Unsupervised Discourse Constituency Parsing Using Viterbi EM

Noriki Nishida and **Hideki Nakayama**

Graduate School of Information Science and Technology
The University of Tokyo
{nishida,nakayama}@nlab.ci.i.u-tokyo.ac.jp

Abstract

In this paper, we introduce an unsupervised discourse constituency parsing algorithm. We use Viterbi EM with a margin-based criterion to train a span-based discourse parser in an unsupervised manner. We also propose initialization methods for Viterbi training of discourse constituents based on our prior knowledge of text structures. Experimental results demonstrate that our unsupervised parser achieves comparable or even superior performance to fully supervised parsers. We also investigate discourse constituents that are learned by our method.

1 Introduction

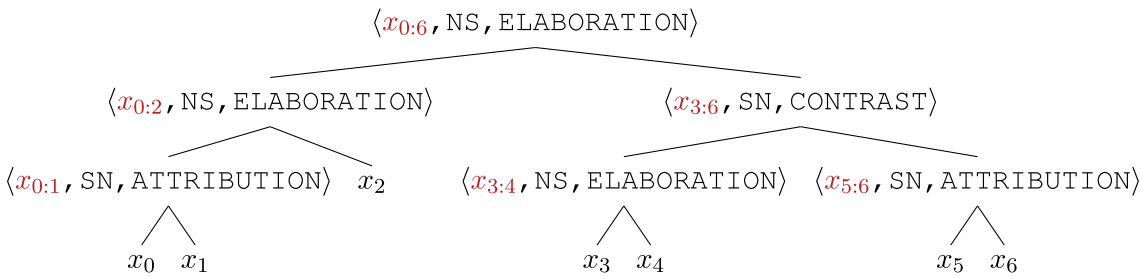
Natural language text is generally coherent (Halliday and Hasan, 1976) and can be analyzed as discourse structures, which formally describe how text is coherently organized. In discourse structure, linguistic units (e.g., clauses, sentences, or larger textual spans) are connected together semantically and pragmatically, and no unit is independent nor isolated. Discourse parsing aims to uncover discourse structures automatically for given text and has been proven to be useful in various NLP applications, such as document summarization (Marcu, 2000; Louis et al., 2010; Yoshida et al., 2014), sentiment analysis (Polanyi and Van den Berg, 2011; Bhatia et al., 2015), and automated essay scoring (Miltsakaki and Kukich, 2004).

Despite the promising progress achieved in recent decades (Carlson et al., 2001; Hernault et al., 2010; Ji and Eisenstein, 2014; Feng and Hirst, 2014; Li et al., 2014; Joty et al., 2015; Morey et al., 2017), discourse parsing still remains a significant challenge. The difficulty is due in part to shortage and low reliability of hand-annotated discourse structures. To develop a better-generalized parser, existing algorithms require a larger amounts of training data. However, manually annotating discourse structures is expensive, time-consuming, and sometimes highly ambiguous (Marcu et al., 1999).

One possible solution to these problems is grammar induction (or unsupervised syntactic parsing) algorithms for discourse parsing. However, existing studies on unsupervised parsing mainly focus on sentence structures, such as phrase structures (Lari and Young, 1990; Klein and Manning, 2002; Golland et al., 2012; Jin et al., 2018) or dependency structures (Klein and Manning, 2004; Berg-Kirkpatrick et al., 2010; Naseem et al., 2010; Jiang et al., 2016), though text-level structural regularities can also exist beyond the scope of a single sentence. For instance, in order to convey information to readers as intended, a writer should arrange utterances in a coherent order.

We tackle these problems by introducing ***unsupervised discourse parsing***, which induces discourse structures for given text without relying on human-annotated discourse structures. Based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), which is one of the most widely accepted theories of discourse structure, we assume that coherent text can be represented as tree structures, such as the one in Figure 1. The leaf nodes correspond to non-overlapping clause-level text spans called *elementary discourse units* (EDUs). Consecutive text spans are combined to each other recursively in a bottom–up manner to form larger text spans (represented by internal nodes) up to a global document span. These text spans are called *discourse constituents*. The internal nodes are labeled with both nuclearity statuses (e.g., *Nucleus-Satellite* or NS) and rhetorical

Transactions of the Association for Computational Linguistics, vol. 8, pp. 215–230, 2020. https://doi.org/10.1162/tacl_a_00312
Action Editor: Yuji Matsumoto. Submission batch: 07/2019; Revision batch: 12/2019; Published 4/2020.



[This maker of electronic devices said]$_{x_0}$ [it replaced all five incumbent directors at a special meeting.]$_{x_1}$ [Elected as directors were Mr. Hollander, . . . , and Rose Pothier.]$_{x_2}$ [Newport officials didn't respond Friday to requests]$_{x_3}$ [to discuss the changes at the company]$_{x_4}$ [but earlier, Mr Weekes had said]$_{x_5}$ [Mr. Hollander wanted to have his own team on the board.]$_{x_6}$

Figure 1: An example of RST-based discourse constituent structure we assume in this paper. Leaf nodes x_i correspond to non-overlapping clause-level text segments, and internal nodes consists of three complementary elements: discourse constituents $x_{i:j}$, discourse nuclearities (e.g., NS), and discourse relations (e.g., ELABORATION).

relations (e.g., ELABORATION, CONTRAST) that hold between connected text spans.

In this paper, we especially focus on unsupervised induction of an unlabeled discourse constituent structure (i.e., a set of unlabeled discourse constituent spans) given a sequence of EDUs, which corresponds to the first tree-building step in conventional RST parsing. Such constituent structures provide hierarchical information of input text, which is useful in downstream tasks (Louis et al., 2010). For instance, a constituent structure [*X* [*Y Z*]] indicates that text span *Y* is preferentially combined with *Z* (rather than *X*) to form a constituent span, and then the text span [*Y Z*] is connected with *X*. In other words, this structure implies that [*X Y*] is a distituent span and requires *Z* to become a constituent span. Our challenge is to find such discourse-level constituentness from EDU sequences.

The core hypothesis of this paper is that discourse tree structures and syntactic tree structures share the same (or similar) constituent properties at a metalevel, and thus, learning algorithms developed for grammar inductions are transferable to unsupervised discourse constituency parsing by proper modifications. Actually, RST structures can be formulated in a similar way as phrase structures in the Penn Treebank, though there are a few differences: The leaf nodes are not words but EDUs (e.g., clauses), and the internal nodes do not contain phrase labels but hold nuclearity statuses and rhetorical relations.

The expectation-maximization (EM) algorithm (Klein and Manning, 2004) has been the dominating unsupervised learning algorithm for grammar induction. Based on our hypothesis and this fact, we develop a span-based discourse parser (in an unsupervised manner) by using Viterbi EM (or ''hard'' EM) (Neal and Hinton, 1998; Spitkovsky et al., 2010; DeNero and Klein, 2008; Choi and Cardie, 2007; Goldwater and Johnson, 2005) with a margin-based criterion (Stern et al., 2017; Gaddy et al., 2018).[1] Unlike the classic EM algorithm using inside-outside re-estimation (Baker, 1979), Viterbi EM allows us to avoid explicitly counting discourse constituent patterns, which are generally too sparse to estimate reliable scores of text spans.

