# Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis

**Shafiq Joty**[*]
sjoty@qf.org.qa
Qatar Computing Research Institute
Qatar Foundation
Doha, Qatar

**Giuseppe Carenini, Raymond Ng, Yashar Mehdad**
{carenini, rng, mehdad}@cs.ubc.ca
Department of Computer Science
University of British Columbia
Vancouver, Canada

## Abstract

We propose a novel approach for developing a two-stage document-level discourse parser. Our parser builds a discourse tree by applying an optimal parsing algorithm to probabilities inferred from two Conditional Random Fields: one for intra-sentential parsing and the other for multi-sentential parsing. We present two approaches to combine these two stages of discourse parsing effectively. A set of empirical evaluations over two different datasets demonstrates that our discourse parser significantly outperforms the state-of-the-art, often by a wide margin.

## 1 Introduction

Discourse of any kind is not formed by independent and isolated textual units, but by related and structured units. Discourse analysis seeks to uncover such structures underneath the surface of the text, and has been shown to be beneficial for text summarization (Louis et al., 2010; Marcu, 2000b), sentence compression (Sporleder and Lapata, 2005), text generation (Prasad et al., 2005), sentiment analysis (Somasundaran, 2010) and question answering (Verberne et al., 2007).

Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), one of the most influential theories of discourse, represents texts by labeled hierarchical structures, called Discourse Trees (DTs), as exemplified by a sample DT in Figure 1. The leaves of a DT correspond to contiguous Elementary Discourse Units (EDUs) (six in the example). Adjacent EDUs are connected by rhetorical relations (e.g., *Elaboration*, *Contrast*), forming larger discourse units (represented by internal nodes), which in turn are also subject to this relation linking. Discourse units linked by a rhetorical relation are further distinguished based on their relative importance in the text: *nucleus* being the central part, whereas *satellite* being the peripheral one. Discourse analysis in RST involves two subtasks: *discourse segmentation* is the task of identifying the EDUs, and *discourse parsing* is the task of linking the discourse units into a labeled tree.

While recent advances in automatic discourse segmentation and sentence-level discourse parsing have attained accuracies close to human performance (Fisher and Roark, 2007; Joty et al., 2012), discourse parsing at the document-level still poses significant challenges (Feng and Hirst, 2012) and the performance of the existing document-level parsers (Hernault et al., 2010; Subba and Di-Eugenio, 2009) is still considerably inferior compared to human gold-standard. This paper aims to reduce this performance gap and take discourse parsing one step further. To this end, we address three key limitations of existing parsers as follows.

First, existing discourse parsers typically model the structure and the labels of a DT separately in a *pipeline* fashion, and also do not consider the sequential dependencies between the DT constituents, which has been recently shown to be critical (Feng and Hirst, 2012). To address this limitation, as the first contribution, we propose a novel document-level discourse parser based on probabilistic discriminative parsing models, represented as Conditional Random Fields (CRFs) (Sutton et al., 2007), to infer the probability of all possible DT constituents. The CRF models effectively represent the structure and the label of a DT constituent jointly, and whenever possible, capture the sequential dependencies between the constituents.

Second, existing parsers apply greedy and sub-optimal parsing algorithms to build the DT for a document. To cope with this limitation, our CRF models support a probabilistic bottom-up parsing

[*]This work was conducted at the University of British Columbia, Vancouver, Canada.

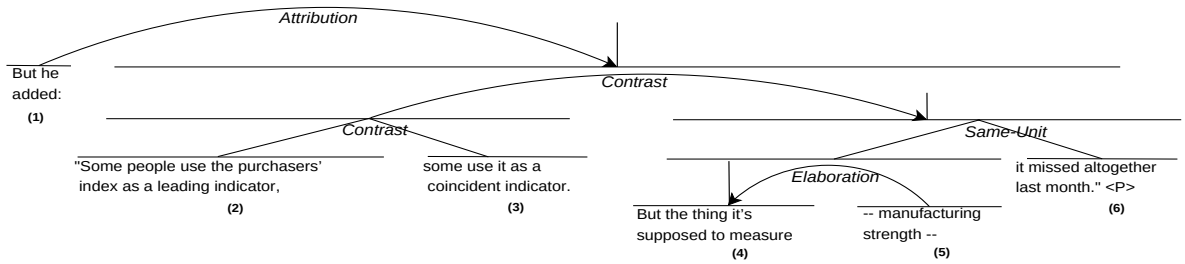


Figure 1: Discourse tree for two sentences in RST-DT. Each of the sentences contains three EDUs. The second sentence has a well-formed discourse tree, but the first sentence does not have one.

algorithm which is non-greedy and optimal.

Third, existing discourse parsers do not discriminate between intra-sentential (i.e., building the DTs for the individual sentences) and multi-sentential parsing (i.e., building the DT for the document). However, we argue that distinguishing between these two conditions can result in more effective parsing. Two separate parsing models could exploit the fact that rhetorical relations are distributed differently intra-sententially vs. multi-sententially. Also, they could independently choose their own informative features. As another key contribution of our work, we devise two different parsing components: one for intra-sentential parsing, the other for multi-sentential parsing. This provides for scalable, modular and flexible solutions, that can exploit the strong correlation observed between the text structure (sentence boundaries) and the structure of the DT.

In order to develop a complete and robust discourse parser, we combine our intra-sentential and multi-sentential parsers in two different ways. Since most sentences have a well-formed discourse sub-tree in the full document-level DT (for example, the second sentence in Figure 1), our first approach constructs a DT for every sentence using our intra-sentential parser, and then runs the multi-sentential parser on the resulting sentence-level DTs. However, this approach would disregard those cases where rhetorical structures violate sentence boundaries. For example, consider the first sentence in Figure 1. It does not have a well-formed sub-tree because the unit containing EDUs 2 and 3 merges with the next sentence and only then is the resulting unit merged with EDU 1. Our second approach, in an attempt of dealing with these cases, builds sentence-level sub-trees by applying the intra-sentential parser on a sliding window covering two adjacent sentences and by then consolidating the results produced by overlapping windows. After that, the multi-sentential parser takes all these sentence-level sub-trees and builds a full rhetorical parse for the document.

