编号：

****

毕业设计(论文)外文翻译

（原文）

学 院：计算机与信息安全学院

专 业： 软件工程

学生姓名： 薛天杭

学 号： 1400330129

指导教师单位： 计算机与信息安全学院

姓 名： 黄桂敏

职 称： 教授

2020年6月20日

RST-Style Discourse Parsing

and Its Applications in Discourse Analysis

by

Wei (Vanessa) Feng

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy

Graduate Department of Computer Science University of Toronto

## Abstract

Discourse parsing is the task of identifying the relatedness and the particular discourse relations among various discourse units in a text. In particular, among various theoretical frameworks of discourse parsing, I am interested in Rhetorical Structure Theory (RST). I hypothesize that, given its ultimate success, discourse parsing can provide a general solution for use in many downstream applications.

This thesis is composed of two major parts. First, I overview my work on discourse seg- mentation and discourse tree-building, which are the two primary components of RST-style discourse parsing. Evaluated on the RST Discourse Treebank (RST-DT), both of my discourse segmenter and tree-builder achieve the state-of-the-art performance.

Later, I discuss the application of discourse relations to some specific tasks in the analysis of discourse, including the evaluation of coherence, the identification of authorship, and the de- tection of deception. In particular, I propose to use a set of application-neutral features, which are derived from the discourse relations extracted by my discourse parser, and compare the performance of these application-neutral features against the classic application-specific ap- proaches to each of these tasks. On the first two tasks, experimental results show that discourse relation features by themselves often perform as well as those classic application-specific fea- tures, and the combination of these two kinds of features usually yields further improvement. These results provide strong evidence for my hypothesis that discourse parsing is able to pro-

vide a general solution for the analysis of discourse. However, we failed to observe a similar effectiveness of discourse parsing on the third task, the detection of deception. I postulate that this might be due to several confounding factors of the task itself.

## Acknowledgements

I am sincerely grateful to my supervisor Professor Graeme Hirst at the Department of Com- puter Science, University of Toronto. It has been my great pleasure to work with him since five years ago, when I began my life in Toronto as a master’s student under his supervision and then proceeded as a Ph.D. student in 2011. Graeme is such a gentleman with great sense of hu- mor. He never fails to provide me with insightful thoughts, recommendations, and inspiration throughout my research. He is a true mentor, who always shows respect and kindness to his students. Moreover, without his patient and careful editing of all my research papers, I would still be a novice in scientific writing, struggling for each presentation that I need to give.

I would like to thank my committee members, Professor Suzanne Stevenson and Professor Gerald Penn, for their helpful advice and criticism while I revolved the directions of my re- search. Although I normally work exclusively with my supervisor on my research projects, the regular checkpoint meetings with my committee members offered great opportunities to learn interesting and useful ideas from different perspectives. I am also grateful to Professor Michael Strube from HITS gGmbH, Germany, who was very kind to agree to serve as my external ex- aminer; to Professor Frank Rudzicz from the CL group and to Professor Jack Chambers from the Linguistics Department, who agreed to be the new committee member for my final the- sis defence. Without their valuable suggestions and insightful criticism, my final thesis work would have been of much less quality.

I am also indebted to my parents Huaying Sun and Hanjie Feng, who stayed in my home- town, Shanghai, China, when I was pursuing my Ph.D. studies in a foreign country. Without their support, I would have a much harder time in Toronto.

I would like to express my gratitude to all my colleagues in the CL group at University of Toronto, which is such an amazing group of talents, and all my friends in Toronto, who spent their time hanging out with me, preventing me from becoming a dull Ph.D. nerd.

Finally, I want to thank the Natural Sciences and Engineering Research Council of Canada and the University of Toronto for their financial support for my research.

# Chapter 1 Introduction

No unit of a well-written text is completely isolated; interpretation requires understanding the relation between the unit and the context. Most rhetorical theories assume a hierarchical structure of discourse, where several small units of texts are related to each other to form a larger unit, which can then be related to other units. From this perspective, building the hierarchical discourse structure for a given text is similar to syntactic parsing, whose purpose is to build a hierarchical structure of a given sentence with respect to the grammatical relations among its text units. Therefore, discovering the hierarchical discourse relations in the text is termed “discourse parsing”.

My ultimate hypothesis in this thesis is that discourse parsing can be successfully done automatically with sufficiently high accuracy, and, given its success, discourse parsing would be able to provide a **general** solution to a variety of problems in the analysis of discourse structures. In this thesis, in order to evaluate my hypothesis, I will first present our work on de- veloping an automatic discourse parser and compare its performance against human judgment. Moreover, based on our parser, I will apply discourse parsing on three particular applications of discourse analysis, and observe how features derived from discourse parsing affect those applications.

The generality of discourse parsing is two-fold: Firstly, it can work on different levels of granularity — from sentences to paragraphs, and finally the whole document. Secondly, discourse parsing aims to discover not only the relatedness of two given text units, e.g., whether they belong to the same subtopic or not, but also the exact coherence relation between them, e.g., Contrast, Causal, and Explanation, which can, but normally does not have to, depend on any specific target application. Therefore, discourse parsing is able to provide rich information about the content and the discourse structure of the text, which is clearly a powerful tool for many applications in the analysis of discourse.

In this chapter, I will first introduce Rhetorical Structure Theory, one of the most widely accepted frameworks for discourse analysis. In addition, I will also briefly introduce the Penn Discourse Treebank, a corpus developed in accordance with another popular discourse frame- work, and its related work, to shed some light on the discussion of discourse analysis from other theories and philosophies.

