博士論文

Unsupervised Induction of Natural Language Discourse Structure Based on Rhetorical Structure Theory

(修辞構造理論に基づく談話構造の教師なし解析)

西田　典起

指導教員：中山　英樹

東京大学　大学院情報理工学系研究科　創造情報学専攻

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Noriki Nishida
December 2019

Acknowledgements

本研究を進めるにあたり、多くのご支援とご指導を賜りました。

指導教員である中山英樹准教授には、修士1年生で研究室に入ってから現在に至るまで、研究に向かう姿勢から研究の進め方、研究の具体的な方策まで丁寧にご指導いただきました。 この5年間、どのような研究課題にでも工学、科学の視点を忘れず、本質を常に問うように研究に向かわれるその姿勢には、何度も励まされました。 また、自然言語処理、特に談話構造解析という中山研究室で取り組まれたことのない領域の課題に取り組むことにも快諾してくださいました。私がこの5年間、研究そのものの楽しさを忘れることなく、たとえうまくいかないときも辛抱強く研究を進めてくることができたのは、中山研究室で研究できたからだと思います。 心から感謝の意を表します。

宮尾祐介教授、千葉滋教授、五十嵐健夫教授、武田朗子教授、塩谷亮太准教授には、本論文作成にあたり、審査委員として多くのご助言を頂きました。 先生方の助言により、本論文の完成度を高めることができました。 深く感謝いたします。

稲葉真理教授、蜂須賀恵也准教授には、修士課程進学時から現在まで、IREF棟で有意義な研究生活を送れるよう様々に取り計らっていただきました。 深く感謝しております。

長谷川禎彦准教授には、学部学生時代、まだ学部4年生であった私に研究の楽しさ、研究者のかっこ良さを教えてくださいました。 また、卒業後も勉強会を通じて、新しいことに貪欲にチャレンジしていく姿勢を教えていただきました。 深く感謝いたします。

研究活動費においては、日本学術振興会からのご支援を頂戴しました。 深くお礼申し上げます。

最後に、これまで私をあたたかく応援してくれた両親と妹に心から感謝します。

Abstract

Natural language text is generally coherent. Discourse coherence can be represented as discourse structures, which discourse parsing aims to analyze automatically for given text. Despite the promising progress achieved in recent decades, discourse parsing still remains a significant challenge. The difficulty is due in part to the high cost and and low reliability of hand-annotated discourse structures. This thesis tackles the problems by introducing ***unsupervised discourse parsing***. Unsupervised discourse parsing is a novel technology that automatically induces discourse structures for input texts without relying on human-annotated discourse structures. Based on Rhetorical Structure Theory (RST), we assume that coherent text can be represented as a tree structure. The leaf nodes correspond to non-overlapping clause-level text spans (called elementary discourse units in RST), while the internal nodes consist of three orthogonal elements: (1) discourse constituents, (2) discourse nuclearities, and (3) discourse relations. Based on this assumption, we first break down the unsupervised discourse parsing problem into smaller subtasks, each corresponding to one of the three orthogonal elements. Then, we propose unsupervised algorithms for the three subtasks. The unsupervised algorithms are developed based on our prior knowledge of the target discourse elements. Experimental results demonstrate that our unsupervised algorithms outperform and improve unsupervised baselines. Moreover, our unsupervised algorithm induces more accurate discourse constituents than recent fully supervised parsers. We also analyze what can or cannot be captured by our unsupervised algorithms, and find that careful consideration of prior knowledge is crucial in unsupervised discourse parsing.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Natural language text is generally *coherent* (Halliday and Hasan, 1976). In coherent text, linguistic units (e.g., clauses, sentences, paragraphs) interact with each other syntactically, semantically, and pragmatically, and no text fragment is independent nor isolated. The meaning of a text as a whole is not the mere summation of the meanings of its individual parts but is computed based on the contextual relationships among the parts.

Discourse structure is a representation of discourse coherence and describes how a document is coherently organized [1] and have been discussed in various computational theories, such as Hobbs (1985), Grosz and Sidner (1986), Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides, 2003), Discourse Lexicalized Tree-Adjoining Grammar (D-LTAG) (Webber, 2004), and Discourse GraphBank (Wolf and Gibson, 2005). In discourse structures, nodes correspond to linguistic units (e.g., clauses, paragraphs), and relevant nodes are linked together by semantic and pragmatic relationships holding between them. For example, Hobbs (1985) cited the following passage from Chomsky (1975) to illustrate a discourse structure:

(1) a. I would like now to consider the so-called "innateness hypothesis,"
    b. to identify some elements in it that are or should be controversial, and
    c. to sketch some of the problems that arise as we try to resolve the controversy.
    d. Then, we may try to see what can be said about the nature and exercise of the linguistic competence that has been acquired, along with some related matters.

[1] In this thesis, we use the terms "text" and "document" interchangeably.

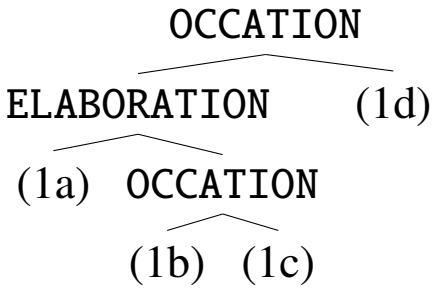


Fig. 1.1 A discourse tree structure based on Hobbs (1985) for text (1a)–(1d).

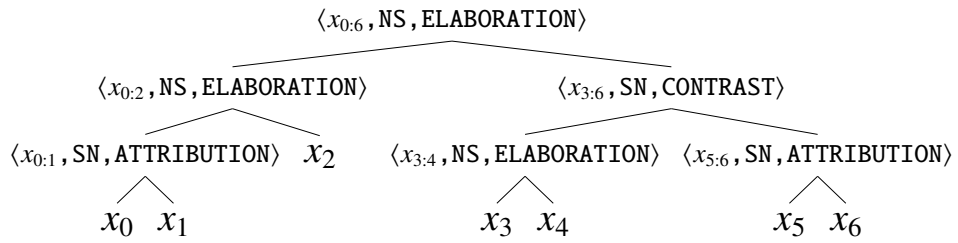
Clause (1b) and clause (1c) are connected by a "then" (OCCATION) relation. The resulting segment (1b)–(1c) *elaborates* the topic stated by clause (1a) by breaking it into two subtopics. There is also a "then" relation between sentence (1d) and sentence (1a)–(1c). Thus, the discourse structure of this text can be illustrated as in Figure 1.1.

Discourse parsing is a computational technology that aims to analyze discourse structure automatically for given text. Discourse parsing has been proven to be useful in various NLP applications, including document summarization (Marcu, 2000b; Louis et al., 2010; Yoshida et al., 2014), sentiment analysis (Polanyi and Van den Berg, 2011; Bhatia et al., 2015), document classification (Ji and Smith, 2017), automated essay scoring (Miltsakaki and Kukich, 2004), and question answering (Verberne et al., 2007; Jansen et al., 2014).

Despite the promising progress achieved in recent decades (Carlson et al., 2001; Hernault et al., 2010b; Ji and Eisenstein, 2014; Feng and Hirst, 2014; Li et al., 2014b; Joty et al., 2015; Morey et al., 2018), discourse parsing still remains a significant challenge. The difficulty is due in part to the high cost and low reliability of hand-annotated discourse structures. Manually annotating discourse structures is expensive, time-consuming, and sometimes highly ambiguous.

This thesis tackles the problems by introducing ***unsupervised discourse parsing***. Unsupervised discourse parsing is a novel technology that automatically induces discourse structures for input texts without relying on human-annotated discourse structures.

In this thesis, based on RST that is one of the most widely accepted theories of discourse structure, we assume that coherent text can be represented as tree structures, such as the one in Figure 1.2. The leaf nodes correspond to non-overlapping clause-level text spans (called EDUs in RST). Consecutive



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

Fig. 1.2 In this thesis, we assume that coherent text can be represented as a RST-like tree structure. Leaf nodes x_i correspond to clause-level segments, while internal nodes consists of three orthogonal elements: discourse constituents $x_{i:j}$, discourse nuclearities (e.g., NS), and discourse relations (e.g., ELABORATION).

text spans are combined to each other recursively in a bottom-up manner to form larger text spans (represented by internal nodes) up to a global document unit. The internal nodes consist of three orthogonal elements: (1) discourse constituents, (2) discourse nuclearities, and (3) discourse relations.

Based on this assumption, we first break down the unsupervised discourse parsing problem into smaller subtasks, each corresponding to one of the three orthogonal elements. Then, we propose unsupervised algorithms for the three subtasks.

The next natural question is how to induce each discourse element without explicit human supervision. Generally, in NLP, unsupervised algorithms are developed based on linguistic insights and prior knowledge of linguistic phenomena of interest. In this thesis, we develop unsupervised discourse parsing algorithms using our prior knowledge of the three discourse element.

Experimental results demonstrate that our unsupervised algorithms outperform and improve unsupervised baselines. Moreover, our unsupervised algorithm induces more accurate discourse constituents than recent fully supervised parsers. We also analyze what can or cannot be captured by our unsupervised algorithms, and find that careful consideration of prior knowledge is crucial in unsupervised discourse parsing.

Thesis Outline

The structure of this thesis is organized as follows:

Chapter 1 In this chapter, we introduce the background of this thesis research. First, we introduce computational theories of discourse structure that assume tree-style representations for discourse coherence like this thesis. Then, we introduce convetional discourse parsing algorithms, especially focusing on RST tree structures. After that, we introduce NLP applications where discourse parsers and discourse structures play a key role in solving the problems.

Chapter 2 In this chapter, first, we describe problems of the conventional discourse parsing approach. Then, we introduce *unsupervised discourse parsing*, a novel approach to discourse parsing. Next, we break down the unsupervised discourse parsing problem into smaller subtasks, based on which we develop a roadmap to achieve the goal. Then, we describe our core approach to developing unsupervised discourse parsing algorithms. At the end of this chapter, we review related work and emphasize the contributions of this thesis.

Chapter 3 In this chapter, we propose an unsupervised method for unlabeled discourse constituency parsing. Based on our hypothesis that the concept of constituent structure is shared between syntactic and discourse trees at a metalevel, we extend the grammar induction algorithms appropriately to unsupervised discourse constituency parsing. Experimental results demonstrate that the proposed unsupervised parser outperforms all the baselines and achieves even higher performance than supervised parsers, even though our parser does not rely on human supervisions. We also show the importance and effectiveness of the proposed initial-tree sampling methods.

Chapter 4 In this chapter, inspired by *deletion test* of Carlson and Marcu (2001), we propose discourse irreducibility measures for unsupervised discourse nuclearity classification. We incorporate extractive summarization techniques and a simple recursive algorithm to compute discourse irreducibility. Experimental results demonstrate that the proposed method outperforms clustering-based baselines. Moreover, with our proposed method, multiple complementary measures can be combined to improve performance.

Chapter 5 In this chapter, we propose an unsupervised method for implicit discourse relation classification. We use coherence modeling to learn soft discourse relationships among local text segments, which are used for (implicit) discourse relation classification. Experimental results demonstrate that the knowledge obtained by coherence modeling is effective for discourse relation classification. Additionally, we found that topic-based coherence modeling is more effective to learn discourse-relation knowledge than order-based counterpart.

Chapter 6 This chapter summarizes all research activities. From the research results, we conclude the thesis study. We also describe the limitations and future work of this study.

1.1 Computational Theories of Discourse Structure

This thesis makes the assumption that coherent text can be represented as discourse-level tree structures, such as the one in Figure 1.2. In this section, we focus on computational theories of discourse structure that assume tree-style representatios for discourse coherence like this thesis.

1.1.1 Hobbs's Theory

Discourse structures are constructed by the discourse relations joining the text segments. The existence of such discourse relations have been pointed out by several researchers (Fillmore, 1974; Grimes, 1975; Longacre, 1976; Crothers, 1979).

Hobbs (1985) focuses especially on contextual and external knowledge that humans use to make inferences for interpreting text. Hobbs gives the following example:

(2) a. John took a book from the shelf.
b. He turned to the index.

In order to recognize the discourse coherence of this text, we require knowledge that "index" of sentence (2b) is that of a book John has taken in sentence (2a).