While previous approaches have been tested on only one corpus, we evaluate our approach on texts from two very different genres: news articles and instructional how-to-do manuals. The results demonstrate that our contributions provide consistent and statistically significant improvements over previous approaches. Our final result compares very favorably to the result of state-of-the-art models in document-level discourse parsing.

In the rest of the paper, after discussing related work in Section 2, we present our discourse parsing framework in Section 3. In Section 4, we describe the intra- and multi-sentential parsing components. Section 5 presents the two approaches to combine the two stages of parsing. The experiments and error analysis, followed by future directions are discussed in Section 6. Finally, we summarize our contributions in Section 7.

## 2 Related work

The idea of staging document-level discourse parsing on top of sentence-level discourse parsing was investigated in (Marcu, 2000a; LeThanh et al., 2004). These approaches mainly rely on discourse markers (or cues), and use hand-coded rules to build DTs for sentences first, then for paragraphs, and so on. However, often rhetorical relations are not explicitly signaled by discourse markers (Marcu and Echihabi, 2002), and discourse structures do not always correspond to paragraph structures (Sporleder and Lascarides, 2004). Therefore, rather than relying on hand-coded rules based on discourse markers, recent approaches employ supervised machine learning techniques with a large set of informative features.

Hernault et al., (2010) presents the publicly available HILDA parser. Given the EDUs in a doc-

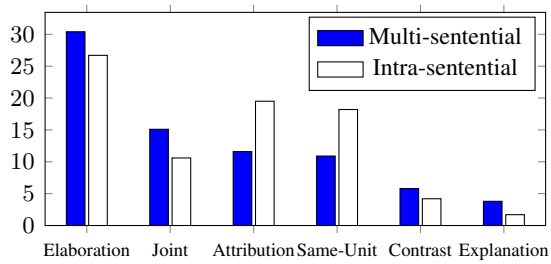


Figure 2: Distributions of six most frequent relations in intra-sentential and multi-sentential parsing scenarios.

ument, HILDA iteratively employs two Support Vector Machine (SVM) classifiers in pipeline to build the DT. In each iteration, a binary classifier first decides which of the adjacent units to merge, then a multi-class classifier connects the selected units with an appropriate relation label. They evaluate their approach on the RST-DT corpus (Carlson et al., 2002) of news articles. On a different genre of instructional texts, Subba and Di-Eugenio (2009) propose a shift-reduce parser that relies on a classifier for relation labeling. Their classifier uses Inductive Logic Programming (ILP) to learn first-order logic rules from a set of features including *compositional semantics*. In this work, we address the limitations of these models (described in Section 1) introducing our novel discourse parser.

## 3 Our Discourse Parsing Framework

Given a document with sentences already segmented into EDUs, the discourse parsing problem is determining which discourse units (EDUs or larger units) to relate (i.e., the structure), and how to relate them (i.e., the labels or the discourse relations) in the resulting DT. Since we already have an accurate sentence-level discourse parser (Joty et al., 2012), a straightforward approach to document-level parsing could be to simply apply this parser to the whole document. However this strategy would be problematic because of scalability and modeling issues. Note that the number of valid trees grows exponentially with the number of EDUs in a document.[1] Therefore, an exhaustive search over the valid trees is often unfeasible, even for relatively small documents.

For modeling, the problem is two-fold. On the one hand, it appears that rhetorical relations are distributed differently intra-sententially vs. multi-sententially. For example, Figure 2 shows a comparison between the two distributions of six most

[1]For $n+1$ EDUs, the number of valid discourse trees is actually the *Catalan number* $C_n$.

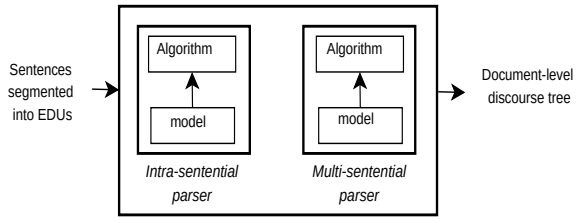


Figure 3: Discourse parsing framework.

frequent relations on a development set containing 20 randomly selected documents from RST-DT. Notice that relations *Attribution* and *Same-Unit* are more frequent than *Joint* in intra-sentential case, whereas *Joint* is more frequent than the other two in multi-sentential case. On the other hand, different kinds of features are applicable and informative for intra-sentential vs. multi-sentential parsing. For example, syntactic features like *dominance sets* (Soricut and Marcu, 2003) are extremely useful for sentence-level parsing, but are not even applicable in multi-sentential case. Likewise, *lexical chain features* (Sporleder and Lascarides, 2004), that are useful for multi-sentential parsing, are not applicable at the sentence level.

Based on these observations, our discourse parsing framework comprises two separate modules: an *intra-sentential parser* and a *multi-sentential parser* (Figure 3). First, the intra-sentential parser produces one or more discourse sub-trees for each sentence. Then, the multi-sentential parser generates a full DT for the document from these sub-trees. Both of our parsers have the same two components: a *parsing model* assigns a probability to every possible DT, and a *parsing algorithm* identifies the most probable DT among the candidate DTs in that scenario. While the two models are rather different, the same parsing algorithm is shared by the two modules. Staging multi-sentential parsing on top of intra-sentential parsing in this way allows us to exploit the strong correlation between the text structure and the DT structure as explained in detail in Section 5. Before describing our parsing models and the parsing algorithm, we introduce some terminology that we will use throughout the paper.