The thesis is organized as the following. In Part [I,](#_bookmark17) I will discuss the two major tasks in RST-style discourse parsing, namely, discourse segmentation and discourse tree-building, and the related work conducted on these two tasks. By the end of Chapter [3,](#_bookmark39) all the necessary components of an RST-style discourse parser will have been presented. In Part [II](#_bookmark87) of this thesis, we will see several specific applications in discourse analysis, on which I will evaluate my ulti- mate hypothesis of the general usefulness of discourse parsing. Those applications include the evaluation of coherence (Chapter [6),](#_bookmark88) the identification of authorship (Chapter [7),](#_bookmark129) and the detec- tion of deception (Chapter [8).](#_bookmark169) I will first describe the application-specific approaches to each of these problems, which are well-established and classic solutions to each specific problem. Afterwards, I will proceed to discuss how information derived from our application-neutral discourse parser can be incorporated into each of these problems and enhance the overall per- formance, and therefore provide evidence to support the postulated generality of discourse parsing.

## Rhetorical Structure Theory

Rhetorical Structure Theory (RST) [(Mann and Thompson, 1988)](#_bookmark262) is one of the most widely ac- cepted frameworks for discourse analysis, and was adopted in the pioneering work of discourse parsing by [Marcu (1997).](#_bookmark263) In the framework of RST, a coherent text, or a fairly independent text fragment, can be represented as a discourse tree. In an RST-style discourse tree, the leaf nodes are non-overlapping text spans called *elementary discourse units* (EDUs) — these are the minimal text units of discourse trees (see Section [1.1.1)](#_bookmark3) — and internal nodes are the concate- nation of continuous EDUs. Adjacent nodes are related through particular discourse relations (see Section [1.1.2](#_bookmark5) for detail) to form a discourse subtree, which can then be related to other adjacent nodes in the tree structure. In this way, the hierarchical tree structure is established.

As discussed in length by [Taboada and Mann (2006),](#_bookmark289) in its original proposal, RST was designed as an open system, allowing flexibility for researchers working on different domains and applications. There are only a few fixed parts enforced in the original design of RST, including dividing a text into a set of non-overlapping discourse units and the tightness between discourse relations and text coherence. Therefore, in order to proceed with introducing the fine detail of the theory, below, I will make connection to a particular annotation scheme and its resulting corpus, the RST Discourse Treebank (RST-DT), and focus on the corresponding definitions as provided by the annotation guidance in this corpus.

The RST Discourse Treebank (RST-DT) [(Carlson et al., 2001),](#_bookmark211) is a corpus annotated in the framework of RST, published by the Linguistic Data Consortium (LDC) with catalog number LDC2002T07 and ISBN 1-58563-223-6[1](#_bookmark0). It consists of 385 documents (347 for training and 38 for testing) from the *Wall Street Journal*. RST-DT has been widely used as a standard benchmark for research in RST-style discourse parsing, as it provides a systematic guideline in defining several intuition-based concepts in the original development of RST by Mann and Thompson, including the definitions of EDUs and several discourse relations. Throughout this thesis, the term *RST-style discourse parsing* will refer to the specific type of discourse parsing in accordance with the annotation framework in RST-DT.

### Elementary Discourse Units

As stated by Mann and Thompson (1988, p. 244), RST provides a general way to describe the relations among clauses in a text, whether or not they are grammatically or lexically signalled. Therefore, elementary discourse units (EDUs), which are the minimal discourse units, are not necessarily syntactic clauses, nor are there explicit lexical cues to indicate boundaries.

In RST-DT, to provide a balance between the consistency and the granularity of annotation, the developers chose clauses as the general basis of EDUs, with the following set of exceptions.consists of one single EDU, instead of two EDUs segmented before *can*. So simply relying on syntactic information is not sufficient for EDU segmentation, and more sophisticated ap- proaches need to be taken. In Chapter [2,](#_bookmark18) I will present my work on developing a discourse segmentation model for determining EDU boundaries.

### Inventory of Discourse Relations

According to RST, there are two types of discourse relation, hypotactic (“mononuclear”) and paratactic (“multi-nuclear”). In mononuclear relations, one of the text spans, the *nucleus*, is more salient than the other, the *satellite*, while in multi-nuclear relations, all text spans are equally important for interpretation.

In RST-DT, the original 24 discourse relations defined by [Mann and Thompson (1988)](#_bookmark262) are further divided into a set of 78 fine-grained rhetorical relations in total (53 mononuclear and 23 multi-nuclear), which provides a high level of expressivity.

The 78 relations can be clustered into 16 relation classes, as shown in Table [1.1.](#_bookmark7) For example, the class Cause is a coarse-grained clustering of the relation types *cause*, *result*, and *consequence*. Moreover, three relations are used to impose structure on the tree: Textual- Organization, Span, and Same-Unit (used to link parts of units separated by embedded units or spans). With nuclearity attached, there are 41 distinct types of discourse relation class, as shown in Table [1.2.](#_bookmark9) For example, there can be three distinct types of Contrast relation: Contrast[N][N] (both spans are nucleus), Contrast[N][S] (the first span is nucleus and the other is satellite), and Contrast[S][N]. And these 41 distinct types of relation class are the level of granularity on which most current work on classifying RST-style discourse relations is focused.