The other technical contribution is to present effective initialization methods for Viterbi training of discourse constituents. We introduce initial-tree sampling methods based on our prior knowledge of document structures. We show that proper initialization is crucial in this task, as observed in grammar induction (Klein and Manning, 2004; Gimpel and Smith, 2012).

On the RST Discourse Treebank (RST-DT) (Carlson et al., 2001), we compared our parse trees with manually annotated ones. We observed that our method achieves a Micro F_1 score of 68.6% (84.6%) in the (corrected) RST-PARSEVAL

[1]Our code can be found at https://github.com/norikinishida/DiscourseConstituencyInduction-ViterbiEM.

(Marcu, 2000; Morey et al., 2018), which is comparable with or even superior to fully supervised parsers. We also investigated the discourse constituents that can or cannot be learned well by our method.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 gives the details of our parsing model and training algorithm. Section 4 describes the experimental setting and Section 5 discusses the experimental results. Conclusions are given in Section 6.

2 Related Work

The earliest studies that use EM in unsupervised parsing are Lari and Young (1990) and Carroll and Charniak (1992), which attempted to induce probabilistic context-free grammars (PCFG) and probabilistic dependency grammars using the classic inside–outside algorithm (Baker, 1979). Klein and Manning (2001b, 2002) perform a weakened version of *constituent tests* (Radford, 1988) by the Constituent-Context Model (CCM), which, unlike a PCFG, describes whether a contiguous text span (such as DT JJ NN) is a constituent or a distituent. The CCM uses EM to learn *constituenthood* over part-of-speech (POS) tags and for the first time outperformed the strong right-branching baseline in unsupervised constituency parsing. Klein and Manning (2004) proposed the Dependency Model with Valence (DMV), which is a head automata model (Alshawi, 1996) for unsupervised dependency parsing over POS tags and also relies on EM. These two models have been extended in various works for further improvements (Berg-Kirkpatrick et al., 2010; Naseem et al., 2010; Golland et al., 2012; Jiang et al., 2016).

In general, these methods use the inside–outside (dynamic programming) re-estimation (Baker, 1979) in the E step. However, Spitkovsky et al. (2010) showed that Viterbi training (Brown et al., 1993), which uses only the best-scoring tree to count the grammatical patterns, is not only computationally more efficient but also empirically more accurate in longer sentences. These properties are, thus, suitable for ‘‘document-level’’ grammar induction, where the document length (i.e., the number of EDUs) tends to be long.[2] In addition, as explained later in Section 3, we incorporate Viterbi EM with a margin-based criterion (Stern et al., 2017; Gaddy et al., 2018); this allows us to avoid explicitly counting each possible discourse constituent pattern symbolically, which is generally too sparse and appears only once.

Prior studies (Klein and Manning, 2004; Gimpel and Smith, 2012; Naseem et al., 2010) have shown that initialization or linguistic knowledge plays an important role in EM-based grammar induction. Gimpel and Smith (2012) demonstrated that properly initialized DMV achieves improvements in attachment accuracies by $20 \sim 40$ points (i.e., $21.3\% \rightarrow 64.3\%$), compared with the uniform initialization. Naseem et al. (2010) also found that controlling the learning process with the prior (universal) linguistic knowledge improves the parsing performance of DMV. These studies usually rely on insights on syntactic structures. In this paper, we explore discourse-level prior knowledge for effective initialization of the Viterbi training of discourse constituency parsers.

Our method also relies on recent work on RST parsing. In particular, one of the initialization methods in our EM training (in Section 3.3 (i)) is inspired by the inter-sentential and multi-sentential approach used in RST parsing (Feng and Hirst, 2014; Joty et al., 2013, 2015). We also follow prior studies (Sagae, 2009; Ji and Eisenstein, 2014) and utilize syntactic information, i.e., dependency heads, which contributes to further performance gains in our method.

The most similar work to that presented here is Kobayashi et al. (2019), who propose unsupervised RST parsing algorithms in parallel with our work. Their method builds an unlabeled discourse tree by using the CKY dynamic programming algorithm. The tree-merging (splitting) scores in CKY are defined as similarity (dissimilarity) between adjacent text spans. The similarity scores are calculated based on distributed representations using pre-trained embeddings. However, similarity between adjacent elements are not always good indicators of constituentness. Consider tag sequences ‘‘VBD IN’’ and ‘‘IN NN’’. The former is an example of a distituent sequence, whereas the latter is a constituent. ‘‘VBD’’, ‘‘IN’’, and ‘‘NN’’ may have similar distributed representations because these tags cooccur frequently in corpora. This implies that it is difficult to distinguish constituents and distituents if we use

[2] Prior studies on grammar induction generally use sentences up to length 10, 15, or 40. On the other hand, about half the documents in the RST-DT corpus (Carlson et al., 2001) are longer than 40.

only similarity (dissimilarity) measures. In this paper, we aim to mitigate this issue by introducing parameterized models to learn discourse constituentness.

3 Methodology

In this section, we first describe the parsing model we develop. Next, we explain how to train the model in an unsupervised manner by using Viterbi EM. Finally, we present the initialization methods we use for further improvements.

3.1 Parsing Model

The parsing problem in this study is to find the unlabeled constituent structure with the highest score for an input text $\boldsymbol{x}$, that is,

$$\hat{T} = \underset{T \in \text{valid}\ (\boldsymbol{x})}{\arg\max}\ s(\boldsymbol{x}, T) \tag{1}$$

where $s(\boldsymbol{x}, T) \in \mathbb{R}$ denotes a real-valued score of a tree T, and valid $(\boldsymbol{x})$ represents a set of all valid trees for $\boldsymbol{x}$. We assume that $\boldsymbol{x}$ has already been manually segmented into a sequence of EDUs: $\boldsymbol{x} = x_0, \ldots, x_{n-1}$.

Inspired by the success of recent span-based constituency parsers (Stern et al., 2017; Gaddy et al., 2018), we define the tree scores as the sum of *constituent scores* over internal nodes, that is,

$$s(\boldsymbol{x}, T) = \sum_{(i,j) \in T} s(i, j). \tag{2}$$

Thus, our parsing model consists of a single scoring function $s(i, j)$ that computes a constituent score of a contiguous text span $x_{i:j} = x_i, \ldots, x_j$, or simply (i, j). The higher the value of $s(i, j)$, the more likely that $x_{i:j}$ is a discourse constituent.

We show our parsing model in Figure 2. Our implementation of $s(i, j)$ can be decomposed into three modules: EDU-level feature extraction, span-level feature extraction, and span scoring. We discuss each of these in turn. Later, we also explain the decoding algorithm that we use to find the globally best-scoring tree.

Feature Extraction and Scoring

Inspired by existing RST parsers (Ji and Eisenstein, 2014; Li et al., 2014; Joty et al., 2015), we first encode the beginning and end words of an EDU:

$$\boldsymbol{v}_i^{\text{bw}} = \text{Embed}_{\text{w}}(b_w), \tag{3}$$
$$\boldsymbol{v}_i^{\text{ew}} = \text{Embed}_{\text{w}}(e_w), \tag{4}$$

where b_w and e_w denote the beginning and end words of the i-th EDU, and Embed $_{\text{w}}$ is a function that returns a parameterized embedding of the input word.