Following (Joty et al., 2012), a DT can be formally represented as a set of constituents of the form $R[i, m, j]$, referring to a rhetorical relation $R$ between the discourse unit containing EDUs $i$ through $m$ and the unit containing EDUs $m+1$ through $j$. For example, the DT for the second sentence in Figure 1 can be represented as

{*Elaboration-NS[4,4,5], Same-Unit-NN[4,5,6]*}. Notice that a relation $R$ also specifies the nuclearity statuses of the discourse units involved, which can be one of *Nucleus-Satellite (NS), Satellite-Nucleus (SN) and Nucleus-Nucleus (NN)*.

## 4 Parsing Models and Parsing Algorithm

The job of our intra-sentential and multi-sentential parsing models is to assign a probability to each of the constituents of all possible DTs at the sentence level and at the document level, respectively. Formally, given the model parameters $\Theta$, for each possible constituent $R[i, m, j]$ in a candidate DT at the sentence or document level, the parsing model estimates $P(R[i, m, j]|\Theta)$, which specifies a joint distribution over the label $R$ and the structure $[i, m, j]$ of the constituent.

### 4.1 Intra-Sentential Parsing Model

Recently, we proposed a novel parsing model for sentence-level discourse parsing (Joty et al., 2012), that outperforms previous approaches by effectively modeling sequential dependencies along with structure and labels jointly. Below we briefly describe the parsing model, and show how it is applied to obtain the probabilities of all possible DT constituents at the sentence level.

Figure 4 shows the intra-sentential parsing model expressed as a Dynamic Conditional Random Field (DCRF) (Sutton et al., 2007). The observed nodes $U_j$ in a sequence represent the discourse units (EDUs or larger units). The first layer of hidden nodes are the structure nodes, where $S_j \in \{0, 1\}$ denotes whether two adjacent discourse units $U_{j-1}$ and $U_j$ should be connected or not. The second layer of hidden nodes are the relation nodes, with $R_j \in \{1 \dots M\}$ denoting the relation between two adjacent units $U_{j-1}$ and $U_j$, where $M$ is the total number of relations in the relation set. The connections between adjacent nodes in a hidden layer encode sequential dependencies between the respective hidden nodes, and can enforce constraints such as the fact that a $S_j = 1$ must not follow a $S_{j-1} = 1$. The connections between the two hidden layers model the structure and the relation of a DT (sentence-level) constituent jointly.

To obtain the probability of the constituents of all candidate DTs for a sentence, we apply the parsing model recursively at different levels of the DT and compute the posterior marginals over the relation-structure pairs. To illustrate the process, let us assume that the sentence contains four EDUs. At the first (bottom) level, when all the units are the EDUs, there is only one possible unit sequence to which we apply our DCRF model (Figure 5(a)). We compute the posterior marginals $P(R_2, S_2{=}1|e_1, e_2, e_3, e_4, \Theta)$, $P(R_3, S_3{=}1|e_1, e_2, e_3, e_4, \Theta)$ and $P(R_4, S_4{=}1|e_1, e_2, e_3, e_4, \Theta)$ to obtain the probability of the constituents $R[1, 1, 2]$, $R[2, 2, 3]$ and $R[3, 3, 4]$, respectively. At the second level, there are three possible unit sequences $(e_{1:2}, e_3, e_4)$, $(e_1, e_{2:3}, e_4)$ and $(e_1, e_2, e_{3:4})$. Figure 5(b) shows their corresponding DCRFs. The posterior marginals $P(R_3, S_3{=}1|e_{1:2}, e_3, e_4, \Theta)$, $P(R_{2:3}\ S_{2:3}{=}1|e_1, e_{2:3}, e_4, \Theta)$, $P(R_4, S_4{=}1|e_1, e_{2:3}, e_4, \Theta)$ and $P(R_{3:4}, S_{3:4}{=}1|e_1, e_2, e_{3:4}, \Theta)$ computed from the three sequences correspond to the probability of the constituents $R[1, 2, 3]$, $R[1, 1, 3]$, $R[2, 3, 4]$ and $R[2, 2, 4]$, respectively. Similarly, we attain the probability of the constituents $R[1, 1, 4]$, $R[1, 2, 4]$ and $R[1, 3, 4]$ by computing their respective posterior marginals from the three possible sequences at the third (top) level.

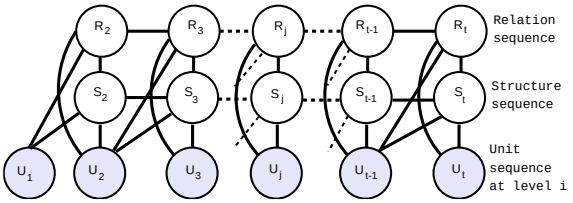


Figure 4: A chain-structured DCRF as our intra-sentential parsing model.

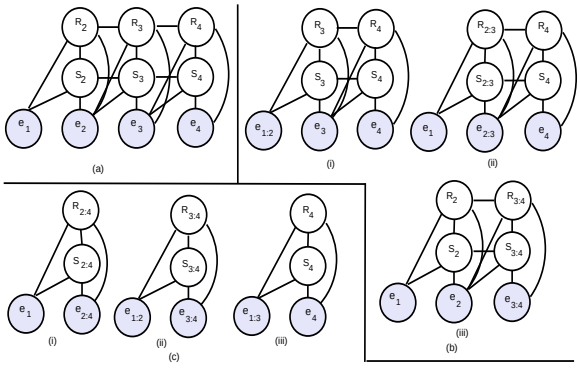

Figure 5: Our parsing model applied to the sequences at different levels of a sentence-level DT. (a) Only possible sequence at the first level, (b) Three possible sequences at the second level, (c) Three possible sequences at the third level.