The definition of each particular RST relation is based on four elements: (1) Constraints on the nucleus; (2) Constraints on the satellite; (3) Constraints on the combination of nucleus and satellite; and (4) Effect achieved on the text receiver. For example, Table [1.3](#_bookmark10) illustrates the definition of the Condition class, with respect to the four definition elements described above[2](#_bookmark0).

### An Example of RST-Style Discourse Tree Representation

The example text fragment shown in Figure [1.1](#_bookmark8) consists of four EDUs (*e*1-*e*4), segmented by square brackets. Its discourse tree representation is shown below in the figure, following the

notational convention of RST. The two EDUs *e*1 and *e*2 are related by a mononuclear relation Attribution, where *e*1 is the more salient span, as denoted by the arrow pointing to *e*1. The span (*e*1-*e*2) and the EDU *e*3 are related by a multi-nuclear relation Same-Unit, where they are equally salient, as denoted by the two straight lines connecting (*e*1-*e*2) and *e*3. Finally, the span (*e*1-*e*3) is related to *e*4 with a mononuclear relation Condition to form the complete dis- course tree for the sentence. In this way, we have a tree-structured hierarchical representation corresponding to the entire sentence.

Note that no constraint is imposed to the scope for an RST-style discourse tree representa- tion, in the sense that the tree-structured representation could be used to describe the discourse structures for texts on different levels: from sentences, to paragraphs, and finally to the entire text. Due to such a capacity to represent discourse relations on different levels of granularity, RST is of particular interest to many researchers in the field of discourse analysis. More im- portantly, it fits nicely with the goal outlined in the beginning of this chapter, i.e., to provide a general solution to a variety of problems in the analysis of discourse structures. As we shall see in later chapters, a number of problems of discourse analysis do benefit from identifying RST-style discourse relations in texts.

### RST-Style Discourse Parsing Pipeline

Due to the nature of the tree-structured representation of discourse relations, RST-style dis- course parsing typically adopts a pipeline framework which consists of two individual stages:

Discourse segmentation: Segment a raw text into non-overlapping EDUs, which are the bottom-level discourse units of the text-level discourse tree representation.

Discourse tree-building: Given the set of segmented EDUs from Stage 1, adopt ap- propriate strategies to build the discourse tree corresponding to the full text, e.g., the example discourse tree shown in Figure [1.1.](#_bookmark8)

In Part [I,](#_bookmark17) Chapters [2](#_bookmark18) and [3](#_bookmark39) will discuss related work and my own work on these two stages in detail.

### Issues with RST and RST-DT

Over its history of nearly three decades, RST has gained unparalleled popularity among various discourse theories, and has been applied to a variety of applications, not only for text generation

— its original motivation and design purpose — but also for a large number of tasks in text understanding. Not coincidentally, there also has been much literature dedicated to questioning or criticizing several aspects of RST.

However, as mentioned previously, according to [Taboada and Mann (2006),](#_bookmark289) most of these criticisms stem from misunderstanding of, or digression from, the original design of RST. In contrast, RST should be considered as an open system with a high extent of flexibility, and encourages innovations and adaption for specific applications and domains. In fact, only the following general rules are enforced when applying RST-style discourse analysis:

*Analysis of a text is performed by applying schemas that obey constraints of completedness (one schema application contains the entire text); connectedness (each span, except for the span that contains the entire text, is either a minimal unit or a constituent of another schema application); uniqueness (each schema application contains a di*ff*erent set of text spans); and adjacency (the spans of each schema application constitute one contiguous text span).*

— [Taboada and Mann (2006),](#_bookmark289) p. 5.

Nevertheless, in terms of current computational approaches toward RST-style discourse analysis, especially due to the use of RST-DT as the benchmark dataset, there are indeed several commonly accepted formulations which are in fact questionable. Here, I briefly talk about some most prominent issues with regard to RST-DT and RST in general.

First of all, the clause-based EDU segmentation rule has been criticized as being too coarse- grained and being unable to capture a few linguistic phenomena. For example, as specified by RST-DT, clauses that are subjects or objects of a main verb are not treated as EDUs (see Section [1.1.1);](#_bookmark3) therefore, the following sentence is regarded as one single EDU.

*His studying hard makes him pass the exam.*

However, this segmentation is not sufficiently fine-grained, as it precludes any representation of the underlying causal relation between the two actions *studying hard* and *passing the exam*. Furthermore, there are concerns about whether it is feasible to represent a text by a tree- shaped discourse structure, and whether such a tree-shaped representation is the only valid representation for the given text. Admittedly, it might be a too strong assumption that a single tree is able to capture the discourse structure in the entire text: For a text written by an average writer, it is normal to see occasional digression from the main topic, or gradual development of thoughts, such that there is a certain degree of coherence within a small text fragment, while relations between different fragments are rather loose. Therefore, to deal with these complications in real texts, [Wolf and Gibson (2005)](#_bookmark291) propose to use an alternative graph-based data structure for analysis, which allows cross dependencies and nodes with more than one parent. However, despite their greater expressivity, graph-based representations also impose

greater challenges to automatic discourse parsing.