Based on inference types, Hobbs defines four classes of discourse relations:

1. A text can be coherent if it describes coherent (or relevant) events in the real world. (OCCATION)

2. A text can be coherent if the utterances can be related to some purpose of the discourse. (EVALUATION)

3. A text can be coherent if the writer's utterance can be related to the reader's prior knowledge. (BACKGROUND and EXPLANATION)

4. A text can be coherent if an utterance is an expansion of the other segment. (PARALLEL, ELABORATION, CONTRAST, VIOLATED EXPECTATION, and EXEMPLIFICATION).

Hobbs argues that discourse relations are applied to text segments recursively to form larger segments and that one tree can span the entire text if it is well-organized written discourse as shown in Figure 1.1.

1.1.2 Grosz & Sidner's Theory

Grosz and Sidner (1986) focus solely on the writer's intention and purpose to formalize discourse structures. A text typically has an overall purpose, which is broken down into subpurposes by the writer. Then, linguistic representations are produced according to each subpurpose. This is natural when considering recursive planning paradigm for text generation (such as scientific-paper writing). Grosz and Sidner argue that such a hierarchy of subpurposes supports discourse coherence of texts.

Moreformally, they define three interacting components for their discourse structure theory: a *linguistic structure*, an *intentional structure*, and an *attentional state*.

The linguistic structure is a collection of discourse segments. A discourse segment consists of consecutive or nonconsecutive utterances, and utterances in the same discourse segment share a common purpose, called *discourse segment purpose* (DSP). The DSP represents why the writer produces the discourse segment to achieve the overall purpose.

The intentional structure is a structure of DSPs. They assume that DSPs are linked together recursively by two structural relations: DOMINANCE and SATISFACTION-PRECEDENCE. DSP2 *dominates* DSP1 if an action satisfies DSP1 is intended to provide part of the satisfaction of DSP2.

The attention state is a state of a stack of *focus spaces*. Each focus space is associated with one discourse segment and contains salient entities and

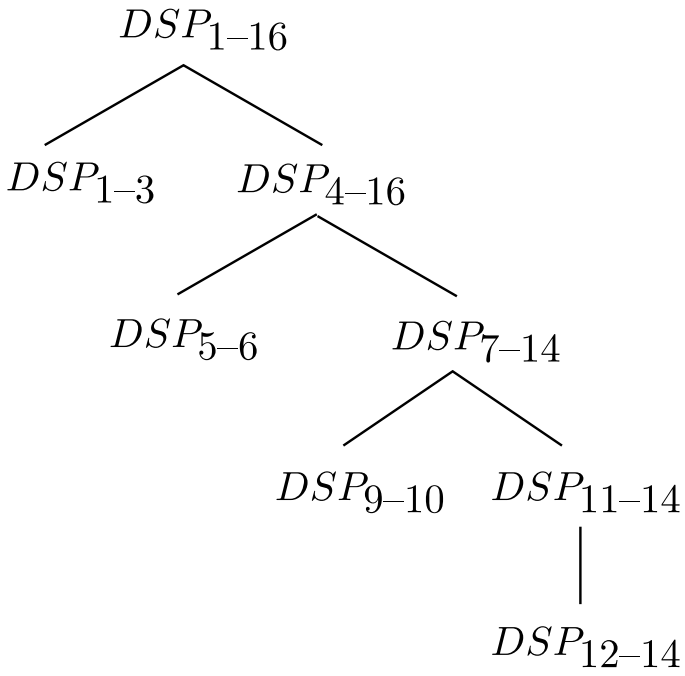


Fig. 1.3 A discourse tree structure (or a dominance hierarchy) based on Grosz and Sidner (1986).

the DSP. Entities and DSPs contained in the stack are used as contextual information to interpret utterances at each point in the text, but information in higher spaces are more salient than those in lower spaces. A new focus space is pushed to the stack when the new DSP is dominated by the immediately preceding DSP. When the new DSP is dominated by a DSP higher in the dominance hierarchy, several focus spaces are popped from the stack and then the new focus space is pushed.

Like Hobbs's theory, Grosz and Sidner considers that a tree structure can be derived for coherent text. Figure 1.3 illustrates a dominance hierarchy of a "movie essey" example used in Grosz and Sidner (1986), which consists of 16 utterances and 8 discourse segments. "$\text{DSP}_{\text{i–j}}$" denotes a DSP for the discourse segment covering from *i*-th utterance to *j*-th utterance. The crucial difference between Grosz and Sidner's theory and Hobbs's theory is that, while Hobbs (1985) focus on knowledge and inference types used in discourse interpretation, Grosz and Sidner use solely intentions of the writer to structurize discourse coherence.

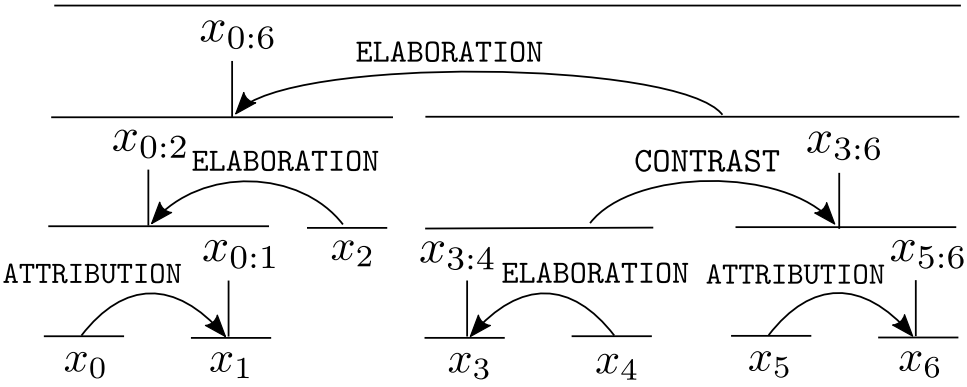


Fig. 1.4 A discourse tree structure based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) (simplified from `wsj_0642` in RST Discourse Treebank (Carlson et al., 2001)). The leaf nodes x_i correspond to contiguous, non-overlapping, minimal text spans called elementary discourse units (EDUs; typically clauses). The horizontal bars represent discourse constituents. The curves represent nuclearities, from a satellite to a nucleus. The curves are also labeled with rhetorical relations.

1.1.3 Rhetorical Structure Theory

Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) is one of the most widely accepted theories of discourse structure. According to RST, a coherent text is composed of several functional units, which are further divided into smaller ones, resulting in a tree structure as in Figure 1.4. The leaf nodes correspond to contiguous and non-overlapping text spans that represent minimal discourse units called *elementary discourse units* (EDUs), which are typically clauses. Consecutive text spans are combined recursively to form larger text spans (represented by internal nodes) up to the global document span. The internal nodes hold information about (1) *discourse constituents*, (2) *nuclearities*, and (3) *rhetorical relations*. Discourse constituents are contiguous text spans covered by the nodes, which represent functional units of discourses. Nuclearities are directional labels that represent relative importance between connected text spans – a *nucleus* corresponds to a more essential span, while a *satellite* corresponds to a supporting or background one. Rhetorical relations are informational or intentional relationships holding between connected text spans, such as `CONDITION` and `MOTIVATION`.

RST-based text analysis has been conducted using RST Discourse Treebank (RST-DT) (Carlson et al., 2001), which contains hand-annotated discourse structures for 385 English news articles from the Wall Street Journal.

This thesis assumes that RST-based tree structures can be derived for coherence text. Therefore, in the following, we provide detailed descriptions about characteristics of central components of RST: elementary discourse units (EDUs), discourse constituents, nuclearities, and rhetorical relations.

Elementary Discourse Units

Elementary discourse units (EDUs) are contiguous, non-overlapping, and minimal functional spans. Carlson et al. (2001) choose clauses as EDUs and uses lexical and syntactic cues to determine EDU boundaries:

(3) [Such trappings suggest a glorious past] [**but** give no hint of a troubled present.]$_{\texttt{wsj\_1302}}$

(4) [Previously, airlines were limiting the programs] [**because** they were becoming too expensive.]$_{\texttt{wsj\_1192}}$

(5) [**Although** Mr. Freeman is retiring,] [he will continue to work as a consultant for American Express on a project basis.]$_{\texttt{wsj\_1317}}$

EDUs like participial clauses may not contain a lexical cue:

(6) [Xerox Corp.'s third-quarter net income grew 6.2% on 7.3% higher revenue,] [**earning** mixed reviews from Wall Street analysts.]$_{\texttt{wsj\_1109}}$

EDUs need not be syntactic clauses. RST lists the following exceptions:

- Clauses that are subjects or objects of main verbs are not EDUs:

(7) [**Making computers smaller** often means sacrificing memory.]$_{\texttt{wsj\_2387}}$

(8) [Atco Ltd. said its utilities arm is considering **building new electric power plants,** ...]$_{\texttt{wsj\_2309}}$

- Clauses that are complements of main verbs are not EDUs:

(9) [Ideally, we'd like **to be the operator {of the project} and a modest equity investor**.]$_{\texttt{wsj\_2309}}$

- Clauses that are complements of attribution verbs (e.g., *say*, *announce*, *declare*, *suggest*, *report*, etc.) are EDUs:

(10) [Mercedes officials **said**] [they expect flat sales next year] [even though they see the U.S. luxury-car market expanded slightly.]$_{\texttt{wsj\_1196}}$

- Relative clauses containing a verbal element, post nominal modifiers containing a non-finite verb, or clauses breaking up other EDUs are embedded EDUs:

(11) [So far they have issued scores of subpoenas,] [**some of which went to members of the New York Merc.**]$_{\texttt{wsj\_[0664]}}$

(12) [According to one dealer,] [Japan said] [it has only 40,000 tons of sugar remaining] [**to be shipped** to it this year by Cuba under current commitments.]$_{\texttt{wsj\_1932}}$

(13) [The Tass news agency said the 1990 budget anticipates income of 429.9 billion rubles] [**($ US693.4 billion)**] [and expenditures of 489.9 billion rubles] [**($ US790.2 billion)**]$_{\texttt{wsj\_0311}}$

- Phrases begining with strong discourse markers (e.g., "because", "in spite of", "as a result of", "according to") are allowed as EDUs.

(14) [**Despite** the yen's weakness with respect to the mark,] [Tokyo traders say] [they don't expect the Bank of Japan to take any action...]$_{\texttt{wsj\_1102}}$

Hierarchy of Discourse Constituents

Discourse constituents are contiguous text spans that represent functional units of a discourse. Sizes of discourse constituents are arbitrary. As shown in Figure 1.4, consective discourse constituents (including EDUs) are connected to form larger discourse constituents recursively in a bottom-up manner, which also indicate functional spans at higher levels. Thus, a RST tree can be modeled as a set of discourse constituents of different granularity. A global tree structure is finally built for the whole text.

Moreformally, RST defines *schemas* that are abstract patterns (or constraints) to combine a small number of constituent text spans. The schemas can be interpreted as discourse-level rules that specfy how constituents are combined, which are loosely analogous to syntactic grammars. The schemas are applied recursively to EDUs, resulting in one RST tree for a text.

Specifically, RST defines five schemas, shown in Figure 1.5. Note that the order of nucleus and satellites is not constrained in the application of schemas.

Such a hierarchy of discourse constituents provides structural information of text, which is complementary to directional information from nuclearities and semantic sense information from rhetorical relations. Actually, the

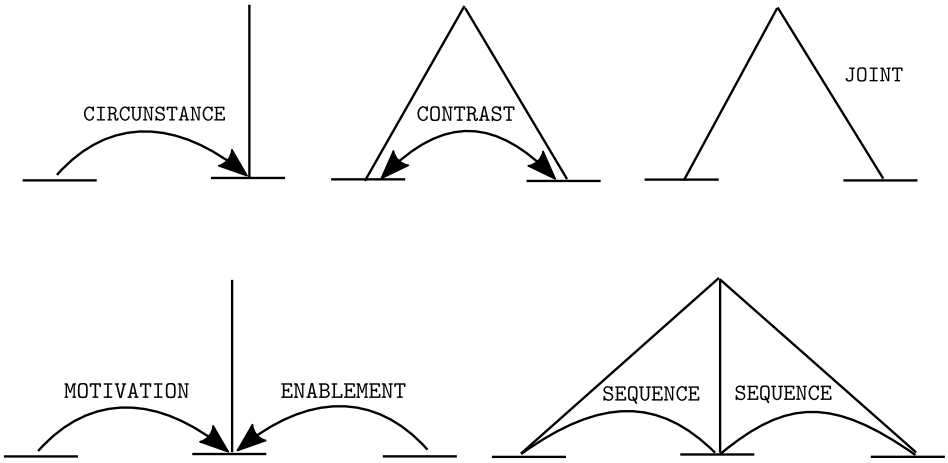


Fig. 1.5 Five *schemas* defined by Mann and Thompson (1988).

distance (or depth) of discourse constituents from the root node can be used as indicators of relative importance of the corresponding text spans in the text (Louis et al., 2010). Text spans closer to the root node take on a more global role, while text spans of deeper depth tend to provide more detailed information.