At this point what is left to be explained is how we generate all possible sequences for a given number of EDUs in a sentence. Algorithm 1 demonstrates how we do that. More specifically, to compute the probabilities of each DT con-

stituent $R[i, k, j]$, we need to generate sequences like $(e_1, \cdots, e_{i-1}, e_{i:k}, e_{k+1:j}, e_{j+1}, \cdots, e_n)$ for $1 \leq i \leq k < j \leq n$. In doing so, we may generate some duplicate sequences. Clearly, the sequence $(e_1, \cdots, e_{i-1}, e_{i:i}, e_{i+1:j}, e_{j+1}, \cdots, e_n)$ for $1 \leq i \leq k < j < n$ is already considered for computing the probability of $R[i+1, j, j+1]$. Therefore, it is a duplicate sequence that we exclude from our list of all possible sequences.

**Input**: Sequence of EDUs: $(e_1, e_2, \cdots, e_n)$
**Output**: List of sequences: $L$
**for** $i = 1 \rightarrow n-1$ **do**
  **for** $j = i+1 \rightarrow n$ **do**
    **if** $j$ == $n$ **then**
      **for** $k = i \rightarrow j-1$ **do**
        $L$.append $((e_1, .., e_{i-1}, e_{i:k}, e_{k+1:j}, e_{j+1}, .., e_n))$
      **end**
    **else**
      **for** $k = i+1 \rightarrow j-1$ **do**
        $L$.append $((e_1, .., e_{i-1}, e_{i:k}, e_{k+1:j}, e_{j+1}, .., e_n))$
      **end**
    **end**
  **end**
**end**

**Algorithm 1:** Generating all possible sequences for a sentence with $n$ EDUs.

Once we obtain the probability of all possible DT constituents, the discourse sub-trees for the sentences are built by applying an optimal probabilistic parsing algorithm (Section 4.4) using one of the methods described in Section 5.

### 4.2 Multi-Sentential Parsing Model

Given the discourse units (sub-trees) for all the sentences of a document, a simple approach to build the rhetorical tree of the document would be to apply a new DCRF model, similar to the one in Figure 4 (with different parameters), to all the possible sequences generated from these units to infer the probability of all possible higher-order constituents. However, the number of possible sequences and their length increase with the number of sentences in a document. For example, assuming that each sentence has a well-formed DT, for a document with $n$ sentences, Algorithm 1 generates $O(n^3)$ sequences, where the sequence at the bottom level has $n$ units, each of the sequences at the second level has $n$-1 units, and so on. Since the model in Figure 4 has a "fat" chain structure,

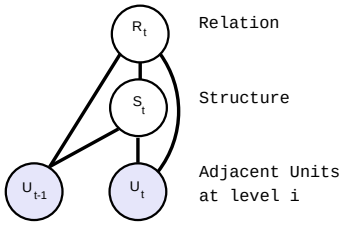


Figure 6: A CRF as a multi-sentential parsing model.

we could use forwards-backwards algorithm for exact inference in this model (Sutton and McCallum, 2012). However, forwards-backwards on a sequence containing $T$ units costs $O(TM^2)$ time, where $M$ is the number of relations in our relation set. This makes the chain-structured DCRF model impractical for multi-sentential parsing of long documents, since learning requires to run inference on every training sequence with an overall time complexity of $O(TM^2n^3)$ per document.

Our model for multi-sentential parsing is shown in Figure 6. The two observed nodes $U_{t-1}$ and $U_t$ are two adjacent discourse units. The (hidden) structure node $S \in \{0, 1\}$ denotes whether the two units should be connected or not. The hidden node $R \in \{1 \ldots M\}$ represents the relation between the two units. Notice that like the previous model, this is also an undirected graphical model. It becomes a CRF if we directly model the hidden (output) variables by conditioning its clique potential (or factor) $\phi$ on the observed (input) variables:

$$P(R_t, S_t|\mathbf{x}, \Theta) = \frac{1}{Z(\mathbf{x}, \Theta)} \phi(R_t, S_t|\mathbf{x}, \Theta) \quad (1)$$

where $\mathbf{x}$ represents input features extracted from the observed variables $U_{t-1}$ and $U_t$, and $Z(\mathbf{x}, \Theta)$ is the partition function. We use a log-linear representation of the factor:

$$\phi(R_t, S_t|\mathbf{x}, \Theta) = \exp(\Theta^T \ f(R_t, S_t, \mathbf{x})) \quad (2)$$

where $f(R_t, S_t, \mathbf{x})$ is a feature vector derived from the input features $\mathbf{x}$ and the labels $R_t$ and $S_t$, and $\Theta$ is the corresponding weight vector. Although, this model is similar in spirit to the model in Figure 4, we now break the chain structure, which makes the inference much faster (i.e., complexity of $O(M^2)$). Breaking the chain structure also allows us to balance the data for training (equal number instances with $S$=1 and $S$=0), which dramatically reduces the learning time of the model.

We apply our model to all possible adjacent units at all levels for the multi-sentential case, and

compute the posterior marginals of the relation-structure pairs $P(R_t, S_t{=}1|U_{t-1}, U_t, \Theta)$ to obtain the probability of all possible DT constituents.

### 4.3 Features Used in our Parsing Models

Table 1 summarizes the features used in our parsing models, which are extracted from two adjacent units $U_{t-1}$ and $U_t$. Since most of these features are adopted from previous studies (Joty et al., 2012; Hernault et al., 2010), we briefly describe them.