We also encode the POS tags corresponding to b_w and e_w as follows:

$$\boldsymbol{v}_i^{\text{bp}} = \text{Embed}_{\text{p}}(b_p), \tag{5}$$
$$\boldsymbol{v}_i^{\text{ep}} = \text{Embed}_{\text{p}}(e_p), \tag{6}$$

where Embed $_{\text{p}}$ is an embedding function for POS tags.

Prior work (Sagae, 2009; Ji and Eisenstein, 2014) has shown that syntactic cues can accelerate discourse parsing performance. We therefore extract syntactic features from each EDU. We apply a (syntactic) dependency parser to each sentence in the input text,[3] and then choose a head word for each EDU. A head word is a token whose parent in the dependency graph is ROOT or is not within the EDU.[4] We also extract the POS tag and the dependency label corresponding to the head word. A dependency label is a relation between a head word and its parent.

To sum up, we now have triplets of head information, $\{(h_w, h_p, h_r)_i\}_{i=0}^{n-1}$, each denoting the head word, the head POS, and the head relation of the i-th EDU, respectively. We embed these symbols using look-up tables:

$$\boldsymbol{v}_i^{\text{hw}} = \text{Embed}_{\text{w}}(h_w), \tag{7}$$
$$\boldsymbol{v}_i^{\text{hp}} = \text{Embed}_{\text{p}}(h_p), \tag{8}$$
$$\boldsymbol{v}_i^{\text{hr}} = \text{Embed}_{\text{r}}(h_r), \tag{9}$$

where Embed$_{\text{r}}$ is an embedding function for dependency relations.

Finally, we concatenate these embeddings:

$$\boldsymbol{v}_i' = [\boldsymbol{v}_i^{\text{bw}}; \boldsymbol{v}_i^{\text{ew}}; \boldsymbol{v}_i^{\text{bp}}; \boldsymbol{v}_i^{\text{ep}}; \boldsymbol{v}_i^{\text{hw}}; \boldsymbol{v}_i^{\text{hp}}; \boldsymbol{v}_i^{\text{hr}}], \tag{10}$$

and then transform it using a linear projection and Rectified Linear Unit (ReLU) activation function:

$$\boldsymbol{v}_i = \text{ReLU}\ (\boldsymbol{W}\boldsymbol{v}_i' + \boldsymbol{b}). \tag{11}$$

In the following, we use $\{\boldsymbol{v}_i\}_{i=0}^{n-1}$ as the feature vectors for the EDUs, $\{x_i\}_{i=0}^{n-1}$.

[3] We apply the Stanford CoreNLP parser (Manning et al., 2014) to the concatenation of the EDUs; `https://stanfordnlp.github.io/CoreNLP/`.

[4] If there are multiple head words in an EDU, we choose the left most one.

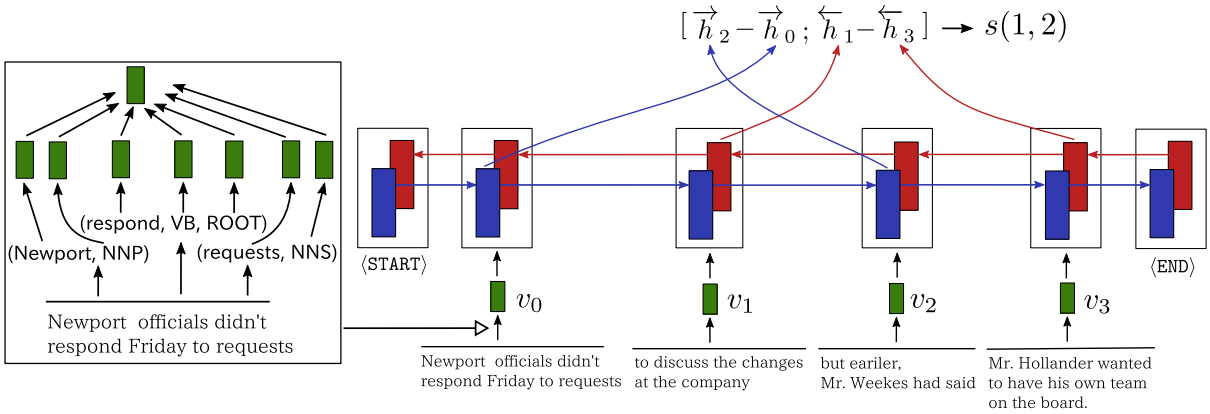


Figure 2: Our span-based discourse parsing model. We first encode each EDU based on the beginning and ending words and POS tags using embeddings. We also embed head information of each EDU. We then run a bidirectional LSTM and concatenate the span differences. The resulting vector is used to predict the constituent score of the text span (i, j). This figure illustrates the process for the span $(1, 2)$.

Following the span-based parsing models developed in the syntax domain (Stern et al., 2017; Gaddy et al., 2018), we then run a bidirectional Long Short-Term Memory (LSTM) over the sequence of EDU representations, $\{\boldsymbol{v}_i\}_{i=0}^{n-1}$, resulting in forward and backward representations for each step i $(0 \leq i \leq n-1)$:

$$\overrightarrow{\boldsymbol{h}}_0, \ldots, \overrightarrow{\boldsymbol{h}}_{n-1} = \overrightarrow{\text{LSTM}}(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1}), \quad (12)$$

$$\overleftarrow{\boldsymbol{h}}_0, \ldots, \overleftarrow{\boldsymbol{h}}_{n-1} = \overleftarrow{\text{LSTM}}(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1}). \quad (13)$$

We then compute a feature vector for a span (i, j) by concatenating the forward and backward span differences:

$$\boldsymbol{h}_{i,j} = [\overrightarrow{\boldsymbol{h}}_j - \overrightarrow{\boldsymbol{h}}_{i-1}; \overleftarrow{\boldsymbol{h}}_i - \overleftarrow{\boldsymbol{h}}_{j+1}]. \quad (14)$$

The feature vector, $\boldsymbol{h}_{i,j}$, is assumed to represent the content of the contiguous text span $x_{i:j}$ along with contextual information captured by the LSTMs.[5]

We did not use any feature templates because we found that they did not improve parsing performance in our unsupervised setting, though we observed that template features roughly following Joty et al. (2015) improved performance in a supervised setting.

Finally, given a span-level feature vector, $\boldsymbol{h}_{i,j}$, we use two-layer perceptrons with the ReLU activation function:

$$s(i, j) = \text{MLP}(\boldsymbol{h}_{i,j}), \quad (15)$$

which computes the constituent score of the contiguous text span $x_{i:j}$.

[5] A detailed investigation of the span-based parsing model using LSTM can be found in Gaddy et al. (2018).

Decoding

We use a Cocke-Kasami-Younger (CKY)-style dynamic programming algorithm to perform a global search over the space of valid trees and find the highest-scoring tree. For a document with n EDUs, we use an $n \times n$ table C, the cell $C[i, j]$ of which stores the subtree score spanning from i to j. For spans of length one (i.e., $i = j$), we assign constant scalar values:

$$\text{C}[i, i] = 1. \quad (16)$$

For general spans $0 \leq i < j \leq n-1$, we define the following recursion:

$$\text{C}[i, j] = s(i, j) + \max_{i \leq k < j} \text{C}[i, k] + \text{C}[k+1, j], \quad (17)$$

where $s(i, j)$ denotes the constituent score computed by our model.

To parse the full document, we first compute $\text{C}[0, n-1]$ in a bottom–up manner and then recursively trace the history of the selected split positions, k, resulting in a binary tree spanning the entire document.