Nuclearity – Nucleus and Satellite

Not all discourse constituents in a text span are equally important. Some constituents represent more essential information of the text span, while some constituents indicate supporting information of others, such as refinement, background, or conditioning. Nuclearity is relative importance between combined constituents – A nucleus is the salient constituent, while a satellite is the supporting constituent. When text spans of the same importance are combined, all the spans are nuclei.

Then, how can nuclearity be determined? Generally, nuclearity can not be determined in isolation. Consider the following two examples in Carlson et al. (2001):

(15) [The earnings were fine and above expectations ...] [Nevertheless, Salomon's stock fell $1.125 yesterday ...]$_{\texttt{wsj_1124}}$

(16) [*Although the earnings were fine and above expectations,*] [Salomon's stock fell $1.125 yesterday.]

In the first example (15), both spans are nuclei. However, in the second example (16), although the semantic content is very similar to example (15),

Circumstance	Antithesis and Concession
Solutionhood	Antithesis
Elaboration	Concession
Background	Condition and Otherwise
Enablement and Motivation	Condition
Enablement	Otherwise
Motivation	Interpretation and Evaluation
Evidence and Justify	Interpretation
Evidence	Evaluation
Justify	Restatement and Summary
Relations of Cause	Restatement
Volitional Cause	Summary
Non-Volitional Cause	Other Relations
Volitional Result	Sequence
Non-Volitional Result	Contrast
Purpose	

Table 1.1 Rhetorical relations defined by Mann and Thompson (1988).

the first span is a satellite and the second one is a nucleus. These examples indicate that nuclearity depends on the context, use of discourse markers, etc.

In general, removing nuclear constituents leads to difficulty for readers in interpreting the text. In contrast, satellite constituents can be removed more easily than nuclei they relate to. Based on this observation, Carlson et al. (2001) introduced the following two tests to distinguish nuclei and satellites:

- *Deletion test*: When a satellite is deleted, the nucleus it relate to can still perform the same function in the text, although it may be somewhat weaker. When the nucleus is deleted, the segment that is left is much less coherent.
- *Replacement test*: A satellite can be replaced with a different content without altering the function of the segment.

Rhetorical Relations

Originally, Mann and Thompson (1988) define 23 types of rhetorical relations listed in Table 1.3. The rhetorical relations are defined based on the following four elements:

- Constraints on Nucleus.
- Constraints on Satellite.

CONCESSION	
Constraints on N:	Writer has positive regard for N.
Constraints on S:	Writer is not claiming that S doesn't hold.
Constraints on N + S:	Writer acknowledges a potential or apparent incompatibility between N and S; Writer regards N and S as compatible; recognizing the compatibility between N and S increases Reader's positive regard for N.
Effect:	Reader's positive regard for N is increased.
PURPOSE	
Constraints on N:	N presents an activity.
Constraints on S:	S presents a situation that is unrealized.
Constraints on N + S:	S presents a situation to be realized through the activity in N.
Effect:	Reader recognizes that the activity in N is initiated in order to realize S.

Table 1.2 Definitions of the CONCESSION and PURPOSE relations in RST, with respect to the four definition elements.

- Constraints on Nucleus + Satellite combination.
- Effect.

These elements are judged by the text analyst solely on functional and semantic information of the discourse. Assignment of rhetorical relations do not depend on morphological nor syntactic cues, such as "if" for Condition. These are judgements of plausibility rather than certainty. Table 1.2 illustrates the definitions of CONCESSION and PURPOSE in RST.

The RST Discourse Treebank (RST-DT) corpus (Carlson et al., 2001) devides the original rhetorical relatios of Mann and Thompson (1988) into 112 fine-grained relations. The 112 relations are further categorized into 18 coarse-grained classes, which are currently broadly used in RST parsing studies. We show the coarse-grained relation classes in Table 1.3.

RST includes rhetorical relations of two types: subject-matter (i.e., informational) relations and presentational (i.e., informational) relations (Grosz and Sidner, 1986). As pointed out by Mann and Thompson (1988) and Moore and Pollack (1992), text spans can relate to each other *simultaneously* at both informational and intentional levels:

(17) a. George Bush supports big business.
b. He's sure to veto House Bill 1711.

ATTRIBUTION	EXPLANATION
BACKGROUND	JOINT
CAUSE	MANNER-MEANS
COMPARISON	TOPIC-COMMENT
CONDITION	SUMMARY
CONTRAST	TEMPORAL
ELABORATION	TOPIC-CHANGE
ENABLEMENT	TEXTUAL-ORGANIZATION
EVALUATION	SAME-UNIT

Table 1.3 Rhetorical relation classes defined in RST-DT (Carlson et al., 2001).

One can recognize `VOLITIONAL CAUSE`, an informational relation, between sentence (17a) and sentence (17b). Simultaneously, `EVIDENCE`, an intentional relation, can be recognized between the sentences.

However, RST presumes that only one rhetorical relation can hold for any two text spans. To alleviate this issue, Moore and Pollack (1992) argue that analyses at both informational and intentional levels must coexist.

1.1.4 Segmented Discourse Representation Theory

Truth-conditional semantics in formal semantics aims to assign a truth condition to each sentence, which represents a meaning of the sentence. Conventionally, truth conditions are determined for each sentence in isolation of the context. However, sentence meanings are affected by the context, and vice versa. Such dynamics can not be captured by the static truth-conditional semantics.

To mitigate this issue, Kamp and Reyle (1993) introduce Discourse Representation Theory (DRT), which considers that meanings of utterances are projections from context to context. DRT maintains *discourse representation structure* (DRS) and updates it dynamically by processing consecutive utterances one by one.

Although DRT can capture each utterance dynamically, DRT can not capture relationships between utterances. Generally, utterances are not arranged linearly but structurized. Such a hierarchical structure affects meanings of each utterance.

Asher and Lascarides (2003) propose Segmented Discourse Representation Theory (SDRT), which introduces rhetorical relations into DRT. As

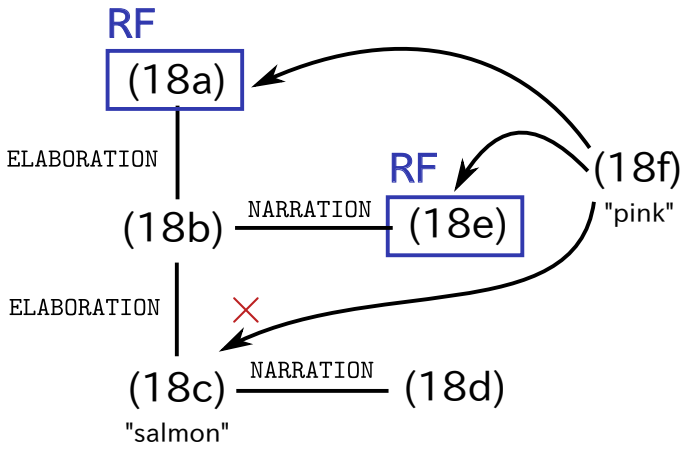


Fig. 1.6 A discourse tree structure based on SDRS (Asher and Lascarides, 2003) for discourse (18).

DRT, SDRT maintains *segmented discourse representation structure* (SDRS) and constructs a discourse tree structure dynamically by processing utterances one by one. Given a new utterance, SDRT first identifies a set of available attachment nodes (i.e., utterances) in the current SDRS to which the new utterance can attach to. The set of available attachment nodes is called the *Right Frontier* (RF) and is determined based on the *Right Frontier Constraint* (RFC) (Asher and Lascarides, 2003): a new utterance must attach to either the immediately previous utterance or one of the nodes that dominate this last node. RFC implies that the resulting SDRS forms a tree structure. Then, SDRT identifies a rhetorical relation between the connected nodes. SDRT distingushes subordinating and coordinating relations: subordinating relations (e.g., `ELABORATION`) expand the structure vertically, while coodinating relations (e.g., `NARRATION`) expands the structure horizontally. Finally, SDRT calculates the propositional information using the rhetorical relation. Figure 1.6 shows a SDRS for the following example used in Asher and Lascarides (2003):

(18) a. Max had a great evening last night.
b. He had a great meal.
c. He ate salmon.
d. He devoured lots of cheese.
e. He then won a dancing competition.
f. ??It was a beautiful pink.

Based on RFC and Figure 1.6, utterance (18f) can be attached to utterance (18e) or utterance (18a). SDRS allows us to explain why "it" in (18f) can not refer to "salmon" in (18c), because utterance (18c) is not on the RF for (18f).

1.1.5 Discourse Lexicalized Tree-Adjoining Grammar

Webber (2004) extend lexicalized Tree-Adjoining Grammar (LTAG) (Joshi and Schabes, 1997) for low-level discourse, called Discourse Lexicalized Tree-Adjoining Grammar (D-LTAG). They focus on how low-level discourse relations and structures are grounded by lexico-syntactic elements in an integrated way with syntactic grammars. Specifically, they focus on how discourse connectives such as "on the one hand" anchor the connection of two unmarked adjacent clauses. They define discourse-level elementary tree structures (i.e., *initial trees* and *auxiliary trees*) that contain discourse connectives, *substitution sites*, and *adjunction sites* at their leaves. They also employ the same operations (i.e., *substitution* and *adjoining*) as LTAG to derive shallow discourse structures with these elementary trees. The resulting structures form low-level trees.

Penn Discourse Treebank (PDTB) (Prasad et al., 2008), which is the largest annotated corpus for *shallow discourse parsing*, was constructed based on D-LTAG. PDTB contains argument pairs labeled with rhetorical relations. Some argument pairs contain discourse connectives explicitly whose relations are called explicit discourse relations. When connectives do not appear but can be inserted by humans (or annotators), those relations are called implicit discourse relations.

1.2 Parsing Algorithms of Discourse Structure

We have already discussed the concept of discourse structure. As presented in Section 1.3, discourse parsers have been applied to a variety of NLP applications. The performance in the downstream tasks strongly rely on the accuracy of the discourse parser they use. Thus, it is crucial to develop an appropriate discourse parser.

In this section, we especially focus on existing algorithms for RST discourse parsing. Conventional algorithms for RST discourse parsing can be roughly categorized into three approaches in terms of how to build a tree for given EDUs. The first category is a greedy parsing (Hernault et al., 2010b; Feng and Hirst, 2014), which builds a tree structure iteratively and greed-

ily in a bottom-up manner. The second one is a chart parsing (Joty et al., 2013; Li et al., 2014a), which finds the globally optimal tree using dynamic programming such as a Cocke-Kasami-Younger (CKY) algorithm. The last one is a Shift-Reduce parsing (Sagae, 2009; Feng and Hirst, 2012; Ji and Eisenstein, 2014), which iteratively constructs a tree structure using a stack and a buffer.

1.2.1 Greedy Parsing

Hernault et al. (2010b) proposed a HILDA system. This discourse parser builds a RST tree structure iteratively and greedily in a bottom-up manner. Starting from a list of EDUs, in each step, a binary SVM classifier is used to determine which adjacent discourse units are combined. Adjacent discourse units with the maximum probability predicted by this binary SVM classifier is selected greedily. Then, using another multi-class SVM classifier, a rhetorical relation label is assigned to the selected discourse units. This process is repeated until a discourse structure that covers the entire document is constructed. Feng and Hirst (2012) showed that the HILDA accuracy is improved by incorporating richer features.

Feng and Hirst (2014) model the tree structure and relations with Conditional Random Fields (CRFs), respectively, so that contextual information can be considered more effectively than HILDA's SVM classifiers.

Strenthes and Weaknesses The strength of the greedy algorithm is its simplicity and low computational cost. The amount of computation is linear with respect to the input document length (i.e., the number of EDUs). On the other hand, the weakness of this algorithm is that its performance tends to be poor due to the local search. The tree structure is built by a series of local and greedy decisions and can not be produced based on the more global scores.