**Organizational** features include the *length* of the units as the number of EDUs and tokens. It also includes the *distances* of the units from the beginning and end of the sentence (or text in the multi-sentential case). **Text structural** features indirectly capture the correlation between text structure and rhetorical structure by counting the number of *sentence* and *paragraph* boundaries in the units. Discourse markers (e.g., *because, although*) carry informative clues for rhetorical relations (Marcu, 2000a). Rather than using a fixed list of discourse markers, we use an empirically learned *lexical* **N-gram** dictionary following (Joty et al., 2012). This approach has been shown to be more robust and flexible across domains (Biran and Rambow, 2011; Hernault et al., 2010). We also include part-of-speech (*POS*) tags for the beginning and end $N$ tokens in a unit.

| **8 Organizational features** | ***Intra & Multi-Sentential*** |
|---|---|
| Number of EDUs in *unit 1* (or *unit 2*). | |
| Number of tokens in *unit 1* (or *unit 2*). | |
| Distance of unit 1 in EDUs to the *beginning* (or to the *end*). | |
| Distance of unit 2 in EDUs to the *beginning* (or to the *end*). | |
| **4 Text structural features** | ***Multi-Sentential*** |
| Number of sentences in *unit 1* (or *unit 2*). | |
| Number of paragraphs in *unit 1* (or *unit 2*). | |
| **8 N-gram features** $N{\in}\{1,2,3\}$ | ***Intra & Multi-Sentential*** |
| *Beginning* (or *end*) lexical N-grams in unit 1. | |
| *Beginning* (or *end*) lexical N-grams in unit 2. | |
| *Beginning* (or *end*) POS N-grams in unit 1. | |
| *Beginning* (or *end*) POS N-grams in unit 2. | |
| **5 Dominance set features** | ***Intra-Sentential*** |
| Syntactic labels of the *head* node and the *attachment* node. | |
| Lexical heads of the *head* node and the *attachment* node. | |
| *Dominance relationship* between the two units. | |
| **8 Lexical chain features** | ***Multi-Sentential*** |
| Number of chains start in unit 1 and end in unit 2. | |
| Number of chains *start* (or *end*) in *unit 1* (or in *unit 2*). | |
| Number of chains skipping both unit 1 and unit 2. | |
| Number of chains skipping *unit 1* (or *unit 2*). | |
| **2 Contextual features** | ***Intra & Multi-Sentential*** |
| *Previous* and *next* feature vectors. | |
| **2 Substructure features** | ***Intra & Multi-Sentential*** |
| Root nodes of the *left* and *right* rhetorical sub-trees. | |

Table 1: Features used in our parsing models.

Lexico-syntactic features **dominance sets** (Soricut and Marcu, 2003) are very effective for intra-sentential parsing. We include *syntactic labels* and *lexical heads* of head and attachment nodes along with their *dominance relationship* as features. **Lexical chains** (Morris and Hirst, 1991) are sequences of semantically related words that can indicate topic shifts. Features extracted from lexical chains have been shown to be useful for finding paragraph-level discourse structure (Sporleder and Lascarides, 2004). We compute lexical chains for a document following the approach proposed in (Galley and McKeown, 2003), that extracts lexical chains after performing word sense disambiguation. Following (Joty et al., 2012), we also encode *contextual* and *rhetorical sub-structure* features in our models. The rhetorical sub-structure features incorporate hierarchical dependencies between DT constituents.

### 4.4 Parsing Algorithm

Given the probability of all possible DT constituents in the intra-sentential and multi-sentential scenarios, the job of the parsing algorithm is to find the most probable DT for that scenario. Following (Joty et al., 2012), we implement a probabilistic CKY-like bottom-up algorithm for computing the most likely parse using dynamic programming. Specifically, with $n$ discourse units, we use the upper-triangular portion of the $n{\times}n$ dynamic programming table $D$. Given $U_x(0)$ and $U_x(1)$ are the start and end EDU Ids of unit $U_x$:

$$D[i,j] = P(R[U_i(0), U_k(1), U_j(1)]) \qquad (3)$$

where, $k = \underset{i \le p \le j}{\operatorname{argmax}}\, P(R[U_i(0), U_p(1), U_j(1)])$.

Note that, in contrast to previous studies on document-level parsing (Hernault et al., 2010; Subba and Di-Eugenio, 2009; Marcu, 2000b), which use a greedy algorithm, our approach finds a discourse tree that is globally optimal.

## 5 Document-level Parsing Approaches

Now that we have presented our intra-sentential and our multi-sentential parsers, we are ready to describe how they can be effectively combined to perform document-level discourse analysis. Recall that a key motivation for a two-stage parsing is that it allows us to capture the correlation between text structure and discourse structure in a scalable, modular and flexible way. Below we describe two different approaches to model this correlation.

### 5.1 1S-1S (1 Sentence-1 Sub-tree)

A key finding from several previous studies on sentence-level discourse analysis is that most sentences have a well-formed discourse sub-tree in the full document-level DT (Joty et al., 2012; Fisher and Roark, 2007). For example, Figure 7(a) shows 10 EDUs in 3 sentences (see boxes), where the DTs for the sentences obey their respective sentence boundaries. The 1S-1S approach aims to maximally exploit this finding. It first constructs a DT for every sentence using our intra-sentential parser, and then it provides our multi-sentential parser with the sentence-level DTs to build the rhetorical parse for the whole document.

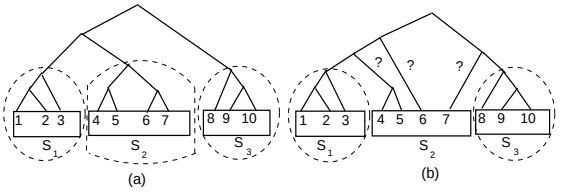


Figure 7: Two possible DTs for three sentences.