Finally, the adjacency constraint in RST, i.e., the spans of each discourse relation constitute one contiguous text span, is not entirely justified either, and the subtlety lies in the presence of embedded discourse units. According to the definition in RST-DT, an embedded discourse unit has one or both of the following properties: (1) It breaks up a unit which is legitimately an EDU on its own; (2) It modifies a portion of an EDU only, not the entire EDU. For instance, Figure [1.2](#_bookmark13) shows a text fragment with three EDUs, where the second EDU is an embedded one. The embedded EDU *e*2 breaks up *e*1 and *e*3, which, when concatenated, is a legitimate EDU on its own. Therefore, in order to characterize the coherence between *e*1 and *e*3, which is essentially a continuation, the developers of RST-DT had to invent a pseudo-relation, called[But maintaining the key components of his strategy]*e*1 [— a stable exchange rate and high levels of imports —]*e*2 [will consume enormous amounts of foreign exchange.]*e*3

Same-Unit. However, in this way, the adjacency constraint is violated by the presence of the embedded EDU *e*2.

## The Penn Discourse Treebank and PDTB-Style Discourse Parsing

The Penn Discourse Treebank (PDTB) [(Prasad et al., 2008)](#_bookmark278) is another annotated discourse cor- pus. Its text is a superset of that of RST-DT (2159 Wall Street Journal articles). Unlike RST- DT, PDTB does not follow the framework of RST; rather, it follows Discourse Lexicalized Tree Adjoining Grammar (D-LTAG) [(Webber, 2004),](#_bookmark290) which is a lexically grounded, predicate- argument approach with a different set of predefined discourse relations. In this framework, a discourse connective (e.g., *because*) is considered to be a predicate that takes two text spans as its arguments. The argument that the discourse connective structurally attaches to is called **Arg2**, and the other argument is called **Arg1**; unlike in RST, the two arguments are not distin- guished by their saliency for interpretation.

An example annotation from PDTB is shown in Example [1.1,](#_bookmark15) in which the explicit con- nective (*when*) is underlined, and the two arguments, **Arg1** and **Arg2**, are shown in *italics* and **bold** respectively. The example is annotated with its three-level hierarchical relation type: it is of the contingency class, the cause type, and the reason subtype.

**Example 1.1.** *Use of dispersants was approved* when **a test on the third day showed some positive results**. (contingency:cause:reason)

(wsj 1347)

In PDTB, relation types are organized hierarchically: there are 4 *classes*: Expansion, Com-

parison, Cause, and Temporal, which can be further divided into 16 *types* and 23 *subtypes*.

After the release of PDTB, several attempts have been made to recognize PDTB-style re- lations. The corpus study conducted by [Pitler et al. (2008)](#_bookmark276) showed that overall discourse con- nectives are mostly unambiguous and allow high accuracy classification of discourse relations: they achieved over 90% accuracy by simply mapping each connective to its most frequent sense. Therefore, the real challenge of discourse parsing lies in implicit relations (discourse relations which are not signaled by explicit connectives), and recent research emphasis is on recognizing these implicit discourse relations.

In particular, [Lin et al.](#_bookmark257) [(2009)](#_bookmark257) attempted to recognize such implicit discourse relations in PDTB by using four classes of features — contextual features, constituent parse features, dependency parse features, and lexical features — and explored their individual influence on performance. They showed that the production rules extracted from constituent parse trees are the most effective features, while contextual features are the weakest. Subsequently, they fully implemented an end-to-end PDTB-style discourse parser [(Lin et al., 2014).](#_bookmark259) [Pitler et al. (2009)](#_bookmark277) adopted a similar set of linguistically motivated features, and performed a series of one vs. others classification for recognizing implicit discourse relations of various types.

Later, based on the insight of [Pitler et al. (2008)](#_bookmark276) described above, [Zhou et al. (2010)](#_bookmark295) pro- posed to solve the problem of recognizing implicit relations by first predicting the appropriate discourse connective and then mapping the predicted connective to its most frequent discourse sense. Specifically, Zhou et al. trained a language model to evaluate the perplexity of a set of synthetic texts, which are formed by inserting every possible discourse connective into the implicit discourse relation of interest. The most probable connective is chosen from the syn- thetic text with the lowest perplexity. However, this approach did not achieve much success.

The main reason is that the synthetic texts formed in this way differ by the inserted connective only; therefore, the computation of perplexity would take into account a very limited number of contextual words near the connective (typically trigram sequences are used in the computa- tion). In fact, usually a much larger proportion of the text is required for correctly interpreting the particular implicit relation.

A more recent research focus of recognizing implicit discourse relations is on feature re- finement. [Park and Cardie](#_bookmark273) [(2012)](#_bookmark273) applied a simple greedy feature selection on the sets of features previously used by [Pitler et al. (2009)](#_bookmark277) to enhance the performance on implicit relation recognition. Recently, [Rutherford and Xue (2014)](#_bookmark282) argued that word pairs, which are shown to be the most effective features for recognizing implicit relations, suffer from sparsity issue when available training samples are limited. Therefore, they proposed to overcome this sparsity issue through representing relations by Brown word cluster pairs[3](#_bookmark0) and coreference patterns. [Ruther-](#_bookmark282) [ford and Xue](#_bookmark282) achieved the current state-of-the-art one-vs-others classification performance of recognizing Level-1 implicit relations in PDTB, ranging from an *F*1 score of 28% (Temporal vs. others) to 80% (Expansion vs. others).

## Di**ff**erences Between the Two Discourse Frameworks

As the two most popular frameworks in the study of discourse parsing, RST and PDTB have several inherent distinctions, which make the two frameworks potentially useful for different kinds of application. In Part [II,](#_bookmark87) we will see several specific applications of discourse analysis, and the different effects of the analysis generated by the two frameworks on the applications.