1.2.2 Chart Parsing

Chart parsing has been traditionally used to analyze syntactic tree structures for given words. chart parsing uses dynamic programming such as Cocke-Kasami-Younger (CKY) algorithm to find the globally optimal tree structure over all valid trees. To avoid doing redundant computation over and over

again, CKY maintains a *chart table* to store partial results of the computation and reuse it.

The chart table C is a $n \times n \times L$ tensor (or matrix), the cell of which $C[i, j, l]$ holds the maximum score of the subtree that spans (i, j) and is labeled with $l \in L$. L denotes a set of valid labels for nodes. The (sub)tree score is generally factorized into the scores of the child nodes and composition scores to combine the children into the parent node:

$$C[i, j, l] = \max_{\substack{i \leq k < j, \\ (l_1, l_2) \in L \times L}} \text{Score}(x^{(l)}_{i:j} \leftarrow x^{(l_1)}_{i:k}\ x^{(l_2)}_{k+1:j}) + C[i, k, l_1] + C[k+1, j, l_2]. \tag{1.1}$$

Backpointer is also filled simultaneously with C:

$$B[i, j, l] = \operatorname*{argmax}_{\substack{i \leq k < j, \\ (l_1, l_2) \in L \times L}} \text{Score}(x^{(l)}_{i:j} \leftarrow x^{(l_1)}_{i:k}\ x^{(l_2)}_{k+1:j}) + C[i, k, l_1] + C[k+1, j, l_2]. \tag{1.2}$$

Since $C[i, k, l_1]$ and $C[k+1, j, l_2]$ have already been computed, they can be reused to compute $C[i, j, l]$ efficiently.

To parse the full input, we first compute $C[0, n-1, l]$ in a bottom-up manner, and then traverse the backpointer from $B[0, n-1, l]$.

There are generally no significant differences across chart parsers in terms of CKY dynamic programming, but the features and models used to compute the tree score can vary greatly. Joty et al. (2013) and Joty et al. (2015) use CRF to model and capture structures and relations simultaneously. Li et al. (2014a) define the composition score using Recursive Neural Networks. Li et al. (2016) improve parsing accuracy by incorporating Attention Mechanism in feature learning.

Strenthes and Weaknesses The strength of chart parsing is that it allows to find the globally optimal tree structures over all the valid trees. The weakness of chart parsing is that its parsing speed is slower than the other approaches. The computational cost is $O(n^3 L^2)$. This can be problematic in discourse parsing, where input length (i.e., the number of EDUs) n tends to be longer than sentence length. Therefore, discourse parsing algorithms using chart-based approach (Joty et al., 2013, 2015; Nishida and Nakayama, to appear) conventionally employ incremental approaches, where a tree

structure is built from smaller discourse chunks (i.e., sentences) to the global document.

1.2.3 Shift-Reduce Parsing

A shift-reduce dependency parsing (Nivre, 2004) is one of the mainstream approaches to syntactic dependency parsing as with the graph-based syntactic dependency parsing (McDonald et al., 2005). Shift-reduce parsing for syntactic constituent structures was also proposed by Sagae and Lavie (2005).

A shift-reduce discourse parser constructs a tree structure by sequentially performing actions on a stack of subtrees and a queue of EDUs. The stack maintains subtrees (including EDUs) being analyzed and is empty at the initial step. The queue contains unprocessed EDUs from left to right. Four actions are defined. SHIFT pushes the top EDU of the queue onto the stack. REDUCE pops the top two elements on the stack and combines them to create a new subtree, and re-pushes it onto the stack. The REDUCE operation must determine the discourse relation and nuclearity status. Thus, the REDUCE action is further divided into: (1) REDUCE-UNARY-*l*, which pops the first element of the stack, gives it a label *l*, and pushes it; (2) REDUCE-LEFT/RIGHT-*l*, which pops the two top elements of the stack, combines them, assign a label *l* to them, and pushes the new subtree onto the stack. Here, LEFT and RIGHT indicate the position of the nucleus subtree and corresponds to nuclearity statuses. These processes are repeated until the queue becomes empty and the stack has only one element. The last element in the stack is the analysis result.

Shift-reduce parsing can be viewed as a classification problem over possible actions given the current situations (i.e., stack and queue). Marcu (1999) uses decision trees as classifiers. Sagae (2009) uses the averaged perceptron classifier. Ji and Eisenstein (2014) use a neural network. Wang et al. (2017) also introduce an attention mechanism.

Strenthes and Weaknesses The advantage of shift-reduce parsing is their computational efficiency. While the computational complexity of chart parsing is cubic, the computational complexity of shift-reduce parsing is linear with respect to input length (i.e., the number of EDUs). This is suitable for discourse parsing where document length tends to be long. On the other hand, the disadvantage of shift-reduce parsing is its locality of the search

procedure. Actions are selected based on a classification score computed based on local features from the stack and the queue. To alleviate this locality issue, beam search is conventionally utilized to obtain candidate trees.

1.3 Applications of Discourse Structure

So far, we have discussed the computational theories of discourse structure and conventional discourse parsing algorithms. Now we focus on NLP applications where discourse parsers and discourse structures play a key role in solving the problems. Discourse parsers and structures have been proven to be useful in various downstream tasks, such as document summarization (Teufel and Moens, 2001; Louis et al., 2010; Hirao et al., 2013; Yoshida et al., 2014), sentiment analysis (Polanyi and Van den Berg, 2011; Bhatia et al., 2015), document categorization (Ji and Smith, 2017), automated essay scoring (Miltsakaki and Kukich, 2004), question answering (Verberne et al., 2007; Jansen et al., 2014), and text generation (Reiter and Dale, 1997; Hendriks, 2002). In this section, we describe three typical applications: summarization, sentiment analysis, and text generation.

1.3.1 Summarization

Text summarization is a task of summarizing single or multiple documents (e.g., news articles, scientific papers) into a relatively shorter text. Text summarization is one of the earliest NLP tasks discourse structures were applied to. Discourse structures that describe hierarchies, head-modifier relations, and rhetorical relations over sentences are useful to capture essential information in multi-sentential texts.

Teufel and Moens (2001) consider that the content of a scientific paper could be divided into 7 categories (called *rhetorical status*): BACKGROUND (description of the scientific background), AIM (the research goal), TEXTUAL (outline of the paper), OWN (methodology, results, discussion), CONTRAST (comparison with other work), BASIC (agreement with other work), and OTHER (neutral description of other work). They classify each sentence of a given scientific article into one of these categories. Consecutive sentences with the same rhetorical status are grouped into *rhetorical zones*. Thus, their method can be interpreted as discourse segmentation, and the system output is a flat discourse structure (or sequence

of discourse chunks). Based on the discourse segmentation results, they perform summarization.

Louis et al. (2010) investigated usefulness of structural features and semantic features of discourse. The structural features are computed by tree or graph representations of discourse structures. The semantic features are computed by rhetorical relations between text units. They found that structural information of discourse provides more robust indicators of sentence importance than the semantic sense information of rhetorical relations. However, it was also demonstrated that structural information and semantic information are complementary to each other and can be combined to improve performance. They also confirmed that discourse features are complementary to non-discourse features, which implies that discourse structures are beneficial to recognize essential meaning of text.

Hirao et al. (2013) utilize RST discourse structures for summarization. They first extract a RST tree structure from a given text, and then transform the tree into a dependency graph to obtain parent-child relationships between EDUs. Finally, the discourse-level dependency graph is used to solve a Tree Knapsack Problem to perform summarization. One weakness of this approach is that it relies on the accuracy of the discourse parser. Yoshida et al. (2014) propose a discourse parser that directly analyzes RST discourse dependency graphs for discourse-based summarization. They demonstrate that the directly analyzed discourse dependency graph improves the summarization performance.

1.3.2 Sentiment Analysis

Semantiment analysis is a task of identifying sentiment polarity (positive or negative) of input text (e.g., product reviews). Document-level sentiment polarity can not be determined in isolation of context. For example, as illustrated by Voll and Taboada (2007), the sentiment polarity of the following review of the movie *The Last Samurai* is negative, although the positive sentiment words outnumber the negative sentiment words:

(19) a. It could have been a great movie.
b. It does have beautiful scenary,
c. some of the best since Lord of the Rings.
d. The acting is well done,
e. and I really liked the son of the leader of the Samurai.

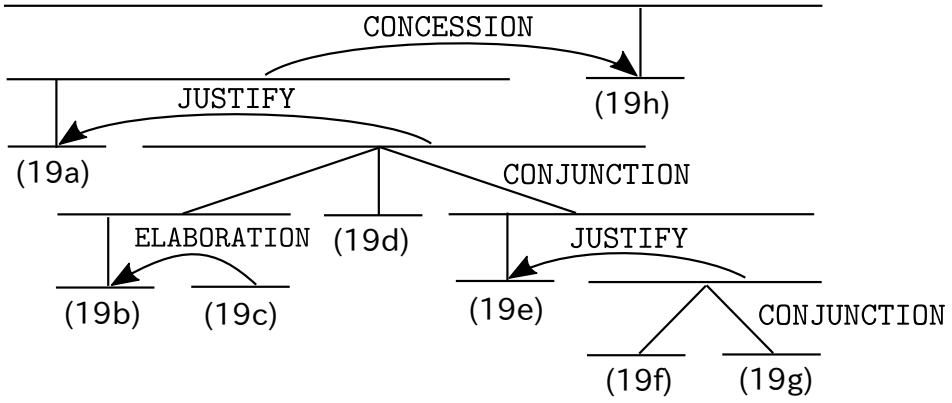


Fig. 1.7 A RST discourse structure for the movie review (19). In this example, sentence (19h) is treated as the most important part, because it is the closest node in the tree structure.

f. He was a likable chap,

g. and I hated to see him die.

h. But, other than all that, this movie is nothing more than hidden rip-offs.

Bhatia et al. (2015) propose to incorporate RST discourse structures as shown in Figure 1.7 for document-level sentiment analysis. Specifically, they assume that sentences closer to the root node in the RST trees are more important. They assign importance scores to each sentence based on the depth in the tree, and then the document-level sentiment polarity is computed by the weighted sum of sentences' polarity. They confirm the usefulness of hierarchical information of discourse structures in document-level sentiment analysis. Somasundaran et al. (2009) also confirms that the polarity classification accuracy can be increased by integrating contextual information from discourse relations into word-based polarity-classification methods.

1.3.3 Text Generation

Discourse is more than just the sum of the meanings of the sentences. The meaning of the entire document is computed in reader's mind dynamically based on the discourse structure and knowledge we share. Therefore, in order to (1) convey information as intended by the writer to readers accurately and (2) improve the readability of the text, we need to pay attention not only to the superficial linguistic representations, but also to the overall text structure.

Text generation is a technology for automatically generating natural language text by a computer. Text generation is also used as a supporting system to help people write more coherent text.

The initial motivation of discourse structure theory is its application to document-level text generation. For example, Reiter and Dale (1997) propose a planning approach based on discourse structure to text generation. After determing a document template using discourse structure, their method then fills the template with concrete linguistic representations. Hendriks (2002) also uses DRT to propose a planning-based text generation system.

Chapter 2

Unsupervised Discourse Parsing: An Overview

In this chapter, we introduce *unsupervised discourse parsing*. Unsupervised discourse parsing is a novel approach that aims to alleviate the high-cost and low-reliability problems of supervised discourse parsing. In Section 2.1, we first raise problems and limitations of the conventional discourse parsing. In Section 2.2, we describe the goal and assumption of this thesis. In Section 2.3, we break down the unsupervised discourse parsing problem into smaller subtasks and build a roadmap towards unsupervised discourse parsing. In Section 2.4, we describe how we develop the unsupervised algorithms. Finally, in Section 2.5, we review related works and emphasize the contributions of this thesis with respect to them.

2.1 Problem Statements

Existing discourse parsing algorithms, such as the ones described in Section 1.2, use supervised learning to develop a parser and rely on hand-annotated discourse structures. Parameters of such supervised discourse parsers are updated so as to minimize the *distance* between gold discourse structures (i.e., human supervision) and the parser outputs.

Although supervised learning is an effective approach to various NLP tasks such as machine translation, there are two problems in supervised discourse parsing, each related to (1) annotation cost and (2) annotation reliability.