3.2 Unsupervised Learning Using Viterbi EM

In this paper, we use Viterbi EM (Brown et al., 1993; Spitkovsky et al., 2010), a variant of the EM algorithm and *self-training* (McClosky et al., 2006a,b), to train the span-based discourse constituency parser (Section 3.1) in an unsupervised manner. Viterbi EM has suitable properties for discourse processing, as described later in this section.

Overall Procedure

We first automatically sample initial trees based on our prior knowledge of document structures (described later in Section 3.3) and then perform the M step on the sampled trees to initialize the model parameters. After the initialization step, we repeat the E step and the M step in turns. To perform early stopping, we use a held-out development set of 30 documents with annotated trees $\mathcal{T}^*_{\text{dev}}$, which are never used as the supervision to estimate the parsing model.

E Step

In the E step of Viterbi EM, based on the current model, we perform discourse constituency parsing for whole training documents $\mathcal{X}$, resulting in a pseudo treebank with discourse constituent structures, i.e.,

$$\mathcal{D} = \{(\boldsymbol{x}, \hat{T}) \mid \boldsymbol{x} \in \mathcal{X}, \hat{T} = \underset{T \in \text{valid } (\boldsymbol{x})}{\arg\max}\ s(\boldsymbol{x}, T)\} \tag{18}$$

where valid $(\boldsymbol{x})$ denotes a set of all valid trees for $\boldsymbol{x}$, $s(\boldsymbol{x}, T)$ is defined in Equation (2), and $\hat{T}$ is the highest-scoring parse tree based on the current model.

Klein and Manning (2001b) and Spitkovsky et al. (2010) count grammatical patterns used to derive syntactic trees in $\mathcal{D}$, which are then normalized and converted to probabilistic grammars in the next M step.

In contrast, ''discourse'' constituents are significantly sparse and tend to appear only once, which implies that it is almost meaningless to explicitly count discourse constituent patterns symbolically. We therefore attempt to directly use the trees in $\mathcal{D}$ to update the model parameters in the next M step.

M Step

In the M step, we re-estimate the next model as if it is supervised by the best parse trees found in the previous E step.

More precisely, we update the model parameters so that the next model satisfies the following constraints:

$$s(\boldsymbol{x}, \hat{T}) \geq s(\boldsymbol{x}, T') + \Delta(\hat{T}, T'), \tag{19}$$

for each instance $(\boldsymbol{x}, \hat{T}) \in \mathcal{D}$, where T' ranges over all valid trees. $\Delta(\hat{T}, T')$ is a tree distance we define as follows:

$$\Delta(\hat{T}, T') = |\hat{T}| - |\hat{T} \cap T'|, \tag{20}$$

where $|T|$ denotes the number of constituent spans (or internal nodes) in T, and $|\hat{T} \cap T'|$ represents the number of spans shared between $\hat{T}$ and T'. In other words, we hope that the score of the best parse tree $\hat{T}$ should be larger than that of the less-probable tree T' by at least the margin $\Delta(\hat{T}, T')$. Please note that $|\hat{T}| = |T'|$ always holds, because the parse tree $\hat{T}$ and the negative-sample tree T' are binary trees. $\Delta(\hat{T}, T') = 0$ holds if, and only if, $\hat{T} = T'$.

These constraints can be rewritten by using the margin-based criterion as follows:

$$\max\ \left(0, \max_{T'} \left[s(\boldsymbol{x}, T') + \Delta(\hat{T}, T')\right] - s(\boldsymbol{x}, \hat{T})\right).$$

We minimize this criterion by using the mini-batch stochastic gradient descent and the back-propagation algorithm.

The highest-scoring negative tree T' $(\neq \hat{T})$ can be efficiently found by modifying the dynamic programming algorithm in Equation (17). In particular, we replace $s(i, j)$ with $s(i, j) + \mathbb{1}[(i, j) \notin \hat{T}]$.

Combining Viterbi training and the margin-based objective function allows us to (1) avoid explicitly counting discourse constituent patterns as symbolic variables and (2) directly use the scores of the trees found in the E step for re-estimation of the next model.

3.3 Initialization in EM

In general, the EM algorithm tends to get stuck in local optima of the objective function (Charniak, 1993). Therefore, proper initialization is vital in order to avoid trivial solutions. This phenomenon has also been observed in EM-based grammar induction (Klein and Manning, 2004; Gimpel and Smith, 2012).

In this section, we introduce the initialization methods we use in Viterbi EM. More precisely, given an input document (i.e., a sequence of EDUs), we automatically build a discourse constituent structure based on our general prior knowledge of document structures. Below, we describe the four pieces of prior knowledge we use for the initial-tree sampling.

(i) Document Hierarchy

It is intuitively reasonable to consider that (elementary) discourse units belonging to the same textual chunk (e.g., sentence, paragraph) tend to form a subtree before crossing over the chunk

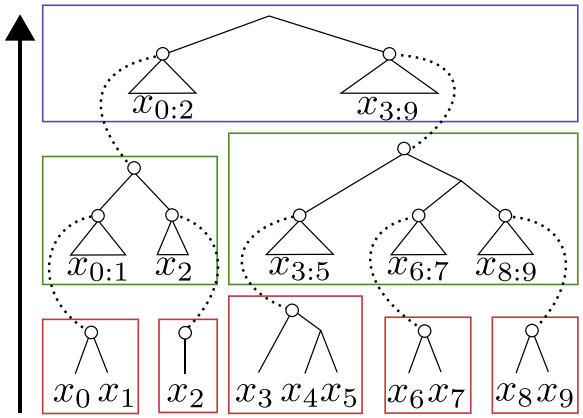


Figure 3: We build a discourse constituent structure incrementally in a bottom-up manner. Sentence-level subtrees are shown in red rectangles, paragraph-level subtrees in green rectangles, and the document-level tree in a blue rectangle.

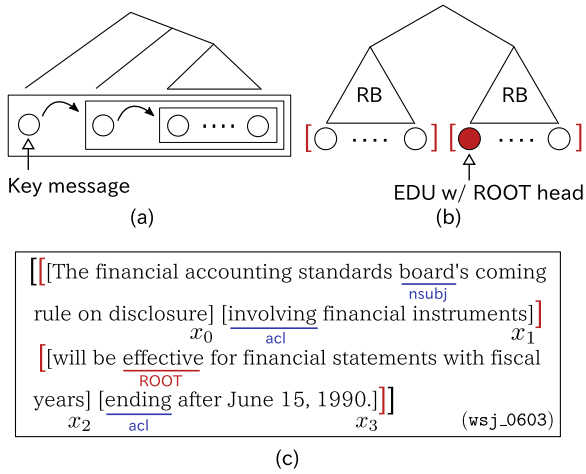


Figure 4: (a) We assume that an important text element tends to appear at earlier positions in the text, and the text following it complements the message, which leads to the right-heavy structure. (b)-(c) We split a intra-sentential EDU sequence into two subsequences based on the location of the EDU with the ROOT word. We build right-branching trees for each subsequence individually and finally bracket them. Head words are underlined.

boundaries. For example, we can assume that EDUs in the same sentence are preferentially connected with each other before getting combined with EDUs in other sentences. Actually, Joty et al. (2013, 2015) and Feng and Hirst (2014) observed that it is effective to incorporate inter-sentential and multi-sentential parsing to build a document-level tree.