The most important difference between the two frameworks is that, in RST-style parsing, the text is ultimately represented as a discourse tree, and thus the discourse structure is fully annotated on different granularities of the text; in PDTB, however, there does not necessarily exist a tree structure covering the full text, i.e., PDTB-style discourse relations exist only in

Brown word clustering is a form of hierarchical clustering of words based on the classes of previous words, proposed by [Brown et al. (1992).](#_bookmark210)a very local contextual window. As will be demonstrated in Section [6.3,](#_bookmark111) the full hierarchy of discourse structure can be quite useful for some particular applications.

Moreover, since, in RST-style parsing, a text is first segmented into non-overlapping EDUs, which are the smallest units in the final discourse tree representation, any given valid discourse unit in the text therefore participates in at least one discourse relation. In other words, the discourse relations in RST-style parsing cover the entire text. However, this is generally not true in PDTB-style discourse parsing. Therefore, the RST-style discourse relations have better coverage of the text than PDTB-style discourse relations. This property of better coverage can be useful for some particular applications as well.

Finally, in general, RST-style discourse relations are more constrained than PDTB-style relations: RST-style relations can exist only between adjacent text spans (a single EDU or the concatenation of multiple continuous EDUs), and two RST-style discourse relations in a text can only be one of the two cases: the texts corresponding to the two relations are completely disjoint with each other, or the text span of one relation is a proper sub-sequence of the text span of the other relation, i.e., the two text spans cannot partially overlap with each other. However, this constraint is not found in PDTB-style discourse relations, and thus there is more flexibility in the annotation for PDTB-style relations.

The differences discussed above do not necessarily lead to a definite statement that one discourse framework is superior to the other; rather, they illustrate the differences between the underlying philosophies of the two frameworks, and thus, we should choose the more suitable one depending on the particular applications in which we are interested. For instance, due to the existence of hierarchical structure and complete coverage in RST-style discourse repre- sentation, RST-style discourse parsing is probably more suitable for those applications where global understanding of the text is required, such as the applications to be discussed in later parts of this thesis. In contrast, because PDTB-style discourse parsing is lexically grounded and represents discourse relations in a fairly local context window, it is thus more effective for those applications where we wish to pinpoint the relevant information and may have little in-terest in the remaining of the text. Examples of such applications include information retrieval and question answering.

# Chapter 2 Discourse Segmentation

As described in Section [1.1,](#_bookmark2) for RST-style discourse parsing, identifying the boundaries of discourse units is the very first stage in the pipeline workflow; therefore, its performance is crucial to the overall accuracy. In this chapter, I will first present some previous work on RST- style discourse segmentation, and then discuss about my own CRF-based discourse segmenter.

## Previous Work

Conventionally, the task of automatic EDU segmentation is formulated as: given a sentence, the segmentation model identifies the boundaries of the composite EDUs by predicting whether a boundary should be inserted before each particular token in the sentence. In particular, previous work on discourse segmentation typically falls into two major frameworks.

The first is to consider each token in the sentence sequentially and independently. In this framework, the segmentation model scans the sentence token by token, and uses a binary classi- fier, such as a support vector machine or logistic regression, to predict whether it is appropriate to insert a boundary before the token being examined. Examples following this framework include [Soricut and Marcu (2003),](#_bookmark283) [Subba and Di Eugenio (2007),](#_bookmark286) [Fisher and Roark (2007),](#_bookmark229) and [Joty et al. (2012).](#_bookmark243)

given sentence is considered as a whole, and the model assigns a label to each token, indicating whether this token is the beginning of an EDU. Conventionally, the class label *B* is assigned to those tokens which serve as the *beginning* of an EDU, and the label *C* is assigned to other tokens. Because the beginning of a sentence is trivially the beginning of an EDU, the first token in the sentence is excluded in this labeling process. For example, Figure [2.1](#_bookmark20) illustrates this sequential labeling process. The example sentence consists of 23 tokens, separated by whitespaces, and the last 22 tokens are considered in the sequential labeling process. Each token is assigned a label, *B* or *C*, by the labeling model. If the token is labeled as *B*, e.g., the token *that* and the token *to* in boldface, an EDU boundary is placed before it. Therefore, the sentence is segmented into three EDUs, indicated by the square bracket pairs. A representative work following this sequential labeling framework is [Hernault et al.](#_bookmark233) [(2010a),](#_bookmark233) in which the sequential labeling is implemented using Conditional Random Fields (CRFs).

An interesting exception to the above two major frameworks is [Bach et al.’](#_bookmark203)s [(2012)](#_bookmark203) rerank- ing model, which obtains the best segmentation performance reported so far: for the *B* class, the *F*1 score is 91.0% and the macro-average over the *B* and *C* classes is 95.1%. The idea is to train a ranking function whose input is the *N*-best output of a base segmenter and outputs a reranked ordering of these *N* candidates. In their work, Bach et al. used a similar CRF-based segmenter to Hernault et al.’s as a base segmenter.

Because the reranking procedure is almost orthogonal to the implementation of the base segmenter, it is worthwhile to explore the enhancement of base segmenters for further per- formance improvement. With respect to base segmenters, which typically adopt the two ma-jor frameworks introduced previously, the best performance is reported by [Fisher and Roark](#_bookmark229) [(2007),](#_bookmark229) with an *F*1 score of 90.5% for recognizing in-sentence EDU boundaries (the *B* class), using three individual feature sets: basic finite-state features, full finite-state features, and context-free features.

