2019.

[2] Edward N Adams III. Consensus techniques and the comparison of taxonomic trees. *Systematic Biology*, 21(4):390–397, 1972.

[3] Giuseppe Attardi and Felice Dell'Orletta. Reverse revision and linear tree combination for dependency parsing. In *Proc. of NAACL*, pages 261–264, 2009.

[4] Olaf RP Bininda-Emonds. *Phylogenetic supertrees: Combining information to reveal the tree of life*, volume 4. Springer, 2004.

[5] Olaf RP Bininda-Emonds, John L Gittleman, and Mike A Steel. The (super) tree of life: Procedures, problems, and prospects. *Annual Review of Ecology and Systematics*, 33(1):265–289, 2002.

[6] S Brants, S Dipper, P Eisenberg, S Hansen-Schirra, E König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. Tiger: Linguistic interpretation of a german corpus. *Research on language and computation*, 2(4):597–620, 2004.

[7] Kåre Bremer. Combinable component consensus. *Cladistics*, 6(4):369–372, 1990.

[8] D Bryant. A classification of consensus methods for phylogenetics. *DIMACS series in discrete mathematics and theoretical computer science*, 61:163–184, 2003.

[9] William HE Day. Optimal algorithms for comparing trees with labeled leaves. *Journal of classification*, 2(1):7–28, 1985.

[10] Glenn De'Ath. Boosted trees for ecological modeling and prediction. *Ecology*, 88(1):243–251, 2007.

[11] Erik D Demaine, Shay Mozes, Benjamin Rossman, and Oren Weimann. An optimal decomposition algorithm for tree edit distance. *ACM Transactions on Algorithms (TALG)*, 6(1):1–19, 2009.

[12] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proc. of KDD*, pages

601–610, 2014.

[13] J Felsenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

[14] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proc. of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, 2018.

[15] Pablo A Goloboff and Diego Pol. Semi-strict supertrees. *Cladistics*, 18(5):514–525, 2002.

[16] Johan Hall, Jens Nilsson, and Joakim Nivre. Single malt or blended? a study in multilingual parser optimization. *Trends in Parsing Technology*, pages 19–33, 2010.

[17] Han He. HanLP: Han Language Processing, 2020.

[18] T Kashiwabara, S Masuda, K Nakajima, and T Fujisawa. Generation of maximum independent sets of a bipartite graph and maximum cliques of a circular-arc graph. *Journal of algorithms*, 13(1):161–174, 1992.

[19] N Kitaev and D Klein. Constituency parsing with a self-attentive encoder. In *Proc. of ACL*, pages 2676–2686, 2018.

[20] A Kuncoro, M Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A. Smith. Distilling an ensemble of greedy dependency parsers into one MST parser. In *Proc. of EMNLP*, pages 1744–1753, 2016.

[21] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. A confidence-aware approach for truth discovery on long-tail data. *PVLDB*, 8(4), 2014.

[22] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proc. of SIGMOD*, pages 1187–1198, 2014.

[23] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *ACM SigKDD Explorations Newsletter*, 17(2):1–16, 2016.

[24] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. On the discovery of evolving truth. In *Proc. of KDD*, pages 675–684, 2015.

[25] J Liu. Maximal independent sets in bipartite graphs. *Journal of graph theory*, 17(4):495–507, 1993.

[26] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proc. of KDD*, pages 745–754, 2015.

[27] Christopher D Manning, M Surdeanu, J Bauer, Jenny R Finkel, S Bethard, and D McClosky. The stanford corenlp natural language processing toolkit. In *Proc. of ACL: System demonstrations*, pages 55–60, 2014.

[28] T. Margush and F.R. McMorris. Consensusn-trees. *Bulletin of Mathematical Biology*, 43(2):239 – 244, 1981.

[29] An T Nguyen, Byron C Wallace, J Li, A Nenkova, and M Lease. Aggregating and predicting sequence labels from crowd annotations. In *Proc. of ACL*, page 299, 2017.

[30] Joakim Nivre and Ryan McDonald. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL*, pages 950–958, 2008.

[31] T Ohta, Y Tateisi, J Kim, H Mima, and J Tsujii. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proc. of the human language technology conference*, pages 73–77, 2002.

[32] Sameer S Pradhan and N Xue. Ontonotes: The 90% solution. In *Proc. of NAACL*, pages 11–12, 2009.

[33] P Probst, Marvin N Wright, and A Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301, 2019.

[34] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *Proc. of the American Control Conference*, pages 1859–1864, 2005.

[35] David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

[36] Nasim Sabetpour, Adithya Kulkarni, and Qi Li. Optsla: An optimization-based approach for sequential label aggregation. In *Proc. of EMNLP: Findings*, pages 1335–1340, 2020.

[37] Nasim Sabetpour, Adithya Kulkarni, Sihong Xie, and Qi Li. Truth discovery in sequence labels from crowds. In *Proc. of ICDM*, 2021.

[38] Kenji Sagae and Alon Lavie. Parser combination by reparsing. In *Proc. of NAACL*, pages 129–132, 2006.

[39] Stefan Schwarz, Mateusz Pawlik, and Nikolaus Augsten. A new perspective on the tree edit distance. In *SISAP*, pages 156–170. Springer, 2017.

[40] C Semple, M Steel, et al. *Phylogenetics*, volume 24. Oxford University Press on Demand, 2003.

[41] Zhi Shi, Fan Yang, Zheyi Zhu, Qi Li, Zhaoran Wang, and Jiawei Han. Dynamic truth discovery on numerical data. In *Proc. of ICDM*, pages 817–826, 2018.

[42] Edwin D. Simpson and Iryna Gurevych. A Bayesian approach for sequence tagging with crowds. In *Proc. of EMNLP-IJCNLP*, pages 1093–1104, 2019.

[43] Robert R Sokal and F James Rohlf. Taxonomic congruence in the leptopodomorpha re-examined. *Systematic Zoology*, 30(3):309–325, 1981.

[44] Mihai Surdeanu and Christopher D Manning. Ensemble models for dependency parsing: Cheap and good? In *Proc. of NAACL*, pages 649–652, 2010.

[45] Robert Endre Tarjan and Anthony E Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977.

[46] A Taylor, M Marcus, and B Santorini. The penn treebank: An overview. In *Treebanks*, pages 5–22. Springer, 2003.

[47] L Yao, L Su, Q Li, Y Li, F Ma, J Gao, and A Zhang. Online truth discovery on time series data. In *Proc. of SDM*, pages 162–170. SIAM, 2018.

[48] Xiaoxin Yin, Jiawei Han, and S Yu Philip. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):796–808, 2008.

[49] D. Zhou, J. C. Platt, S. Basu, and Y. Mao. Learning from the wisdom of crowds by minimax entropy. In *NeurIPS*, pages 2204–2212, 2012.