First, we split an input document into sentence-level and paragraph-level segments by detecting sentence and paragraph boundaries, respectively. We obtain sentence segmentation by applying the Stanford CoreNLP (Manning et al., 2014) to the concatenation of EDUs. We also extract paragraph boundaries by detecting empty lines in the raw documents.[6] We then build a discourse constituent structure incrementally from sentence-level subtrees to paragraph-level subtrees and then to the document-level tree in a bottom-up manner. Figure 3 shows this process.

(ii) Discourse Branching Tendency

The second prior knowledge relates to information order in discourses and the branching tendencies of discourse trees. In general, an important text element tends to appear at earlier positions in the document, and then the text following it complements the message, which is reflected in the Right Frontier Constraint (Polanyi, 1985) in Segmented Discourse Representation Theory (Asher and Lascarides, 2003). This tendency can be assumed to hold recursively. Therefore, it is reasonable to consider that discourse structures tend to form right-heavy trees, as shown in Figure 4(a). Based on this assumption, we build right-branching trees for sentence-level, paragraph-level, and document-level discourse structures in the initial-tree sampling.

(iii) Syntax-Aware Branching Tendency

As already discussed, this work assumes that discourse structures tend to form right-heavy trees. However, in our preliminary experiments, we found that this naive assumption produces about 44% erroneous trees for sentence-level structures with 3 EDUs. For sentences with 4 EDUs, the error rate increases to about 70%, which is a non-negligible number in the initialization step.

To resolve this problem, we introduce another, more fine-grained, knowledge concept for sentence-level discourse structures. We expect that sentence-level trees are more strongly affected by syntactic cues (e.g., dependency graphs) than paragraph-level or document-level trees. More specifically, given an EDU sequence of one sentence, $x_i, \cdots, x_j$, we focus on a position of the EDU x_k with a head word that is in a ROOT relation with its parent in the dependency graph. We

[6]Therefore, our ‘‘paragraph’’ boundaries do not strictly correspond to paragraph segmentation. However, we found that this pseudo ‘‘paragraph’’ segmentation improves the parsing accuracy. We used the raw WSJ files (‘‘*.out’’) in RST-DT, e.g., ‘‘wsj_1135.out.’’

assume that the sub-sequence after the ROOT EDU, $x_{k:j}$, roughly corresponds to the predicate of the sentence, and the sub-sequence before the ROOT EDU, $x_{i:k-1}$, corresponds to the subject. We build right-branching trees for each sub-sequence individually and finally bracket them. We illustrate the procedure in Figure 4(b)-(c).

(iv) Locality Bias

Inspired by Smith and Eisner (2006), we introduce a structural *locality bias* as the last prior knowledge. The locality bias was observed to improve the accuracy of dependency grammar induction. We hypothesize that discourse constituents of shorter spans are preferable to those of longer ones.

Instead of introducing the locality bias into the initial-tree sampling, we encode it into the decoding algorithm in training and inference. More precisely, we re-write the CKY recursion in Equation (17) as follows:

$$\mathrm{C}[i,j] = s(i,j) + \frac{\lambda}{|i-j+1|} + \max_{i \le k < j} \mathrm{C}[i,k] + \mathrm{C}[k+1,j], \quad (21)$$

where λ denotes the hyperparameter and we empirically set $\lambda = 10$. The second term decreases in inverse proportion to the span distance.

4 Experiment Setup

4.1 Data

We use the RST Discourse Treebank (RST-DT) built by Carlson et al. (2001),[7] which consists of 385 *Wall Street Journal* articles manually annotated with RST structures (Mann and Thompson, 1988). We use the predefined split of 347 training articles and 38 test articles. We also prepare a development set with 30 instances randomly sampled from the training set, which is used only for hyper-parameter tuning and early stopping.

We tokenized the documents using Stanford CoreNLP tokenizer and converted them to lowercase. We also replaced digits with ‘‘7’’ (e.g., ‘‘12.34’’ $\rightarrow$ ‘‘77.77’’) to reduce data sparsity. We also replaced out-of-vocabulary tokens with special symbols ‘‘⟨ UNK ⟩.’’

[7] https://catalog.ldc.upenn.edu/LDC2002T07.

4.2 Metrics

Following existing studies in unsupervised syntactic parsing (Klein, 2005; Smith, 2006), we quantitatively evaluate unsupervised parsers by comparing parse trees with the manually annotated ones. We use the standard (unlabeled) constituency metrics in PARSEVAL: Unlabeled Precision (UP), Unlabeled Recall (UR), and their Micro F_1, which can indicate how well the parser identifies the linguistically reasonable structures.

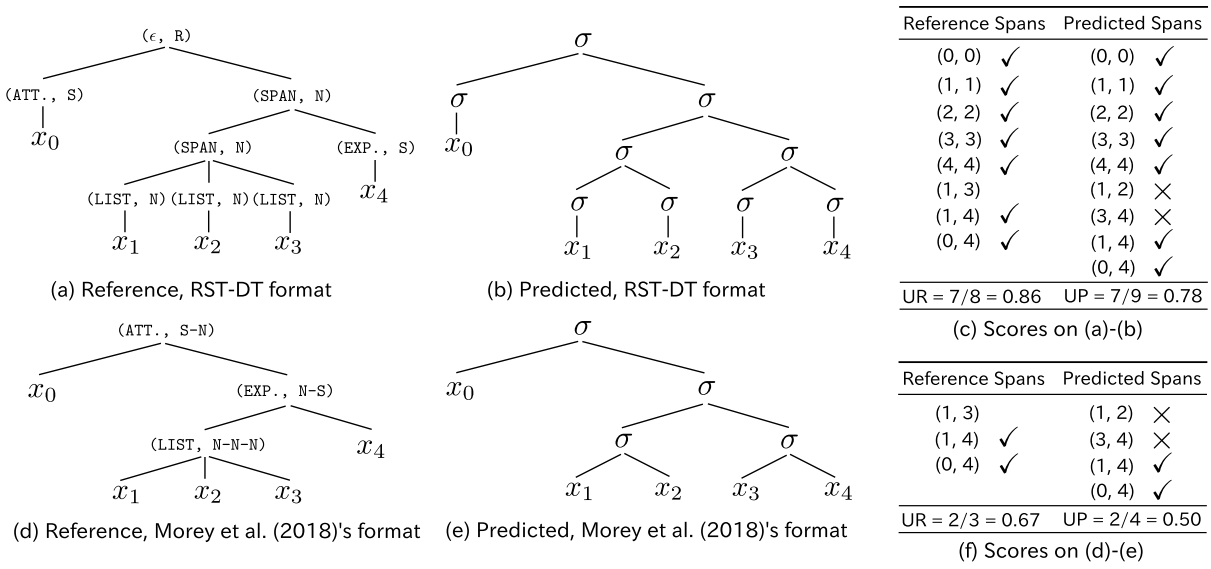
The traditional evaluation procedure for RST parsing is RST-PARSEVAL (Marcu, 2000), which adapts the PARSEVAL for the RST representation shown in Figure 5(a)-(b). However, Morey et al. (2018) showed that, as shown in Figure 5(c), traditional RST-PARSEVAL gives a higher-than-expected score because it considers pre-terminals (i.e., spans of length 1), which cannot be incorrect in the unlabeled constituency metrics. We therefore follow Morey et al. (2018) and perform the encoding of RST trees as shown in Figure 5(d)-(f). That is, we exclude spans of length 1 and include the root node. We also do not binarize the gold-standard trees.