### 5.2 Sliding Window

While the assumption made by 1S-1S clearly simplifies the parsing process, it totally ignores the cases where discourse structures violate sentence boundaries. For example, in the DT shown in Figure 7(b), sentence $S_2$ does not have a well-formed sub-tree because some of its units attach to the left (4-5, 6) and some to the right (7). Vliet and Redeker (2011) call these cases as 'leaky' boundaries. Even though less than 5% of the sentences have leaky boundaries in RST-DT, in other corpora this can be true for a larger portion of the sentences. For example, we observe over 12% sentences with leaky boundaries in the Instructional corpus of (Subba and Di-Eugenio, 2009). However, we notice that in most cases where discourse structures violate sentence boundaries, its units are merged with the units of its adjacent sentences, as in Figure 7(b). For example, this is true for 75% cases in our development set containing 20 news articles from RST-DT and for 79% cases in our development set containing 20 how-to-do manuals from the Instructional corpus. Based on this observation, we propose a sliding window approach.

In this approach, our intra-sentential parser works with a window of two consecutive sentences, and builds a DT for the two sentences. For example, given the three sentences in Figure 7, our intra-sentential parser constructs a DT for $S_1$-$S_2$ and a DT for $S_2$-$S_3$. In this process, each sentence in a document except the first and the last will be associated with two DTs: one with the previous sentence (say $DT_p$) and one with the next (say $DT_n$). In other words, for each non-boundary sentence, we will have two decisions: one from $DT_p$ and one from $DT_n$. Our parser consolidates the two decisions and generates one or more sub-trees for each sentence by checking the following three mutually exclusive conditions one after another:

• *Same in both:* If the sentence has the same (in terms of both structure and labels) well-formed sub-tree in both $DT_p$ and $DT_n$, we take this sub-tree for the sentence. For example, in Figure 8(a), $S_2$ has the same sub-tree in the two DTs, i.e. a DT for $S_1$-$S_2$ and a DT for $S_2$-$S_3$. The two decisions agree on the DT for the sentence.

• *Different but no cross:* If the sentence has a well-formed sub-tree in both $DT_p$ and $DT_n$, but the two sub-trees vary either in structure or in labels, we pick the most probable one. For example, consider the DT for $S_1$-$S_2$ in Figure 8(a) and the DT for $S_2$-$S_3$ in Figure 8(b). In both cases $S_2$ has a well-formed sub-tree, but they differ in structure. We pick the sub-tree which has the higher probability in the two dynamic programming tables.

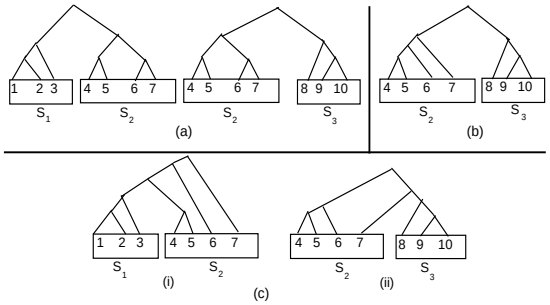


Figure 8: Extracting sub-trees for $S_2$.

• *Cross:* If either or both of $DT_p$ and $DT_n$ segment the sentence into multiple sub-trees, we pick the one with more sub-trees. For example, consider the two DTs in Figure 8(c). In the DT for $S_1$-$S_2$, $S_2$ has three sub-trees (4-5,6,7), whereas in the DT for $S_2$-$S_3$, it has two (4-6,7). So, we extract the three sub-trees for $S_2$ from the first DT. If the sentence has the same number of sub-trees in both $DT_p$ and $DT_n$, we pick the one with higher probability in the dynamic programming tables.

At the end, the multi-sentential parser takes all these sentence-level sub-trees for a document, and builds a full rhetorical parse for the document.

## 6 Experiments

### 6.1 Corpora

While previous studies on document-level parsing only report their results on a particular corpus, to show the generality of our method, we experiment with texts from two very different genres. Our first corpus is the standard *RST-DT* (Carlson et al., 2002), which consists of 385 Wall Street Journal articles, and is partitioned into a training set of 347 documents and a test set of 38 documents. 53 documents, selected from both sets were annotated by two annotators, based on which we measure human agreement. In RST-DT, the original 25 rhetorical relations defined by (Mann and Thompson, 1988) are further divided into a set of 18 coarser relation classes with 78 finer-grained relations. Our second corpus is the *Instructional* corpus prepared by (Subba and Di-Eugenio, 2009), which contains 176 how-to-do manuals on home-repair. The corpus was annotated with 26 informational relations (e.g., *Preparation-Act, Act-Goal*).

### 6.2 Experimental Setup

We experiment with our discourse parser on the two datasets using our two different parsing approaches, namely 1S-1S and the sliding window. We compare our approach with *HILDA* (Hernault et al., 2010) on RST-DT, and with the *ILP*-based approach of (Subba and Di-Eugenio, 2009) on the Instructional corpus, since they are the state-of-the-art on the respective genres. On RST-DT, the standard split was used for training and testing purposes. The results for HILDA were obtained by running the system with default settings on the same inputs we provided to our system. Since we could not run the ILP-based system of (Subba and Di-Eugenio, 2009) (not publicly available) on the Instructional corpus, we report the performances presented in their paper. They used 151 documents for training and 25 documents for testing. Since we did not have access to their particular split, we took 5 random samples of 151 documents for training and 25 documents for testing, and report the average performance over the 5 test sets.