Existing base segmentation models, as introduced in the beginning of this section, have certain limitations. First, the adopted feature sets are all centered on individual tokens, such as the part-of-speech of the token, or the production rule of the highest node in the syntactic tree which the particular token is the lexical head of. Although contextual information can be partially captured via features such as *n*-grams or part-of-speech *n*-grams, the representation capacity of these contextual features might be limited. In contrast, we hypothesize that, instead of utilizing features centered on individual tokens, it is beneficial to equally take into account the information from pairs of adjacent tokens, in the sense that the elementary input unit of the segmentation model is a pair of tokens, in which each token is represented by its own set of features. Moreover, existing models never re-consider their previous segmentation decisions, in the sense that the discourse boundaries are obtained by running the segmentation algorithm only once. However, since individual decisions are inter-related with one another, by perform- ing a second pass of segmentation incorporating features which encode global characteristics of the segmentation, we may be able to correct some incorrect segmentations of the initial run. Therefore, in this work, we propose to overcome these two limitations by our pairing features and a two-pass segmentation procedure, to be introduced in Section [2.2.](#_bookmark21)

## Methodology

Figure [2.2](#_bookmark22) shows our segmentation model in the form of a linear-chain Conditional Random Field. Each sentence is represented by a single linear chain. For each pair of adjacent tokens in a sentence, i.e., *Ti*−1 and *Ti*, there is an associated binary node *Li* to determine the label of the pair, i.e., the existence of a boundary in between: if *Li* = *B*, an EDU boundary is inserted

before *Ti*; if *Li* = *C*, the two adjacent tokens are considered a continuous portion in an EDU.

We choose a CRF-based model to label the whole sequence of tokens in a sentence, because a CRF is capable of taking into account the sequential information in the context, and solving the problem of determining boundaries in one single pass, which has been shown to be effective by [Hernault et al. (2010a)](#_bookmark233) and [Bach et al. (2012).](#_bookmark203) This sequential labeling framework is also beneficial to the training process, in the sense that no additional effort needs to be made to deal with the sparsity of EDU boundaries in the data, which is usually an issue for traditional binary classifiers.

As introduced previously, our segmentation model differs from previous work on RST-style discourse segmentation in two important ways.

First, rather than using a feature representation centered on a single token (possibly with some specifically designed features to partially incorporate contextual information), our bound- ary nodes take the input from a pair of adjacent tokens, to fully incorporate contextual infor- mation, allowing competition between neighboring tokens as well.

Secondly, rather than producing predictions of EDU boundaries by one pass of model ap- plication, we adopt a two-pass segmentation algorithm, which works as follows. We first apply our segmentation model once for each sentence. We then perform a second pass of segmen- tation, by considering some **global features** (to be described in Section [2.3)](#_bookmark23) derived from the initial segmentation. The intuition behind these novel global features is that whether a given token should be tagged as an EDU boundary sometimes depends on the neighboring EDU boundaries. For example, as suggested by [Joty et al. (2012),](#_bookmark243) since EDUs are often multi-word expressions, the distance between the current token and the neighboring boundaries can be a useful indication. In addition, it is also helpful to know whether the tokens between the cur- rent token and the neighboring boundary form a valid syntactic constituent. Since these global indicators are available only if we have an initial guess of EDU boundaries, a second pass of segmentation is necessary.

## Features

As shown in Figure [2.2,](#_bookmark22) each boundary node *Bi* in the linear-chain CRF takes the input of a pair of adjacent tokens, *Ti* and *Ti*+1, in the sentence. Each such pair is encoded using a list of surface lexical and syntactic features, as shown below. The features are partitioned into three subsets: basic features, global features, and contextual features, where the basic and contextual features are applicable for both the first and second pass, and the global features are applicable for the second pass only.

## Comparison with Other Models

We first study how our proposed two-pass discourse segmenter based on pairing features per- forms against existing segmentation models. In this experiment, we train our linear-chain CRF models on the RST Discourse Treebank (RST-DT) [(Carlson et al., 2001),](#_bookmark211) which is a large dis- course corpus annotated in accordance with RST. By convention, the corpus is partitioned into a training set of 347 documents and a test set of 38 documents. The detailed characteristics of the corpus are shown in Table [2.1.](#_bookmark24)

The data are preprocessed using Charniak and Johnson’s reranking parser [(Charniak and](#_bookmark213) [Johnson, 2005)](#_bookmark213) to obtain syntactic structures. Our linear-chain CRFs are designed using CRF- Suite [(Okazaki, 2007),](#_bookmark270) which is a fast implementation of linear-chain CRFs.

To apply our two-pass segmentation strategy (introduced in Section [2.2),](#_bookmark21) we first train our model by representing each sentence with a single linear chain, using the basic features and the contextual features as shown in Section [2.3.](#_bookmark23) For both the training and the test set, we apply the trained one-pass model to obtain an initial EDU segmentation for each sentence. We then derive global features from this initial segmentation, and train our second-pass CRF model, together with the basic and the contextual features.

Two evaluation methods have been used in this task: the first is to evaluate the precision, recall, and *F*1 scores of retrieving the in-sentence boundaries (the *B* class), which is the class that we care more about. The second is to evaluate the performance of both the two classes, *B* and *C*, based on the macro-averaged precision, recall, and *F*1 scores of retrieving each class.