4.3 Baselines

To quantitatively evaluate our unsupervised discourse constituency parser, it is necessary to develop strong baseline parsers. We thus propose *Combinational Incremental Parsers* (CIPs), which automatically and incrementally build a discourse (unlabeled) constituent structure from an EDU sequence based on the prior knowledge introduced in Section 3.3. That is, CIPs first build sentence-level discourse trees based on sentence segmentation using an *elementary parser* f_s. They then build paragraph-level trees using another elementary parser f_p, and finally output the document-level tree using f_d. An elementary parser is a function that returns a single tree given a sequence of EDUs or subtrees. CIPs can be represented as a triplet of elementary parsers, namely,

$$\langle f_s, f_p, f_d \rangle. \quad (22)$$

Inspired by earlier studies in unsupervised syntactic constituency parsing (Klein and Manning, 2001a,b; Klein, 2005; Seginer, 2007), we prepare



Reference Spans	Predicted Spans
(0, 0) ✓	(0, 0) ✓
(1, 1) ✓	(1, 1) ✓
(2, 2) ✓	(2, 2) ✓
(3, 3) ✓	(3, 3) ✓
(4, 4) ✓	(4, 4) ✓
(1, 3)	(1, 2) ×
(1, 4) ✓	(3, 4) ×
(0, 4) ✓	(1, 4) ✓
	(0, 4) ✓
UR = 7/8 = 0.86	UP = 7/9 = 0.78

(c) Scores on (a)-(b)

Reference Spans	Predicted Spans
(1, 3)	(1, 2) ×
(1, 4) ✓	(3, 4) ×
(0, 4) ✓	(1, 4) ✓
	(0, 4) ✓
UR = 2/3 = 0.67	UP = 2/4 = 0.50

(f) Scores on (d)-(e)

Figure 5: Variants of RST encodings and the corresponding unlabeled constituency scores: Unlabeled Recall (UR) and Unlabeled Precision (UP).

the following four candidates for the elementary parsers:

Right Branching (RB) Given a sequence of elements (i.e., EDUs or subtrees), RB always chooses the left-most element as a left terminal node and then treats the remaining elements as a right non-terminal (or terminal). This procedure is recursively applied to the remaining elements on the right, resulting in $(x_0\,(x_1\,(x_2\,\ldots)))$. As described in Section 3.3, we predict that RB somewhat captures the branching tendency of discourse informational structures. RB was also used as a strong baseline for unsupervised syntactic constituency parsing in Klein and Manning (2001b).

Left Branching (LB) Contrary to RB, LB always chooses the right-most element as the right terminal and then transforms the remaining elements on the left to a subtree, resulting in $(((\ldots\,x_{n-3})\,x_{n-2})\,x_{n-1})$.

Adaptive Right Branching (RB*) We augment RB by considering the syntax-aware branching tendency, described in Section 3.3(iii). That is, based on the position of the head EDU (with the ROOT relation), we split the sentence into two parts and then perform RB for each sub-sequence.

Random Bottom–Up (BU) BU randomly selects two adjacent elements and brackets them. This operation is repeated in a bottom–up manner until we obtain a single binary tree spanning the whole sequence.

4.4 Hyperparameters

We set the dimensionalities of the word embeddings, POS embeddings, relation embeddings, forward/backward LSTM hidden layers, and MLP to 300, 10, 10, 125, and 100, respectively. We initialized the word embeddings with the GloVe vectors trained on 840 billion tokens (Pennington et al., 2014). During the training, we did not fine-tune the word embeddings. We run the initialization steps for 3 epochs. We used a minibatch size of 10. We also used the Adam optimizer (Kingma and Ba, 2015).

5 Results and Discussion

In this section we report the results of the experiments and discuss them. We first discuss the comparison results of our method with baselines and the fully supervised RST parsers, including the results published in literature (Section 5.1). We then investigate the impact of initialization methods (Section 5.2). Finally, we provide our analysis on discourse constituents induced by our method (Section 5.3).

5.1 Performance Comparison

We compared our method with the baselines described in Section 4.3. We also included the previous work (Kobayashi et al., 2019) on unsupervised RST parsing as our baseline, though it is not a fair comparison because they use binarized

Method	UP	UR	Micro F_1
Unsupervised			
RB	7.5	7.7	7.6 (54.6)
$\langle \mathrm{RB}_s, \mathrm{RB}_d \rangle$	47.9	49.7	48.8 (74.8)
$\langle \mathrm{RB}_s, \mathrm{RB}_p, \mathrm{RB}_d \rangle$	57.9	60.2	59.0 (79.9)
LB	7.5	7.7	7.6 (54.6)
$\langle \mathrm{LB}_s, \mathrm{LB}_d \rangle$	41.7	43.3	42.5 (71.7)
$\langle \mathrm{LB}_s, \mathrm{LB}_p, \mathrm{LB}_d \rangle$	50.5	52.5	51.5 (76.2)
BU	19.2	19.9	19.5 (60.5)
$\langle \mathrm{BU}_s, \mathrm{BU}_d \rangle$	47.9	49.8	48.8 (74.9)
$\langle \mathrm{BU}_s, \mathrm{BU}_p, \mathrm{BU}_d \rangle$	54.5	56.6	55.5 (78.1)
$\langle \mathrm{RB}^*_s, \mathrm{RB}_p, \mathrm{RB}_d \rangle \cdots$ (a)	64.5	67.0	65.7 (83.2)
$\langle \mathrm{RB}^*_s, \mathrm{RB}_p, \mathrm{LB}_d \rangle \cdots$ (b)	65.6	68.1	66.8 (83.7)
Kobayashi et al. (2019)	–	–	– (80.8)
Ours, initialized by (a)	66.2	68.8	67.5 (84.0)
Ours, initialized by (b)	**66.8**	**69.4**	**68.0** (**84.3**)
Ours (b) + Aug.	**67.3**	**69.9**	**68.6** (**84.6**)
Supervised			
Ours, supervised	68.3	70.9	69.6 (85.1)
Feng and Hirst (2014)*	–	–	– (84.4)
Joty et al. (2015)*	–	–	– (82.5)
Human	–	–	– (88.7)

Table 1: Unlabeled constituency scores in the corrected RST-PARSEVAL (Morey et al., 2018) against non-binarized trees. UP and UR represent Unlabeled Precision and Unlabeled Recall, respectively. For reference, we also show the traditional RST-PARSEVAL Micro F_1 scores in parentheses. Asterisk indicates that we have borrowed the score from Morey et al. (2018).

golden trees for evaluation.[8] For reference, we also compared our method with fully supervised parsers: the supervised version of our model[9] and recent supervised parsers (Feng and Hirst, 2014; Joty et al., 2015) that incorporate intra-sentential and multi-sentential parsing as in our parser.

Table 1 shows the unlabeled constituency scores in the corrected RST-PARSEVAL (Morey et al., 2018) against non-binarized trees. We also show the traditional RST-PARSEVAL Micro F_1 scores in parentheses. $\langle f_s, f_d \rangle$ indicates that we used only sentence boundaries and discarded paragraph boundaries. The scores of external supervised parsers (Feng and Hirst, 2014; Joty et al., 2015) are borrowed from Morey et al. (2018).