To evaluate the parsing performance, we use the standard unlabeled (i.e., hierarchical spans) and labeled (i.e., nuclearity and relation) precision, recall and F-score as described in (Marcu, 2000b). To compare with previous studies, our experiments on RST-DT use the 18 coarser relations. After attaching the nuclearity statuses (NS, SN, NN) to these relations, we get 41 distinct relations. Following (Subba and Di-Eugenio, 2009) on the Instructional corpus, we use 26 relations, and treat the reversals of non-commutative relations as separate relations. That is, *Goal-Act* and *Act-Goal* are considered as two different relations. Attaching the nuclearity statuses to these relations gives 76 distinct relations. Analogous to previous studies, we map the $n$-$ary$ relations (e.g., *Joint*) into nested right-branching binary relations.

### 6.3 Results and Error Analysis

Table 2 presents F-score parsing results for our parsers and the existing systems on the two corpora.[2] On both corpora, our parser, namely, 1S-1S (TSP 1-1) and sliding window (TSP SW), outperform existing systems by a wide margin ($p<7.1e-05$).[3] On RST-DT, our parsers achieve absolute F-score improvements of 8%, 9.4% and 11.4% in span, nuclearity and relation, respectively, over HILDA. This represents relative error reductions of 32%, 23% and 21% in span, nuclearity and relation, respectively. Our results are also close to the upper bound, i.e. human agreement on this corpus.

On the Instructional genre, our parsers deliver absolute F-score improvements of 10.5%, 13.6% and 8.14% in span, nuclearity and relations, respectively, over the ILP-based approach. Our parsers, therefore, reduce errors by 36%, 27% and 13% in span, nuclearity and relations, respectively.

If we compare the performance of our parsers on the two corpora, we observe higher results on RST-DT. This can be explained in at least two ways. First, the Instructional corpus has a smaller amount of data with a larger set of relations (76 when nuclearity attached). Second, some frequent relations are (semantically) very similar (e.g., Preparation-Act, Step1-Step2), which makes it difficult even for the human annotators to distinguish them (Subba and Di-Eugenio, 2009).

Comparison between our two models reveals that TSP SW significantly outperforms TSP 1-1 only in finding the right structure on both corpora ($p<0.01$). Not surprisingly, the improvement is higher on the Instructional corpus. A likely explanation is that the Instructional corpus contains more leaky boundaries (12%), allowing the sliding

[2] Precision, Recall and F-score are the same when manual segmentation is used (see Marcu, (2000b), page 143).

[3] Since we did not have access to the output or to the system of (Subba and Di-Eugenio, 2009), we were not able to perform a significance test on the Instructional corpus.

| | RST-DT | | | | Instructional | | |
|---|---|---|---|---|---|---|---|
| Metrics | HILDA | TSP 1-1 | TSP SW | Human | ILP | TSP 1-1 | TSP SW |
| Span | 74.68 | **82.47*** | **82.74*†** | 88.70 | 70.35 | **79.67** | **80.88†** |
| Nuclearity | 58.99 | **68.43*** | **68.40*** | 77.72 | 49.47 | **63.03** | **63.10** |
| Relation | 44.32 | **55.73*** | **55.71*** | 65.75 | 35.44 | **43.52** | **43.58** |

Table 2: Parsing results of different models using manual (gold) segmentation. Performances significantly superior to HILDA (with p<7.1e-05) are denoted by *. Significant differences between TSP 1-1 and TSP SW (with p<0.01) are denoted by †.

| | T-C | T-O | T-CM | M-M | CMP | EV | SU | CND | EN | CA | TE | EX | BA | CO | JO | S-U | AT | EL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T-C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| T-O | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T-CM | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 7 |
| M-M | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 3 |
| CMP | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 1 | 0 | 1 | 0 | 3 | 3 | 0 | 1 | 1 | 0 | 2 |
| EV | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 2 | 11 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 12 |
| CND | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 3 | 2 |
| EN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 24 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 7 |
| CA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 0 | 4 | 2 | 2 | 7 | 0 | 3 | 11 |
| TE | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 0 | 7 | 1 | 9 | 1 | 9 | 0 | 3 | 4 |
| EX | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 5 | 0 | 12 | 0 | 1 | 3 | 0 | 3 | 12 |
| BA | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 1 | 19 | 2 | 6 | 1 | 5 | 12 |
| CO | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 2 | 33 | 7 | 0 | 0 | 9 |
| JO | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 1 | 1 | 1 | 2 | 57 | 1 | 0 | 13 |
| S-U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 85 | 1 | 0 |
| AT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 272 | 9 |
| EL | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 6 | 1 | 8 | 1 | 0 | 8 | 2 | 2 | 359 |

Figure 9: Confusion matrix for relation labels on the RST-DT test set. Y-axis represents *true* and X-axis represents *predicted* relations. The relations are Topic-Change (T-C), Topic-Comment (T-CM), Textual Organization (T-O), Manner-Means (M-M), Comparison (CMP), Evaluation (EV), Summary (SU), Condition (CND), Enablement (EN), Cause (CA), Temporal (TE), Explanation (EX), Background (BA), Contrast (CO), Joint (JO), Same-Unit (S-U), Attribution (AT) and Elaboration (EL).

window approach to be more effective in finding those, without inducing much noise for the labels. This clearly demonstrates the potential of TSP SW for datasets with even more leaky boundaries e.g., the Dutch (Vliet and Redeker, 2011) and the German Potsdam (Stede, 2004) corpora.

Error analysis reveals that although TSP SW finds more correct structures, a corresponding improvement in labeling relations is not present because in a few cases, it tends to induce noise from the neighboring sentences for the labels. For example, when parsing was performed on the first sentence in Figure 1 in isolation using 1S-1S, our parser rightly identifies the *Contrast* relation between EDUs 2 and 3. But, when it is considered with its neighboring sentences by the sliding window, the parser labels it as *Elaboration*. A promising strategy to deal with this and similar problems that we plan to explore in future, is to apply both approaches to each sentence and combine them by consolidating three probabilistic decisions, i.e. the one from 1S-1S and the two from sliding window.