Table [2.2](#_bookmark26) demonstrates the performance evaluated on the *B* class. We compare against several existing models. In the first section, CRFSeg [(Hernault et al., 2010a)](#_bookmark233) is a model that adopts a similar CRF-based sequential labeling framework as ours, but with no pairing and global features involved. The second section lists four previous works following the framework of independent binary classification for each token, including SPADE [(Soricut and Marcu,](#_bookmark283) [2003),](#_bookmark283) S&E [(Subba and Di Eugenio, 2007),](#_bookmark286) Joty et al. [(Joty et al., 2012),](#_bookmark243) and F&R [(Fisher](#_bookmark229) [and Roark, 2007).](#_bookmark229) The last model, Reranking [(Bach et al., 2012),](#_bookmark203) implements a discriminative reranking model by exploiting subtree features to rerank the *N*-best outputs of a base CRF segmenter, and obtained the best segmentation performance reported so far[1](#_bookmark0). As can be seen, in comparison to all the previous models, our two-pass model obtains the best performance on all three metrics across two classes. In fact, we obtain the same recall as Joty et al., but their precision is noticeably lower than ours. With respect to the *F*1 score, our model achieves an error-rate reduction of 17.8% over the best baseline, i.e., Reranking, and approaches to level of 95% of human performance on this task[2](#_bookmark0). Moreover, since the reranking framework of Bach et al. is almost orthogonal to our two-pass methodology, in the sense that our two- pass segmentation model can serve as a stronger base segmenter, further improvement can be expected by plugging our two-pass model into the reranking framework.

Table [2.3](#_bookmark27) demonstrates the performance evaluated on both classes and their macro-average. Only two previous models, CRFSeg and Reranking, reported their performance based on this evaluation, so other previous models are not included in this comparison. As can be seen, among the three models considered here, our two-pass segmentation model with pairing fea- tures performs the best not only on the *B* class but also on the *C* class, resulting in a macro- averaged *F*1 score of 96.0%.

## Error Propagation to Discourse Parsing

As introduced previously, discourse segmentation is the very first stage in an RST-style dis- course parser. Therefore, it is helpful to evaluate how the overall performance of discourse parsing is influenced by the results of different segmentation models.

To evaluate the performance, we use the standard unlabeled and labeled F-score for Span, Nuclearity, and Relation, as defined by [Marcu (2000).](#_bookmark264) Moreover, to further illustrate the effect of automatic segmentation on different levels of the text, we conduct the evaluation on the intra-sentential, multi-sentential, and text levels separately. On the intra-sentential level, the evaluation units are discourse subtrees which do not cross sentence boundaries. On the multi- sentential level, all discourse subtrees which span at least two sentences are considered. On the text level, all discourse subtrees are evaluated.

The results are shown in Table [2.4.](#_bookmark28) As can be seen, on the intra-sentential level, the in- fluence of segmentation is significant. Evaluated on Span, Nuclearity, and Relation, using our own segmentation results in a 10% difference in F-score (*p* < .01 in all cases)[4](#_bookmark0), while the dif- ference is even larger when using Joty et al.’s segmentation. Nonetheless, the overall parsing performance is significantly better (*p* < .01) when using our segmentation model than using Joty et al.’s.

However, the difference between using manual and automatic segmentation almost disap- pears when evaluated on multi-sentential level. In fact, the absolute difference on all metrics is less than 1% and insignificant as well. Actually, this is not a surprising finding: Most discourse constituents in an RST-style discourse parse tree conform to the sentence boundaries, in the sense that EDUs rarely span over multiple sentences. Moreover, the target discourse parser we adopt in this experiment takes a two-stage parsing strategy: in the first stage, sentences are processed to form sentence-level discourse subtrees, which in turn serve as the basic process- ing unit in the second parsing stage. Therefore, due to the nature of the RST-style discourse trees and the particular parsing algorithm in the target discourse parser, the influence of dif- ferent segmentation is very much confined within each sentence, and thus has little effect on

that, in this released version, sentence splitting is incorporated as part of the preprocessing procedure of the software. For the sake of fair comparison, to rule out the complication of different sentence splitting between their software and our own models, we modified their code to ensure all EDU segmenters are fed with the same set of sentences as input.All significance tests are performed using the Wilcoxon signed-rank test.

higher levels of the tree. Based on the analysis, the influence of segmentation on the text level is almost entirely attributed to its influence on the intra-sentential level.

## Feature Analysis

In this section, we study the effect of our pairing and global features, the two distinct charac- teristics of our two-pass segmentation model, on the overall performance, and their generality across different segmentation frameworks.

First, starting from our full model, we perform a series of feature ablation experiments. In each of these experiments, we remove one of the component features or their combinations from the feature set in training, and evaluate the performance of the resulting model.

**Removing Pairing Features (**−*p***)** By removing pairing features, our CRF-based segmenta- tion model shown in Figure [2.2](#_bookmark22) reduces to the one shown in Figure [2.3,](#_bookmark30) in which the input to each label node *Li* is a single token *Ti*, rather than a pair of adjacent tokens *Ti*−1 and *Ti*. Note that the first token *T*1 is excluded in the sequence because there always exists an EDU boundary (the beginning of the sentence) before *T*1. In accordance, the features listed in Section [2.3](#_bookmark23) now reduce to features describing each single token *Ti*.

**Removing Global Features (**−g**)** By removing global features, our two-pass segmentation model reduces to a simple one-pass model, in which only the basic and contextual features in Section [2.3](#_bookmark23) are used in training the model.