[8]However, scores against the binarized trees and the original trees are quite similar (Morey et al., 2018).

[9]We used the same model and hyperparameters as the unsupervised model. The only difference is that we used conventional supervised learning with manually annotated trees in stead of Viterbi EM.

We observe that: (1) the incremental tree-construction approach with boundary information consistently improves parsing performances of the baselines; (2) RB-based CIPs are better than those with LB or BU; and (3) replacing RB with RB^* yields further improvements. These results confirm the reasonability of the prior knowledge of document structures. The best baseline is $\langle \mathrm{RB}^*_s, \mathrm{RB}_p, \mathrm{LB}_d \rangle$, which achieves a Micro F_1 score of 66.8% (83.7%) without any learning. Quite shockingly, the score is competitive with those of the supervised parsers.

Table 1 also demonstrates that our method outperforms all the baselines and achieves an F_1 score of 67.5% (84.0%). If we use the best baseline for initial-tree sampling in Viterbi EM, the performance further improves to 68.0% (84.3%).

To investigate the potential of our unsupervised parser, we also augmented the training dataset with an external unlabeled corpus. We used about 2,000 news articles from *Wall Street Journal* in Penn Treebank (Marcus et al., 1993) that are not shared with the RST-DT test set. We split the raw documents into EDU segmentations by using an external pre-trained EDU segmenter (Wang et al., 2018)[10] and found that the larger unlabeled dataset can improve parsing performance to 68.6%.

It is worth noting that our method outperforms the baselines used for the initialization, which implies that our method learns some knowledge of discourse constituentness in an unsupervised manner.

Our method also achieves comparable or superior results to supervised models. We suspect that the reason why the supervised version of our model outperforms the external supervised parsers (Feng and Hirst, 2014; Joty et al., 2015) is mostly dependent on feature extraction the introduction of paragraph boundaries.

5.2 Impact of Initialization Methods

Here, we evaluate the importance of initialization in Viterbi EM. Beginning with uniform initialization, we incrementally applied the initialization techniques introduced in Section 3.3 and investigated their impact on the results.

Table 2 shows the results. We observe that our model yields the lowest score of 58.9% with

[10]`https://github.com/PKU-TANGENT/NeuralEDUSeg`.

Knowledge	Initial Trees	Micro F_1
No (Uniform)	BU	58.9
(i)	$\langle \text{BU}_s, \text{BU}_p, \text{BU}_d \rangle$	59.1
(i)+(ii)	$\langle \text{RB}_s, \text{RB}_p, \text{RB}_d \rangle$	59.7
(i)+(ii)+(iii)	$\langle \text{RB}^*_s, \text{RB}_p, \text{RB}_d \rangle$	66.3
(i)+(ii)+(iii)+(iv)	$\langle \text{RB}^*_s, \text{RB}_p, \text{RB}_d \rangle$	67.5
Best baseline	$\langle \text{RB}^*_s, \text{RB}_p, \text{LB}_d \rangle$	68.0

Table 2: Comparison of initialization methods in our Viterbi training.

Relation	Ours	Supervised
ATTRIBUTION	90.7	92.7
ENABLEMENT	87.0	82.6
MANNER-MEANS	77.8	85.2
TEMPORAL	76.5	64.7
TOPIC-CHANGE	57.1	42.9
EXPLANATION	56.4	56.4
EVALUATION	56.3	55.0
SUMMARY	50.0	71.9
Total	69.9	70.9

Table 3: The best four and worst four rhetorical relations with their corresponding Unlabeled Recall scores. The relations are ordered according to scores of the unsupervised parser.

uniform initialization (no prior knowledge). By introducing Document Hierarchy in Section 3.3(i), parsing performance improves slightly to 59.1%. This result is interesting because the unlabeled constituency scores of BU and $\langle \text{BU}_s, \text{BU}_p, \text{BU}_d \rangle$ are quite different (19.5 vs. 55.5; see Table 1). We then introduced Discourse Branching Tendency in Section 3.3(ii) by replacing BU with RB in the CIP, which also improved the performance, slightly, to 59.7%. We then introduced Syntax-Aware Branching Tendency in Section 3.3(iii) by replacing RB with RB^* only for the sentence level, which brought a considerable performance gain of 6.6 points (66.3%). Finally, we introduced Locality Bias in Section 3.3(iv) and achieved 67.5%. We also found that our model can be improved further to 68.0% if we use the best baseline for initialization.

In total, these initialization techniques made a difference of 9.1 points compared with uniform initialization (i.e., 58.9 $\rightarrow$ 68.0), which implies that initialization should be carefully considered in unsupervised discourse (constituency) parsing using EM and that the prior knowledge we proposed in Section 3.3(i)-(iv) can capture some of the tendencies of document structures. We also found that Syntax-Aware Branching Tendency is most effective among the techniques, which suggests that more detailed knowledge can yield further improvements.

5.3 Learned Discourse Constituentness

Here, we further investigate the discourse constituentness learned by our method.

First, we calculated Unlabeled Recall (UR) scores for each relation class in RST-DT. We used 18 coarse-grained classes. Please note that we only focus on constituent spans $\{(i, j)\}$ because our method does not predict relation labels. Table 3 shows the results of the best four and the worst four relation classes of our method. We compare the results with the supervised version.

We observe that although our method uses an unsupervised approach and does not rely on structural annotations, some scores are comparable to those of the supervised version. We also found that relation classes with relatively higher scores can be assumed to form right-heavy structures (e.g., ATTRIBUTION, ENABLEMENT), whereas relations with lower scores can be considered to form left-heavy structures (e.g., EVALUATION, SUMMARY). These results are natural because the initialization methods we used in the Viterbi training strongly rely on RB-based CIP. This implies that, to capture discourse constituency phenomena of SUMMARY or EVALUTION relations, it is necessary to introduce other initialization techniques (or prior knowledge) in future.

Lastly, we qualitatively inspected the discourse constituentness learned by our method. We computed span scores $s(i, j)$ for all possible spans (i, j) in the RST-DT test set without using any boundary information. We then sampled text spans $x_{i:j}$ with relatively higher constituent scores, $s(i, j) > 10.0$.

As shown in the upper part of Table 4, we can observe that our method learns some aspects of discourse constituentness that seems linguistically reasonable. In particular, we found that our method has a potential to predict brackets for (1) clauses with connectives qualifying other clauses from right to left (e.g., ‘‘X [because B.]’’) and (2) attribution structures (e.g., ‘‘say that [B]’’). These results indicate that our method is good at identifying discourse constituents near the end