High cost of annotation There are a theoretically infinite number of discourse patterns in the real world. This means that, to parse such a variety of discourse patterns accurately, we require a massive amount of hand-annotated discourse structures, from which parsing models can learn essential knowledge of discourse coherence and structures. However, manually annotating discourse structures is significantly expensive and time-consuming. To construct a large-scale discourse treebank, it is necessary to hire a number of expert annotators who are highly trained on the discourse structure theory such as RST.

Low reliability of annotation Another problem is relevant with the reliability of hand-annotated discourse structures. As presented in Section 1.1, there is still no single discourse structure theory with consensus. For example, the number of discourse relations and their granularity each theory assumes vary greatly: Grosz and Sidner (1986) assumes only 2 discourse relations, while Hovy and Maier (1995) lists more than 400 relations. Additionally, Moore and Pollack (1992) pointed out the importance to analyze discourse structures simultaneously at both information and intentional levels, although RST presumes only one analysis for given text. Such wide arbitrariness of discourse structures is due to the ambiguity of what we focus on to describe/explain discourse coherence, and such ambiguity of discourse structures leads to low reliability of hand-annotated discourse structures. Actually, Marcu (2000a) reported that the inter-annotator agreement scores is only around 83%. Carlson et al. (2001) also reported that inter-annotator agreement scores on RST-DT are: 88.70% for unlabeled discourse constituency parsing, 77.72% for nuclearity assignments, and 65.75% for rhetorical relation labeling. Such low reliability of human supervision can be a critical issue for supervised discourse parsing.

2.2 Aim of the Thesis

To mitigate the problems raised in Section 2.1, we introduce unsupervised discourse parsing. Unsupervised discourse parsing aims to induce discourse structures for given text without relying on hand-annotated discourse structures. Unsupervised discourse parsing does not suffer from the high-cost and low-reliability problems of annotated discourse structures.

Based on Rhetorical Structure Theory (Mann and Thompson, 1988), this thesis assumes that coherent text can be represented as **tree structures**, such

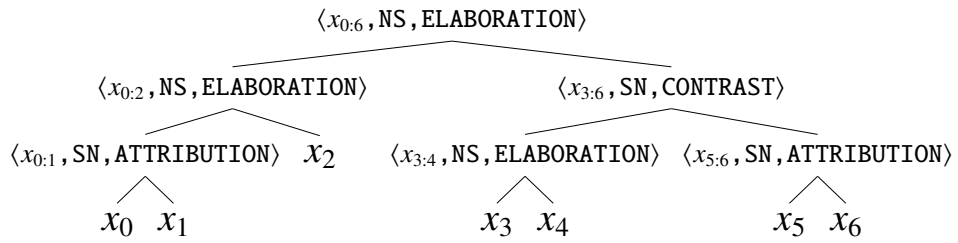


Fig. 2.1 A RST-based discourse tree structure we assume in this thesis.

as the one in Figure 2.1. The leaf nodes correspond to non-overlapping clause-level text spans (called EDUs in RST). Consecutive text spans are combined recursively in a bottom-up manner to form larger text spans (represented by internal nodes) up to the global document span. The internal nodes consist of three orthogonal elements:

(1) ***discourse constituents*** that are contiguous text spans working as functional units in the discourse;

(2) ***discourse nuclearities*** that specify relative importance between connected text spans (constituents);

(3) ***discourse relations*** that are semantic relationships holding between text spans (constituents).

For instance, an internal node $\langle x_{i:j}, \texttt{SN}, \texttt{CONTRAST} \rangle$ indicates that the node corresponds to a discourse constituent $x_{i:j}$, the nuclearity is `SN` (i.e., the right child span is more salient than the left child span), and a `CONTRAST` relation holds between the child spans.

There are two additional motivations to investigate unsupervised discourse parsing. First, unsupervised discourse parsing can provide suggestions on how to utilize raw data effectively in a semi-supervised setting. A semi-supervised setting is a natural choice in the real world where we generally have a small amount of annotated data and a large amount of raw data (e.g., web pages). How to use raw data is significant important and of central interest in a semi-supervised setting. The second advantage is that it can also provide suggestions on what discourse-structural patterns should be annotated more heavily (lightly). Some discourse elements are expected to be easier to induce in an unsupervised manner, while other elements may be much more difficult to induce without explicit human supervision. Therefore, by comparing the outputs of supervised and unsupervised algorithms, we

can recognize what kind of discourse-structural patterns should be annotated more (less). This allows us to efficiently collect effective human annotations and reduce the size of annotated training data.

2.3 Building a Roadmap

It is almost always a reasonable strategy to break down a complicated problem into simpler subproblems. Each subproblem is expected to be sufficiently small so that they can be handled realistically.

In this thesis, we follow this strategy and break down the unsupervised discourse parsing problem into smaller subtasks. As described in the previous section, discourse tree structures we assume in this thesis are composed of three **orthogonal** elements: (1) discourse constituents, (2) discourse nuclearities, and (3) discourse relations. Actually, it is impossible to reconstruct each discourse element from the other two elements, and each of these elements convey important information of the discourse. Discourse constituents focus on a hierarchical structure, while discourse nuclearities represent directional information between nodes, which can not be reconstructed from the other elements. Discourse nuclearities are rather structural than semantic and can be combined with discourse constituents to build unlabeled discourse dependency structures (Li et al., 2014b; Morey et al., 2018). Discourse relations represent more semantic information than the others. Please note that discourse relation classes (e.g., `CONTRAST`, `MOTIVATION`) do not contain directional information.

Thus, it is natural to divide the overall parsing process according to these elements into four subtasks:

(0) EDU segmentation;

(1) unlabeled discourse constituency parsing;

(2) discourse nuclearity classification;

(3) discourse relation classification.

Step (0) is the preliminary task to prepare leaf nodes for tree building. Step (1), Step (2), and Step (3) correspond to one of the three orthogonal elements in discourse tree structures. Of course, several subtasks can be tackled simultaneously. However, fusing elementary tasks (elements) into more

complicated tasks (elements) increases the informational complexity that models must capture, which is contrary to the problem-dividing strategy.

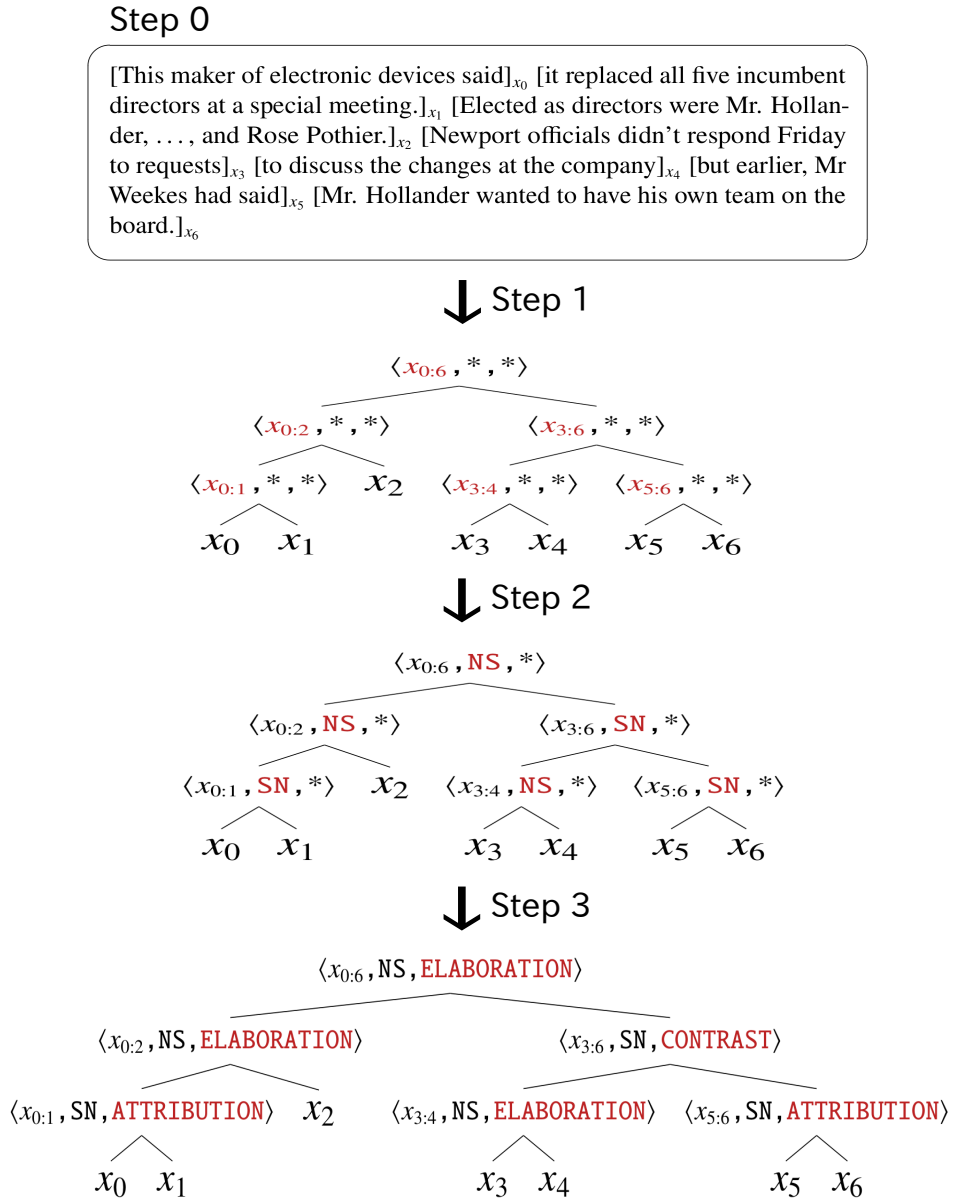


Fig. 2.2 Our roadmap towards unsupervised discourse parsing. $x_0, \ldots, x_6$ denote EDUs of input text. Based on this roadmap, we propose unsupervised algorithms for each subtask: (1) unlabeled discourse constituency parsing (2) discourse nuclearity classification, and (3) discourse relation classification.

Based on this factorization, we aim to achieve the original goal (i.e., unsupervised discourse parsing) by proposing unsupervised algorithms for each subtask. We illustrate our roadmap in Figure 2.2. EDU boundaries in Step (0) are relatively easier to annotate, compared to other subtasks. Thus,

in this thesis, we focus especially on unsupervised algorithms for the latter three parts. We describe our unsupervised algorithms for the three subtasks in turn in Chapter 3, Chapter 4, and Chapter 5.

2.4 Using Prior Knowledge in Unsupervised Algorithms

The next natural question is how to induce discourse constituents, discourse nuclearities, and discourse relations without explicit human supervision. It seems significantly difficult to induce such discourse elements without human supervision. Generally, instead of human supervision, linguistic insights and prior knowledge of linguistic phenomena of interest are used to develop unsupervised algorithms in NLP. For instance, the skip-gram model (Mikolov et al., 2013a), one of the most popular unsupervised algorithms for word embedding, is based on distributional hypothesis of Harris (1954): if contexts of two words are similar, the meanings of the two words are also similar.

In this thesis, we formalize unsupervised algorithms using our prior knowledge that is relevant with discourse constituents, discourse nuclearities, and discourse relations. More specifically, for each subtask, we first observe discourse phenomena carefully to find knowledge that is relevant with the target elements. Then, we use the prior knowledge in our unsupervised algorithm that induces the target element.

2.5 Related Work

In this section, we briefly review related works and emphasize the contributions of this thesis with respect to them. More methodologically related works will be introduced in the corresponding chapters: Chapter 3, Chapter 4, or Chapter 5.

As we have already described in Section 1.2 and Section 2.1, existing discourse parsers require hand-annotated discourse structures to tune the parameters of the learnable models (e.g., scoring functions). HILDA (Hernault et al., 2010b) and DPLP (Ji and Eisenstein, 2014), the two most popular off-the-shelf English discourse parsers, are trained on the RST-DT corpus (Carlson et al., 2001) and have been utilized in a variety of downstream applications (Hirao et al., 2013; Yoshida et al., 2014; Bhatia et al., 2015;

Ji and Smith, 2017). However, there is still a significant gap between the automatically parsed trees and the human-annotated trees. This large gap is due to the fact that these parsers rely on a small amount of annotated discourse structures, which are expensive, time-consuming to create, and sometimes highly ambiguous. In this thesis, we aim to tackle this problem and introduce unsupervised algorithms for discourse parsing. To the best of our knowledge, this is the first attempt to develop a fully unsupervised discourse parser.