To further analyze the errors made by our parser on the hardest task of relation labeling, Figure 9 presents the confusion matrix for TSP 1-1 on the RST-DT test set. The relation labels are ordered according to their frequency in the RST-DT training set. In general, the errors are produced by two different causes acting together: (i) imbalanced distribution of the relations, and (ii) semantic similarity between the relations. The most frequent relation *Elaboration* tends to mislead others especially, the ones which are semantically similar (e.g., *Explanation*, *Background*) and less frequent (e.g., *Summary*, *Evaluation*). The relations which are semantically similar mislead each other (e.g., *Temporal:Background*, *Cause:Explanation*).

These observations suggest two ways to improve our parser. We would like to employ a more robust method (e.g., *ensemble* methods with *bagging*) to deal with the imbalanced distribution of relations, along with taking advantage of a richer semantic knowledge (e.g., compositional semantics) to cope with the errors caused by semantic similarity between the rhetorical relations.

## 7 Conclusion

In this paper, we have presented a novel discourse parser that applies an optimal parsing algorithm to probabilities inferred from two CRF models: one for intra-sentential parsing and the other for multi-sentential parsing. The two models exploit their own informative feature sets and the distributional variations of the relations in the two parsing conditions. We have also presented two novel approaches to combine them effectively. Empirical evaluations on two different genres demonstrate that our approach yields substantial improvement over existing methods in discourse parsing.

## Acknowledgments

We are grateful to Frank Tompa and the anonymous reviewers for their comments, and the NSERC BIN and CGS-D for financial support.

## References


O. Biran and O. Rambow. 2011. Identifying Justifications in Written Dialogs by Classifying Text as Argumentative. *International Journal of Semantic Computing*, 5(4):363–381.

L. Carlson, D. Marcu, and M. Okurowski. 2002. RST Discourse Treebank (RST-DT) LDC2002T07. *Linguistic Data Consortium, Philadelphia.*

V. Feng and G. Hirst. 2012. Text-level Discourse Parsing with Rich Linguistic Features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL '12, pages 60–68, Jeju Island, Korea. Association for Computational Linguistics.

S. Fisher and B. Roark. 2007. The Utility of Parse-derived Features for Automatic Discourse Segmentation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, ACL '07, pages 488–495, Prague, Czech Republic. Association for Computational Linguistics.

M. Galley and K. McKeown. 2003. Improving Word Sense Disambiguation in Lexical Chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI '07, pages 1486–1488, Acapulco, Mexico.

H. Hernault, H. Prendinger, D. duVerle, and M. Ishizuka. 2010. HILDA: A Discourse Parser Using Support Vector Machine Classification. *Dialogue and Discourse*, 1(3):1–33.

S. Joty, G. Carenini, and R. T. Ng. 2012. A Novel Discriminative Framework for Sentence-Level Discourse Analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 904–915, Jeju Island, Korea. Association for Computational Linguistics.

H. LeThanh, G. Abeysinghe, and C. Huyck. 2004. Generating Discourse Structures for Written Texts. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Geneva, Switzerland. Association for Computational Linguistics.

A. Louis, A. Joshi, and A. Nenkova. 2010. Discourse Indicators for Content Selection in Summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, pages 147–156, Tokyo, Japan. Association for Computational Linguistics.

W. Mann and S. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281.

D. Marcu and A. Echihabi. 2002. An Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 368–375. Association for Computational Linguistics.

D. Marcu. 2000a. The Rhetorical Parsing of Unrestricted Texts: A Surface-based Approach. *Computational Linguistics*, 26:395–448.

D. Marcu. 2000b. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

J. Morris and G. Hirst. 1991. Lexical Cohesion Computed by Thesaural Relations as an Indicator of Structure of Text. *Computational Linguistics*, 17(1):21–48.

R. Prasad, A. Joshi, N. Dinesh, A. Lee, E. Miltsakaki, and B. Webber. 2005. The Penn Discourse TreeBank as a Resource for Natural Language Generation. In *Proceedings of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*, pages 25–32, Birmingham, U.K.

S. Somasundaran, 2010. *Discourse-Level Relations for Opinion Analysis*. PhD thesis, University of Pittsburgh.

R. Soricut and D. Marcu. 2003. Sentence Level Discourse Parsing Using Syntactic and Lexical Information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, NAACL-HLT '03, pages 149–156, Edmonton, Canada. Association for Computational Linguistics.

C. Sporleder and M. Lapata. 2005. Discourse Chunking and its Application to Sentence Compression. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 257–264, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

C. Sporleder and A. Lascarides. 2004. Combining Hierarchical Clustering and Machine Learning to Predict High-Level Discourse Structure. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Geneva, Switzerland. Association for Computational Linguistics.

M. Stede. 2004. The Potsdam Commentary Corpus. In *Proceedings of the ACL-04 Workshop on Discourse Annotation*, Barcelona. Association for Computational Linguistics.

R. Subba and B. Di-Eugenio. 2009. An Effective Discourse Parser that Uses Rich Linguistic Information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL '09, pages 566–574, Boulder, Colorado. Association for Computational Linguistics.

C. Sutton and A. McCallum. 2012. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

C. Sutton, A. McCallum, and K. Rohanimanesh. 2007. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machine Learning Research (JMLR)*, 8:693–723.

S. Verberne, L. Boves, N. Oostdijk, and P. Coppen. 2007. Evaluating Discourse-based Answer Extraction for Why-question Answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736, Amsterdam, The Netherlands. ACM.

N. Vliet and G. Redeker. 2011. Complex Sentences as Leaky Units in Discourse Parsing. In *Proceedings of Constraints in Discourse*, Agay-Saint Raphael, September.