[The bankruptcy-court reorganization is being challenged ... by a dissident group of claimants] **[because it places a cap on the total amount of money available] [to settle claims.]** [It also bars future suits against ...] (11.74)
[The first two GAF trials were watched closely on Wall Street] **[because they were considered to be important tests of goverment's ability] [to convince a jury of allegations] [stemming from its insider-trading investigations.]** [In an eight-court indictment, the goverment charged GAF, ...] (10.16)
[The posters were sold for $1,300 to $6,000,] **[although the government says] [they had a value of only $53 to $200 apiece.]** [Henry Pitman, the assistant U.S. attorney] [handling the case,] [said] [about ...] (11.31)
[The office, an arm of the Treasury, said] [it doesn't have data on the financial position of applications] **[and thus can't determine] [why blacks are rejected more often.]** [Nevertheless, on Capital Hill,] [where ...] (11.57)
[After 93 hours of deliberation, the jurors in the second trial said] **[they were hopelessly deadlocked,] [and another mistrial was declared on March 22.]** [Meanwhile, a federal jury found Mr. Bilzerian ...] (11.66)
[(''I think I she knows me,] [but I'm not sure '')] [and Bridget Fonda, the actress] **[(''She knows me,] [but we're not really the best of friends'').]** [Mr. Revson, the gossip columnist, says] [there are people] [who ...] (11.11)
[its vice president ... resigned] [and its Houston work force has been trimmed by 40 people, of about 15%.] **[The maker of hand-held computers and computer systems said] [the personnel changes were needed]** [to improve the efficiency of its manufacturing operation.] [The company said] [it hasn't named a successor ...] (4.44)
[its vice president ... resigned] [and its Houston work force has been trimmed by 40 people, of about 15%.] [The maker of hand-held computers and computer systems said] **[the personnel changes were needed] [to improve the efficiency of its manufacturing operation.]** [The company said] [it hasn't named a successor...] (11.04)
[its vice president ... resigned] [and its Houston work force has been trimmed by 40 people, of about 15%.] [The maker of hand-held computers and computer systems said] [the personnel changes were needed] **[to improve the efficiency of its manufacturing operation.] [The company said]** [it hasn't named a successor...] (5.50)
[its vice president ... resigned] **[and its Houston work force has been trimmed by 40 people, of about 15%.] [The maker of hand-held computers and computer systems said] [the personnel changes were needed] [to improve the efficiency of its manufacturing operation.]** [The company said] [it hasn't named a successor...] (7.68)

Table 4: Discourse constituents and their predicted scores (in parentheses). We show the discourse constituents (in bold) in the RST-DT test set, which have relatively high span scores. We did NOT use any sentence/paragraph boundaries for scoring.

of sentences (or paragraphs), which is natural because RB is mainly used for generating initial trees in EM training. The bottom part of Table 4 demonstrates that the beginning position of the text span is also important to estimate constituenthood, along with the ending position.

6 Conclusion

In this paper, we introduced an unsupervised discourse constituency parsing algorithm that uses Viterbi EM with a margin-based criterion to train a span-based neural parser. We also introduced initialization methods for the Viterbi training of discourse constituents. We observed that our unsupervised parser achieves comparable or even superior performance to the baselines and fully supervised parsers. We also found that learned discourse constituents depend strongly on initialization used in Viterbi EM, and it is necessary to explore other initialization techniques to capture more diverse discourse phenomena.

We have two limitations in this study. First, this work focuses only on unlabeled discourse constituent structures. Although such hierarchical information is useful in downstream applications (Louis et al., 2010), both nuclearity statuses and rhetorical relations are also necessary for a more complete RST analysis. Second, our study uses only English documents for evaluation. However, different languages may have different structural regularities. Hence, it would be interesting to investigate whether the initialization methods are effective in different languages, which we believe gives suggestions on discourse-level universals. We leave these issues as a future work.

Acknowledgments

The research results have been achieved by ''Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation'', the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan. This work was also supported by JSPS KAKENHI grant number JP19K22861, JP18J12366.

References

Hiyan Alshawi. 1996. Head automata and bilingual tiling: Translation with minimal representations. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*.

Nicholas Asher and Alex Lascarides. 2003. *Logics and Conversation*, Cambridge University Press.

James K. Baker. 1979. Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustic Society of America*.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. In *Working Notes of the Workshop Statistically-based NLP Techniques*.

Eugene Charniak. 1993. *Statistical language learning*. MIT Press.

Yejin Choi and Claire Cardie. 2007. Structured local training and biased potential functions for conditional random fields with application to coreference resolution. In *Proceedings of the 2007 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

John DeNero and Dan Klein. 2008. The complexity of phrase alighment problems. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Vanessa Wei Feng and Graema Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

David Gaddy, Mitchell Stern, and Dan Klein. 2018. What's going on in neural constituency parsers? An analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Kevin Gimpel and Noah A. Smith. 2012. Concavity and initialization for unsupervised dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies.

Sharon Goldwater and Mark Johnson. 2005. Representation bias in unsupervised learning of syllable structure. In *Proceedings of the 9th Conference on Natural Language Learning*.

Dave Golland, John DeNero, and Jakob Uszkoreit. 2012. A feature-rich constituent context model for grammar induction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*, Longman.

Hugo Hernault, Helmut Prendinger, David A. DuVerle, and Mitsuru Ishizuka. 2010. HILDA: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3):1–33.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2016. Unsupervised neural dependency parsing. In *Proceedings of the 2016 Conference of Empirical Methods in Natural Language Processing*.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2018. Unsupervised grammar induction with depth-bounded pcfg. *Transactions of the Association for Computational Linguistics*, 6:211–224.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2015. CODRA a novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Shafiq Joty, Giuseppe Carenini, Raymond T. Ng, and Yashar Mehdad. 2013. Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference Learning Representations*.

Dan Klein. 2005. The unsupervised learning of natural language structure. Ph.D. Thesis, Stanford University

Dan Klein and Christopher D. Manning. 2001a. Distributional phrase structure induction. In *Proceedings of the 2001 Workshop on Computational Natural Language Learning*.

Dan Klein and Christopher D. Manning. 2001b. Natural language grammar induction using a constituent-context model. In *Advances in Neural Information Processing Systems*.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of constituency and dependency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Naoki Kobayashi, Tsutomu Hirao, Kengo Nakamura, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2019. Split of merge: Which is better for unsupervised RST parsing? In *Proceedings of the 2019 Conference of Empirical Methods in Natural Language Processing*.

Karim Lari and Steve J. Young. 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35–56.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *SIGDIAL'10*.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Towards a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Danial Marcu, Magdalena Romera, and Estibaliz Amorrortu. 1999. Experiments in constructing a corpus of discourse trees: Problems, annotation choices, issues. In *Proceedings of the ACL'99 Workshop on Standards and Tools for Discourse Tagging*.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*, MIT Press.

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

David McClosky, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*.

Eleni Miltsakaki and Karen Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10(1):25–55.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2017. How much progress have we made on rst discourse parsing? A replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2018. A dependency perspective on RST discourse parsing and evaluation. *Computational Linguistics*, 44(2):197–235.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Radford M. Neal and Geoffrey E. Hinton. 1998. *A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants*, Learning and Graphical Models.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Livia Polanyi. 1985. A theory of discourse structure and discourse coherence. In *Proceedings of the 21st Regional Meeting of the Chicago Linguistics Society*.

Livia Polanyi and Martin Van den Berg. 2011. Discourse structure and sentiment. In *2011 IEEE 11th International Conference on Data Mining Workshops*.

Andrew Radford. 1988. *Transformational Grammar*, Cambridge University Press.

Kenji Sagae. 2009. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Workshop on Parsing Technology*.

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Noah A. Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*.

Noah Ashton Smith. 2006. Novel estimation methods for unsupervised discovery of latent structure in natural language text. Ph.D. Thesis, Johns Hopkins University.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010. Viterbi training improves unsupervised

dependency parsing. In *Proceedings of the 14th Conference on Computational Natural Language Learning*.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural consituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.