Unsupervised parsing (or grammar induction) has been studied over the decades (Lari and Young, 1990; Clark, 2001; Klein, 2005; Smith, 2006; Naseem et al., 2010; Jin et al., 2018). However, existing studies on unsupervised parsing mainly focus on sentence structures, such as phrase structures (Lari and Young, 1990; Klein and Manning, 2002; Golland et al., 2012; Jin et al., 2018) or dependency structures (Klein and Manning, 2004; Berg-Kirkpatrick et al., 2010; Naseem et al., 2010; Jiang et al., 2016), though text-level structural regularities can also exist beyond the scope of a single sentence. In contrast, this thesis proposes unsupervised "discourse" parsing algorithms.

A number of unsupervised algorithms have been proposed in NLP. Generally, those algorithms are based on some linguistic insights and prior linguistic knowledge. Mikolov et al. (2013a,b) and Pennington et al. (2014) proposed unsupervised pre-training algorithms of word embeddings based on distributional hypothesis of Harris (1954). The distributional hypothesis is extended for sentence-level or document-level embeddings (Kiros et al., 2015; Le and Mikolov, 2014; Cer et al., 2018). Ling et al. (2015) and Nishida and Nakayama (2017) hypothesized that word order is crucially relavant with natural language grammars and proposed to learn syntactically plausible word embeddings by considering word positions. Klein and Manning (2004), Naseem et al. (2010), and Gimpel and Smith (2012) showed that linguistically motivated initialization techniques improve unsupervised syntactic parsing performance by 20 $\sim$ 40 points, compared to the random initialization. Unlike these studies, in this thesis, we explore prior knowledge that focuses especially on discourse structures.

Chapter 3

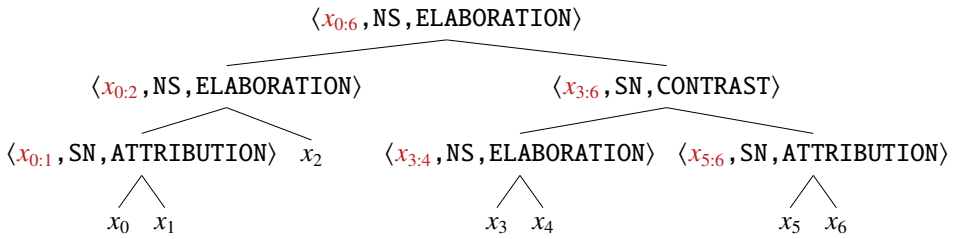
Unsupervised Learning for Discourse Constituency Parsing

3.1 Motivation

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

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

One possible solution to these problems is grammar induction (or unsupervised syntactic parsing) algorithms for discourse parsing. However, existing studies on unsupervised parsing mainly focus on sentence structures, such as phrase structures (Lari and Young, 1990; Klein and Manning, 2002;



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

Fig. 3.1 An example of RST-based discourse constituent structure we assume in this chapter. Leaf nodes x_i correspond to non-overlapping clause-level text segments, while internal nodes consists of three orthogonal elements: discourse constituents $x_{i:j}$, discourse nuclearities (e.g., `NS`), and discourse relations (e.g., `ELABORATION`).

Golland et al., 2012; Jin et al., 2018) or dependency structures (Klein and Manning, 2004; Berg-Kirkpatrick et al., 2010; Naseem et al., 2010; Jiang et al., 2016), though text-level structural regularities can also exist beyond the scope of a single sentence. For instance, in order to convey information to readers as intended, a writer should arrange utterances in a coherent order.

We tackle these problems by introducing ***unsupervised discourse parsing***, which induces discourse structures for given text without relying on human-annotated discourse structures. Based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) that is one of the most widely accepted theories of discourse structure, we assume that coherent text can be represented as tree structures, such as the one in Figure 3.1. The leaf nodes correspond to non-overlapping clause-level text spans called *elementary discourse units* (EDUs). Consecutive text spans are combined to each other recursively in a bottom-up manner to form larger text spans (represented by internal nodes) up to a global document span. These text spans are called *discourse constituents*. The internal nodes are labeled with both nuclearity statuses (e.g., *Nucleus-Satellite* or `NS`) and rhetorical relations (e.g., `ELABORATION`, `CONTRAST`) that hold between connected text spans.

In this chapter[1], we especially focus on unsupervised induction of an unlabeled discourse constituent structure (i.e., a set of unlabeled discourse

[1] This chapter is mainly based on Nishida and Nakayama (to appear).

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

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

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

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

On the RST Discourse Treebank (RST-DT) (Carlson et al., 2001), we compared our parse trees with manually-annotated ones. We observed that our method achieves a Micro F_1 score of 68.6% (84.6%) in the (corrected) RST-PARSEVAL (Marcu, 2000b; Morey et al., 2018), which is comparable

with or even superior to fully supervised parsers. We also investigated the discourse constituents that can or cannot be learned well by our method.

The rest of this chapter is organized as follows: Section 3.2 introduces the related work. Section 3.3 gives the details of our parsing model and training algorithm. Section 3.4 describes the experimental setting and Section 3.5 discusses the experimental results. We summarize this chapter in Section 3.6.

3.2 Related Work

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

In general, these methods employ the inside-outside (dynamic programming) re-estimation (Baker, 1979) in the E step. However, Spitkovsky et al. (2010) showed that Viterbi training (Brown et al., 1993), which uses only the best-scoring tree to count the grammatical patterns, is not only computationally more efficient but also empirically more accurate in longer sentences. These properties are, thus, suitable for "document-level" grammar induction, where the document length (i.e., the number of EDUs) tends to be long.[2] In addition, as explained later in Section 3.3, we incorporate Viterbi EM with a margin-based criterion (Stern et al., 2017; Gaddy et al., 2018), which allows us to avoid explicitly counting each possible discourse constituent pattern symbolically, which is generally too sparse and appears only once.

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

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

Our method also relies on recent work on RST parsing. In particular, one of the initialization methods in our EM training (in Subsection 3.3.3(i)) is inspired by the inter-sentential and multi-sentential approach used in RST parsing (Feng and Hirst, 2014; Joty et al., 2013, 2015). We also follow prior studies (Sagae, 2009; Ji and Eisenstein, 2014) and utilize syntactic information, i.e., dependency heads, which contributes to further performance gains in our method.

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

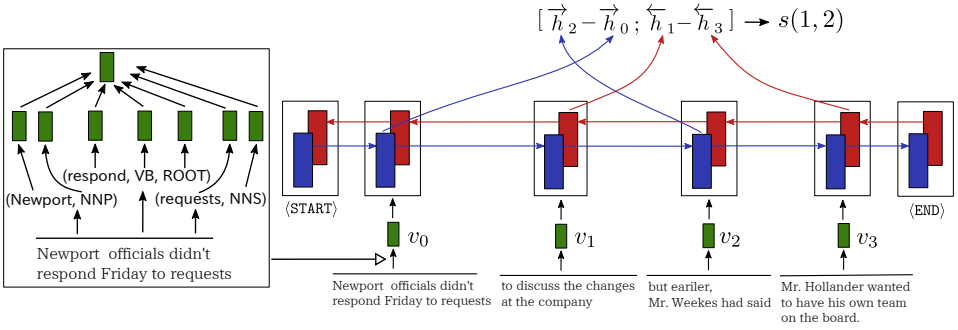


Fig. 3.2 Our span-based discourse parsing model. We first encode each EDU based on the beginning and ending words and Part-of-Speech tags using embeddings. We also embed head information of each EDU. We then run a bidirectional LSTM and concatenate the span differences. The resulting vector is used to predict the constituent score of the text span (i, j). This figure illustrates the process for the span $(1, 2)$.

3.3 Methodology

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

3.3.1 Parsing Model

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

$$\hat{T} = \underset{T \in \text{valid}(\boldsymbol{x})}{\text{argmax}} \; s(\boldsymbol{x}, T) \tag{3.1}$$

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

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

$$s(\boldsymbol{x}, T) = \sum_{(i,j) \in T} s(i, j). \tag{3.2}$$

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

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

Feature Extraction and Scoring

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

$$\boldsymbol{v}_i^{\text{bw}} = \text{Embed}_{\text{w}}(b_w), \tag{3.3}$$

$$\boldsymbol{v}_i^{\text{ew}} = \text{Embed}_{\text{w}}(e_w), \tag{3.4}$$

where b_w and e_w denote the beginning and end words of the i-th EDU, and Embed_{w} is a function that returns a parameterized embedding of the input word. We also encode the POS tags corresponding to b_w and e_w as follows:

$$\boldsymbol{v}_i^{\text{bp}} = \text{Embed}_{\text{p}}(b_p), \tag{3.5}$$

$$\boldsymbol{v}_i^{\text{ep}} = \text{Embed}_{\text{p}}(e_p), \tag{3.6}$$

where Embed_{p} is an embedding function for POS tags.

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

To sum up, we now have triplets of head information, $\{(h_w, h_p, h_r)_i\}_{i=0}^{n-1}$, each denoting the head word, the head POS, and the head relation of the i-th

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

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

EDU, respectively. We embed these symbols using look-up tables:

$$\boldsymbol{v}_i^{\mathrm{hw}} = \mathrm{Embed}_{\mathrm{w}}(h_w), \tag{3.7}$$

$$\boldsymbol{v}_i^{\mathrm{hp}} = \mathrm{Embed}_{\mathrm{p}}(h_p), \tag{3.8}$$

$$\boldsymbol{v}_i^{\mathrm{hr}} = \mathrm{Embed}_{\mathrm{r}}(h_r), \tag{3.9}$$

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

Finally, we concatenate these embeddings:

$$\boldsymbol{v}'_i = [\boldsymbol{v}_i^{\mathrm{bw}}; \boldsymbol{v}_i^{\mathrm{ew}}; \boldsymbol{v}_i^{\mathrm{bp}}; \boldsymbol{v}_i^{\mathrm{ep}}; \boldsymbol{v}_i^{\mathrm{hw}}; \boldsymbol{v}_i^{\mathrm{hp}}; \boldsymbol{v}_i^{\mathrm{hr}}], \tag{3.10}$$

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

$$\boldsymbol{v}_i = \mathrm{ReLU}(\boldsymbol{W}\boldsymbol{v}'_i + \boldsymbol{b}). \tag{3.11}$$

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

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

$$\overrightarrow{\boldsymbol{h}}_0, \ldots, \overrightarrow{\boldsymbol{h}}_{n-1} = \overrightarrow{\mathrm{LSTM}}(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1}), \tag{3.12}$$

$$\overleftarrow{\boldsymbol{h}}_0, \ldots, \overleftarrow{\boldsymbol{h}}_{n-1} = \overleftarrow{\mathrm{LSTM}}(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1}). \tag{3.13}$$

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

$$\boldsymbol{h}_{i,j} = [\overrightarrow{\boldsymbol{h}}_j - \overrightarrow{\boldsymbol{h}}_{i-1}; \overleftarrow{\boldsymbol{h}}_i - \overleftarrow{\boldsymbol{h}}_{j+1}]. \tag{3.14}$$

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

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

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

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

$$s(i, j) = \text{MLP}(\boldsymbol{h}_{i,j}), \tag{3.15}$$

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

Decoding

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

$$\text{C}[i, i] = 1. \tag{3.16}$$

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

$$\begin{aligned} \text{C}[i, j] = s(i, j) \\ + \max_{i \leq k < j} \text{C}[i, k] + \text{C}[k + 1, j], \end{aligned} \tag{3.17}$$

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

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

3.3.2 Unsupervised Learning Using Viterbi EM

In this work, we use Viterbi EM (Brown et al., 1993; Spitkovsky et al., 2010), a varient of the EM algorithm and *self-training* (McClosky et al., 2006a,b), to train the span-based discourse constituency parser (Subsection 3.3.1) in an unsupervised manner. Viterbi EM has suitable properties for discourse processing, as described later in this section.