**Removing Both Features (**−*p*g**)** In this case, our model reduces to a simple one-pass model, in which only the basic and contextual features are used, and all features are based on each individual token *Ti*, rather than the token pair *Ti* and *Ti*+1.

Moreover, we wish to explore the generality of our pairing features and the two-pass strat- egy, by evaluating their effects across different segmentation frameworks. In particular, since our two-pass segmentation model itself is a CRF-based sequential labeling model, in this ex- periment, we also study the effect of removing pairing and global features in the framework of independent binary classification. Recall that in the framework of independent binary classifi- cation, each token (excluding *T*1) in a sentence is examined independently in a sequence, and a binary classifier is used to predict the label for that token.

Figure [2.4](#_bookmark33) shows our models in the framework of independent binary classification. If pairing features are enabled, as shown in Figure [2.4a,](#_bookmark33) in each classification, a pair of adjacent tokens, rather than a single token, are examined, and the classifier predicts whether an EDU boundary exists in between. If pairing features are disabled, the model reduces the one shown in Figure [2.4b.](#_bookmark33)

In this experiment, we explore two underlying classifiers in independent binary classifica- tion: Logistic Regression (LR) and a linear-kernel Support Vector Machine (SVM). We imple- ment these two classifiers using Scikit-learn [(Pedregosa et al., 2011).](#_bookmark274) For LR, all parameters are kept to their default values, while for SVM, we use auto class-weights, which are adjusted based on the distribution of different classes in the training data, to overcome the sparsity of class *B* in the dataset.

Table [2.5](#_bookmark34) demonstrates the results of our feature analysis. The first section lists the per- formance of our full model in different segmentation frameworks. As can be seen, our full models perform similarly across different frameworks, where the absolute difference in *F*1 is less than 0.2% and insignificant. This is consistent with [Hernault et al.’](#_bookmark233)s [(2010a)](#_bookmark233) finding that, when a large number of contextual features are incorporated, binary classifiers such as SVM can achieve competitive performance with CRFs. The second section lists the performance of our models with no pairing features (−*p*). For all three resulting models, CRF−*p*, LR−*p*, and SVM−*p*, the performance is significantly lower (*p* < .01) than their corresponding full model in the first section. A similar trend is observed when global features are removed (−g) in the third section. However, with respect to the underlying frameworks themselves, SVM is significantlyworse than CRF and LR (*p* < .01), while such a significant difference is not observable when pairing features are removed. Finally, when both sets of features are removed (−*p*g), as shown in the last section, the performance of our models drops drastically (from above 90% to below 85%). This suggests that the pairing and global features, which have an important effect on the performance by themselves, are even more important in their combination.

In this experiment, we demonstrate that the pairing features and the global features have in- dividual effect in improving the overall segmentation performance, and such an improvement is significant. Moreover, we observe similar effects across different frameworks, which suggests the generality of these two novel aspects of our segmentation model.

We now conduct a token-by-token error analysis to study the distributions of the errors made by our CRF-based models with different feature settings. In particular, we evaluate the labeling errors made on each token in the test set by our fully-fledged two-pass segmentation model or the models trained with pairing or global features removed. Here, we restrict our comparisons to models following the sequential labeling framework, i.e., the CRF-based models with −*p* or superscript in Table [2.5.](#_bookmark34) Once again, all tokens which are the beginning of the sentence are not included in this analysis.

The results are shown in Table [2.6.](#_bookmark36) One interesting observation is that, as demonstrated in the second section of the table, on top of CRF−g, by adding global features, our full model is able to correct the 21 errors made by CRF−g while introducing no additional errors in the process. In addition, as illustrated in the third section of the table, pairing and global features are almost complementary to each other, in the sense that the 39% of the errors made by CRF−*p* occur on cases where CRF−g is correct, and reciprocally, 32% of errors made by CRF−g happen on cases where CRF−*p* is correct.

Finally, in Figure [2.5,](#_bookmark37) we show some examples sentences, which our fully-fledged two-pass segmentation model labels correctly, while the weaker models make some errors.

## nclusion and Future Work

In this chapter, we presented our two-pass RST-style discourse segmentation model based on linear-chain CRFs. In contrast to the typical approach to EDU segmentation, which relies on token-centered features in modeling, the features in our segmenter are centered on pairs of tokens, to equally take into account the information from the previous and the following token surrounding a potential EDU boundary position. Moreover, we propose a novel two- pass segmentation strategy. After the initial pass of segmentation, we obtain a set of global features to characterize the segmentation result in a whole, which are considered in the second pass for better segmentation.

Comparing with several existing discourse segmentation models, we achieved the best per- formance on identifying both the boundaries and non-boundaries. Moreover, we studied the effect of our novel pairing and global features, and demonstrated that these two sets of features are both important to the overall performance, and such importance is observable across differ- ent segmentation frameworks and classifiers. Finally, we experimented with our segmentation model as a plug-in in our discourse parser and evaluated its influence to the overall parsing accuracy. We found that the automatic segmentation has a huge influence on the parsing ac- curacy, when evaluated on intra-sentential level; however, such an influence is very minor on multi-sentential level.

For future work, we wish to explore the incorporation of our two-pass segmentation model in the reranking framework of [Bach et al. (2012).](#_bookmark203) Since our model is shown to be a stronger base segmenter, with the reranking procedure, further improvement in segmentation accuracy may be expected.