Overall Procedure

The overall learning procedure is outlined in Algorithm 1. We first automatically sample initial trees based on our prior knowledge of document structures (described later in Subsection 3.3.3) and then perform the M step

Algorithm 1 Overall Learning Procedure

function TRAIN($\mathcal{X}, \mathcal{X}_{\text{dev}}, \mathcal{T}^*_{\text{dev}}, \theta_{\text{init}}$)
  $\mathcal{T} \leftarrow$ SAMPLETREES($\mathcal{X}$); ▷ Init.
  $\theta \leftarrow$ UPDATE($\mathcal{X}, \mathcal{T}, \theta_{\text{init}}$); ▷ M step
  $\delta \leftarrow \infty$, $\text{F}_1^{\text{old}} \leftarrow 0$;
  while $\delta > 0$ **do**
    $\mathcal{T} \leftarrow$ PARSE($\mathcal{X}, \theta$); ▷ E step
    $\theta \leftarrow$ UPDATE($\mathcal{X}, \mathcal{T}, \theta$); ▷ M step
    $\text{F}_1^{\text{new}} \leftarrow$ EVAL($\mathcal{X}_{\text{dev}}, \theta, \mathcal{T}^*_{\text{dev}}$); ▷ Val.
    $\delta \leftarrow \text{F}_1^{\text{new}} - \text{F}_1^{\text{old}}$, $\text{F}_1^{\text{old}} \leftarrow \text{F}_1^{\text{new}}$;
  end while
  return θ;
end function

on the sampled trees to initialize the model parameters. After the initialization step, we repeat the E step and the M step in turns. To perform early stopping, we use a held-out development set of 30 documents with annotated trees $\mathcal{T}^*_{\text{dev}}$, which are never used as the supervision to estimate the parsing model.

E Step

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

$$\mathcal{D} = \{(\boldsymbol{x}, \hat{T}) \mid \boldsymbol{x} \in \mathcal{X}, \hat{T} = \underset{T \in \text{valid}(\boldsymbol{x})}{\text{argmax}}\ s(\boldsymbol{x}, T)\} \tag{3.18}$$

where valid($\boldsymbol{x}$) denotes a set of all valid trees for $\boldsymbol{x}$, $s(\boldsymbol{x}, T)$ is defined in Equation (3.2), and $\hat{T}$ is the highest-scoring parse tree based on the current model.

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

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

M Step

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

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

$$s(\boldsymbol{x}, \hat{T}) \geq s(\boldsymbol{x}, T') + \Delta(\hat{T}, T'), \tag{3.19}$$

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

$$\Delta(\hat{T}, T') = |\hat{T}| - |\hat{T} \cap T'|, \tag{3.20}$$

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

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

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

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

The highest-scoring negative tree T' ($\neq \hat{T}$) can be efficiently found by modifying the dynamic programming algorithm in Equation (3.17). In particular, we replace $s(i, j)$ with $s(i, j) + 1[(i, j) \notin \hat{T}]$.

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

3.3.3 Initialization in EM

In general, the EM algorithm tends to get stuck in local optima of the objective function (Charniak, 1993). Therefore, proper initialization is vital

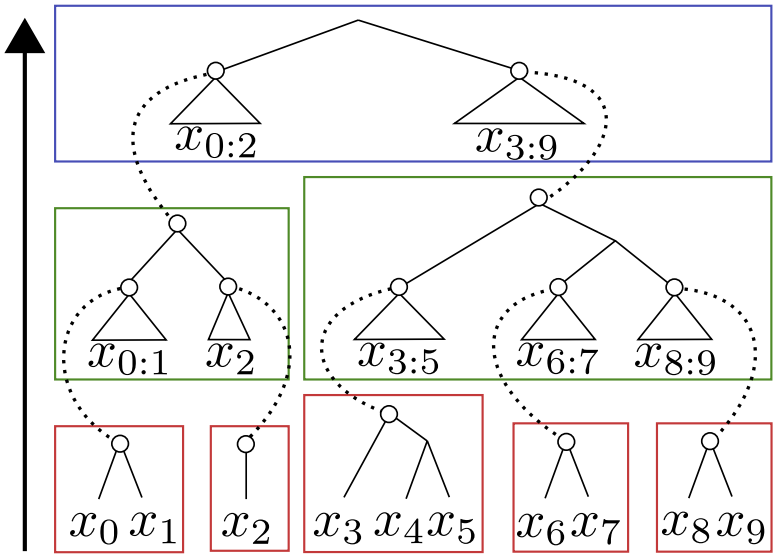


Fig. 3.3 We build a discourse constituent structure incrementally in a bottom-up manner. Sentence-level subtrees are shown in red rectangles, paragraph-level subtrees in green rectangles, and the document-level tree in a blue rectangle.

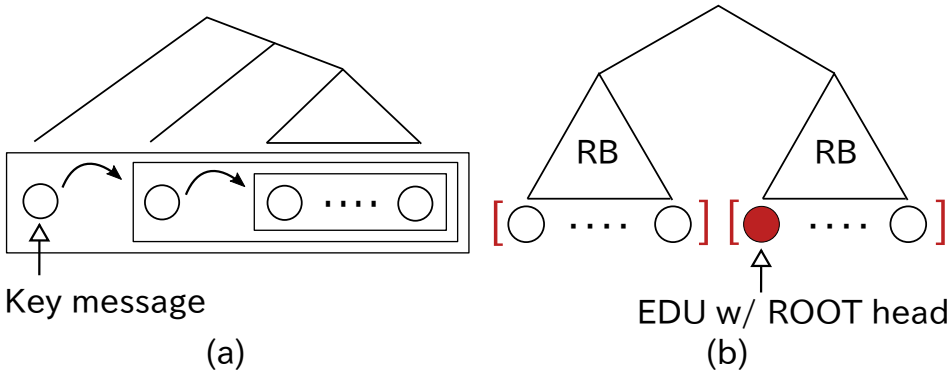
in order to avoid trivial solutions. This phenomenon has also been observed in EM-based grammar induction (Klein and Manning, 2004; Gimpel and Smith, 2012).

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

(i) Document Hierarchy

It is intuitively reasonable to consider that (elementary) discourse units belonging to the same textual chunk (e.g., sentence, paragraph) tend to form a subtree before crossing over the chunk boundaries. For example, we can assume that EDUs in the same sentence are preferentially connected with each other before getting combined with EDUs in other sentences. Actually, Joty et al. (2013, 2015) and Feng and Hirst (2014) observed that it is effective to incorporate inter-sentential and multi-sentential parsing to build a document-level tree.

First, we split an input document into sentence-level and paragraph-level segments by detecting sentence and paragraph boundaries, respectively. We



[[[The financial accounting standards board's coming
nsubj
rule on disclosure] [involving financial instruments]]
x_0 acl x_1
[[will be effective for financial statements with fiscal
ROOT
years] [ending after June 15, 1990.]]]
x_2 acl x_3 (wsj_0603)

(c)

Fig. 3.4 (a) We assume that an important text element tends to appear at earlier positions in the text, and the text following it complements the message, which leads to the right-heavy structure. (b)-(c) We split a intra-sentential EDU sequence into two subsequences based on the location of the EDU with the ROOT word. We build right-branching trees for each subsequence individually and finally bracket them. Head words are underlined.

obtain sentence segmentation by applying the Stanford CoreNLP (Manning et al., 2014) to the concatenation of EDUs. We also extract paragraph boundaries by detecting empty lines in the raw documents.[6] We then build a discourse constituent structure incrementally from sentence-level subtrees to paragraph-level subtrees and then to the document-level tree in a bottom-up manner. Figure 3.3 shows this process.

(ii) Discourse Branching Tendency

The second prior knowledge relates to information order in discourses and the branching tendencies of discourse trees. In general, an important text element tends to appear at earlier positions in the document, and then the

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

text following it complements the message, which is reflected in the Right Frontier Constraint (Polanyi, 1985) in Segmented Discourse Representation Theory (Asher and Lascarides, 2003). This tendency can be assumed to hold recursively. Therefore, it is reasonable to consider that discourse structures tend to form right-heavy trees, as shown in Figure 3.4(a). Based on this assumption, we build right-branching trees for sentence-level, paragraph-level, and document-level discourse structures in the initial-tree sampling.

(iii) Syntax-Aware Branching Tendency

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

To resolve this problem, we introduce another, more fine-grained, knowledge concept for sentence-level discourse structures. We expect that sentence-level trees are more strongly affected by syntactic cues (e.g., dependency graphs) than paragraph-level or document-level trees. More specifically, given an EDU sequence of one sentence, $x_i, \cdots, x_j$, we focus on a position of the EDU x_k with a head word that is in a ROOT relation with its parent in the dependency graph. We assume that the sub-sequence after the ROOT EDU, $x_{k:j}$, roughly corresponds to the predicate of the sentence, and the sub-sequence before the ROOT EDU, $x_{i:k-1}$, corresponds to the subject. We build right-branching trees for each sub-sequence individually and finally bracket them. We illustrate the procedure in Figure 3.4(b)-(c).

(iv) Locality Bias

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

Instead of introducing the locality bias into the initial-tree sampling, we encode it into the decoding algorithm in training and inference. More

precisely, we re-write the CKY recursion in Equation (3.17) as follows:

$$\mathrm{C}[i, j] = s(i, j) + \frac{\lambda}{|i - j + 1|} + \max_{i \leq k < j} \mathrm{C}[i, k] + \mathrm{C}[k + 1, j], \tag{3.21}$$

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

3.4 Experiment Setup

3.4.1 Data

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

We tokenized the documents using Stanford CoreNLP tokenizer and converted them to lower cases. We also replaced digits with "7" (e.g., "12.34" → "77.77") to reduce data sparsity. We also replaced out-of-vocabulary tokens with special symbols "⟨ UNK ⟩."

3.4.2 Metrics

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

The traditional evaluation procedure for RST parsing is RST-PARSEVAL (Marcu, 2000b), which adapts the PARSEVAL for the RST representation shown in Figure 3.5(a)-(b). However, Morey et al. (2018) showed that, as shown in Figure 3.5(c), traditional RST-PARSEVAL gives a higher-than-expected score because it considers pre-terminals (i.e., spans of length 1), which can-

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

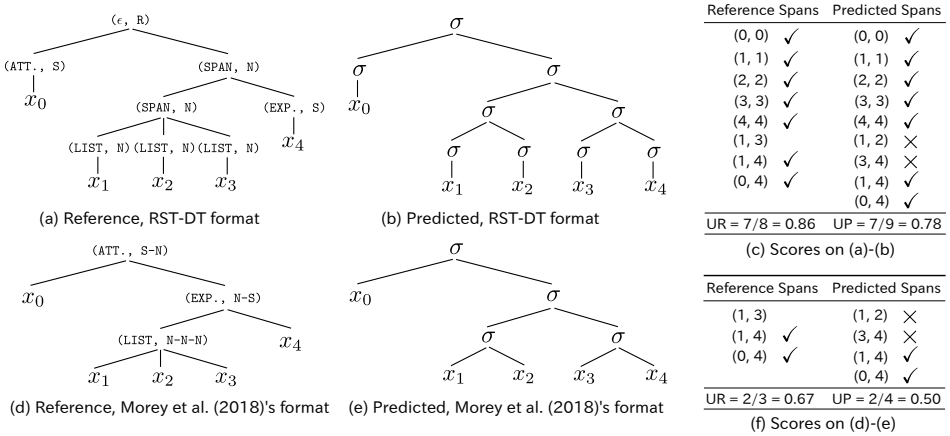


Fig. 3.5 Variants of RST encodings and the corresponding unlabeled constituency scores: Unlabeled Recall (UR) and Unlabeled Precision (UP).

not be incorrect in the unlabeled constituency metrics. We therefore follow Morey et al. (2018) and perform the encoding of RST trees as shown in Figure 3.5(d)-(f). That is, we exclude spans of length 1 and include the root node. We also do not binarize the gold-standard trees.

3.4.3 Baselines

To quantitatively evaluate our unsupervised discourse constituency parser, it is necessary to develop strong baseline parsers. We thus propose *Combinational Incremental Parsers* (CIPs), which automatically and incrementally build a discourse (unlabeled) constituent structure from an EDU sequence based on the prior knowledge introduced in Subsection 3.3.3. That is, CIPs first build sentence-level discourse trees based on sentence segmentation using an *elementary parser* f_s. They then build paragraph-level trees using another elementary parser f_p, and finally output the document-level tree using f_d. An elementary parser is a function that returns a single tree given a sequence of EDUs or subtrees. CIPs can be represented as a triplet of elementary parsers, i.e.,

$$\langle f_s, f_p, f_d \rangle. \tag{3.22}$$

Inspired by earlier studies in unsupervised syntactic constituency parsing (Klein and Manning, 2001a,b; Klein, 2005; Seginer, 2007), we prepare the following four candidates for the elementary parsers:

Right Branching (RB) Given a sequence of elements (i.e., EDUs or subtrees), RB always chooses the left-most element as a left terminal node and then treats the remaining elements as a right nonterminal (or terminal). This procedure is recursively applied to the remaining elements on the right, resulting in $(x_0\ (x_1\ (x_2\ \ldots)))$. As described in Subsection 3.3.3, we predict that RB somewhat captures the branching tendency of discourse informational structures. RB was also used as a strong baseline for unsupervised syntactic constituency parsing in Klein and Manning (2001b).

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

Adaptive Right Branching (RB*) We augment RB by considering the syntax-aware branching tendency, described in Subsection 3.3.3(iii). That is, based on the position of the head EDU (with the ROOT relation), we split the sentence into two parts and then perform RB for each sub-sequence.

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

3.4.4 Hyperparameters

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

3.5 Results and Discussion

In this section we report the results of the experiments and discuss them. We first discuss the comparison results of our method with baselines and the fully supervised RST parsers, including the results published in literature (Subsection 3.5.1). We then investigate the impact of initialization methods

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

Table 3.1 Unlabeled constituency scores in the corrected RST-PARSEVAL (Morey et al., 2018) against non-binarized trees. UP and UR represent Unlabeled Precision and Unlabeled Recall, respectively. For reference, we also show the traditional RST-PARSEVAL Micro F_1 scores in parentheses. Asterisk indicates that we have borrowed the score from Morey et al. (2018).

(Subsection 3.5.2). Finally, we provide our analysis on discourse constituents induced by our method (Subsection 3.5.3).

3.5.1 Performance Comparison

We compared our method with the baselines described in Subsection 3.4.3. We also included the previous work (Kobayashi et al., 2019) on unsupervised RST parsing as our baseline, though it is not a fair comparison because they use binarized golden trees for evaluation.[8] For reference, we also compared our method with fully supervised parsers: the supervised version of our

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

model[9] and recent supervised parsers (Feng and Hirst, 2014; Joty et al., 2015) that incorporate intra-sentential and multi-sentential parsing as in our parser.

Table 3.1 shows the unlabeled constituency scores in the corrected RST-PARSEVAL (Morey et al., 2018) against non-binarized trees. We also show the traditional RST-PARSEVAL Micro F_1 scores in parentheses. $\langle f_s, f_d \rangle$ indicates that we used only sentence boundaries and discarded paragraph boundaries. The scores of external supervised parsers (Feng and Hirst, 2014; Joty et al., 2015) are borrowed from Morey et al. (2018).

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

Table 3.1 also demonstrates that our method outperforms all the baselines and achieves an F_1 score of 67.5% (84.0%). If we use the best baseline for initial-tree sampling in Viterbi EM, the performance further improves to 68.0% (84.3%).

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

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

Our method also achieves comparable or superior results to supervised models. We suspect that the reason why the supervised version of our model outperforms the external supervised parsers (Feng and Hirst, 2014; Joty et al.,

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

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

Knowledge	Initial Trees	Micro F_1
No (Uniform)	BU	58.9
(i)	$\langle BU_s, BU_p, BU_d \rangle$	59.1
(i)+(ii)	$\langle RB_s, RB_p, RB_d \rangle$	59.7
(i)+(ii)+(iii)	$\langle RB^*_s, RB_p, RB_d \rangle$	66.3
(i)+(ii)+(iii)+(iv)	$\langle RB^*_s, RB_p, RB_d \rangle$	67.5
Best baseline	$\langle RB^*_s, RB_p, LB_d \rangle$	68.0

Table 3.2 Comparison of initialization methods in our Viterbi training.

2015) is mostly dependent on feature extraction and the use of paragraph boundaries.

3.5.2 Impact of Initialization Methods

Here, we evaluate the importance of initialization in Viterbi EM. Beginning with uniform initialization, we incrementally applied the initialization techniques introduced in Subsection 3.3.3 and investigated their impact on the results.

Table 3.2 shows the results. We observed that our model yields the lowest score of 58.9% with uniform initialization (no prior knowledge). By introducing Document Hierarchy in Subsection 3.3.3(i), parsing performance improves slightly to 59.1%. This result is interesting because the unlabeled constituency scores of BU and $\langle BU_s, BU_p, BU_d \rangle$ are quite different (19.5 vs. 55.5; see Table 3.1). We then introduced Discourse Branching Tendency in Subsection 3.3.3(ii) by replacing BU with RB in the CIP, which also improved the performance, slightly, to 59.7%. We then introduced Syntax-Aware Branching Tendency in Subsection 3.3.3(iii) by replacing RB with RB^* only for the sentence level, which brought a considerable performance gain of 6.6 points (66.3%). Finally, we introduced Locality Bias in Subsection 3.3.3(iv) and achieved 67.5%. We also found that our model can be improved further to 68.0% if we use the best baseline for initialization.

In total, these initialization techniques made a difference of 9.1 points compared to uniform initialization, i.e., 58.9 $\rightarrow$ 68.0, which implies that initialization should be carefully considered in unsupervised discourse (constituency) parsing using EM and that the prior knowledge we proposed in Subsection 3.3.3(i)-(iv) can capture some of the tendencies of document structures. We also found that Syntax-Aware Branching Tendency is most

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

Table 3.3 The best four and worst four rhetorical relations with their corresponding Unlabeled Recall scores. The relations are ordered according to scores of the unsupervised parser.

effective among the techniques, which suggests that more detailed knowledge can yield further improvements.

3.5.3 Learned Discourse Constituentness

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

First, we calculated Unlabeled Recall (UR) scores for each relation class in RST-DT. We used 18 coarse-grained classes. Please note that we only focus on constituent spans $\{(i, j)\}$ because our method does not predict relation labels. Table 3.3 shows the results of the best four and the worst four relation classes of our method. We compare the results with the supervised version.

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

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

As shown in the upper part of Table 3.4, we can observe that our method learns some aspects of discourse constituentness that seems linguistically reasonable. In particular, we found that our method has a potential to predict brackets for (1) clauses with connectives qualifying other clauses from right to left (e.g., "X [because B.]") and (2) attribution structures (e.g., "say that [B]"). These results indicate that our method is good at identifying discourse constituents near the end of sentences (or paragraphs), which is natural because RB is mainly used for generating initial trees in EM training. The bottom part of Table 3.4 demonstrates that the beginning position of the text span is also important to estimate constituenthood, along with the ending position.

3.6 Summary

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

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

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

Table 3.4 Discourse constituents and their predicted scores (in parentheses). We show the discourse constituents (in bold) in the RST-DT test set, which have relatively high span scores. We did NOT use any sentence/paragraph boundaries for scoring.

Chapter 4

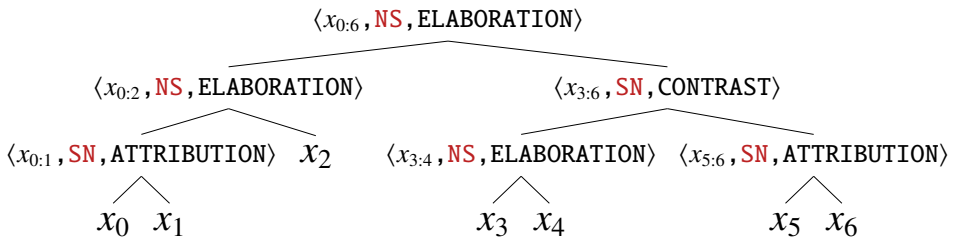
Unsupervised Induction for Discourse Nuclearity Classification

4.1 Motivation

As discussed in Section 2.1, discourse parsing is still a challenge bacause of the high cost and low reliability of hand-annotated discourse structures. The goal of this thesis is to alleviate these problems by unsupervised discourse parsing algorithms, which induces discourse structures for given text without relying on manually annotated structures.

Here, in Figure 4.1, we again show a RST-based discourse tree structure that we assume throughout this thesis. In the previous chapter (Nishida and Nakayama, to appear), we have already proposed an unsupervised algorithm that builds an "unlabeled" (or naked) constituent structures from a sequence of EDUs, which corresponds to the first subtask in our roadmap described in Section 2.3.

In this chapter, we tackle the second subtask in our roadmap. That is, given an unlabeled discourse tree structure, we aim to assign ***nuclearities*** to each internal node in an unsupervised manner. Nuclearity is a type of directional information that holds between two connected text spans (Mann and Thompson, 1988). In particular, for simplicity, we assume the following two nuclearity classes: *Nucleus-Satellite* (`NS`), *Satellite-Nucleus* (`SN`). A *nucleus* represents the salient text span, while a *satellite* indicates supporting or background span to the nucleus. This assumption always holds if we (1) apply a right-heavy binarization procedure to n-ary trees following Soricut



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

Fig. 4.1 In this chapter, we assume that coherent text can be represented as a RST-like tree structure. Leaf nodes x_i correspond to clause-level segments, while internal nodes consists of three orthogonal elements: discourse constituents $x_{i:j}$, discourse nuclearities (e.g., NS), and discourse relations (e.g., ELABORATION).

and Marcu (2003) and (2) treat a preceding (left) span as the main nucleus if two connected spans are equally important following Li et al. (2014b).

Carlson and Marcu (2001) propose the following *deletion test* to determine nuclearities:

- *Deletion test*: If a nucleus is deleted, the text span that is left is much less coherent. If a satellite is deleted, the text segment that is left is still coherent, although it may be somewhat weaker.

We aim to perform the deletion test automatically to identify nuclearities in an unsupervised manner. To computer discourse irreducibility, we incorporate sentence-importance measures in extractive summarization (Nenkova and McKeown, 2012; Allahyari et al., 2017) and a simple recursive algorithm. Nuclearities are determined based on the comparison of discourse irreducibility scores of text spans. It is worth noting that our proposal in this chapter is not for training some models but for directly inducing nuclearities from discourse irreducibility scores. The discourse irreducibility scores are computed by off-the-shelf sentence-scoring techniques in (unsupervised) extractive summarization.

We compare five irreducibility measures and report nuclearity-classification accuracies on the RST-DT corpus (Carlson et al., 2001). The experimental

results demonstrate that our unsupervised approach outperforms clustering-based classifiers using K-Means. We also found that multiple measures can be combined complementarily for further improvements and achieve 58.52% balanced accuracy.

4.2 Why Not Clustering?

Clustering may be the simplest approach to unsupervised classification. Actually, clustering is the first approach we adopted in our preliminary experiments. For example, after performing K-Means with $K = 2$ (each corresponding to NS and SN) on discourse-constituent features, we can develop a classifier that assigns a nuclearity status to an unseen discourse constituent according to the distance from the centroid vectors in the feature space:

$$c^* = \underset{c \in \{\texttt{NS},\texttt{SN}\}}{\operatorname{argmin}} \|f(x) - \mathbf{v}_c\|, \tag{4.1}$$

where $f(x)$ represents the feature vector of a discourse constituent x and $\mathbf{v}_c$ denotes the centroid vector of the cluster $c \in \{\texttt{NS}, \texttt{SN}\}$.[1]

However, can nuclearity classes be really found by a clustering algorithm? Can they be separately distributed in a feature space? Actually, both syntactic structures and meanings of discourse constituents do not always match their nuclearity statuses:

(20) [The explosions began]$_{\pi_1}$ [*when a seal blow out.*]$_{\pi_2}$

(21) [*When a seal blow out,*]$_{\pi_1}$ [the explosions began.]$_{\pi_2}$

(22) [Never mind.]$_{\pi_1}$ [*You already know the answer.*]$_{\pi_2}$

We show the nuclei in normal font and the satellites in italics. The examples in (20) and (21) represent the same meaning as a whole. However, their nuclearity statuses are different (i.e., NS vs. SN). Moreover, (22) belongs to the same nuclearity status with (20), though (22) has a totally different meaning and a syntactic structure from (20). These phenomena make it hard to extract nuclearity-oriented features.

To empirically examine this argument, consider Figure 4.2. In Figure 4.2, each point is a discourse constituent stored in RST-DT, colored by their corresponding nuclearity statuses. We first embed two connected text spans as a

[1] To use this clustering-based classifier, we have to align each cluster ID $\in \{1, 2\}$ with each nuclearity status, which is also a difficult task without annotated data.

Fig. 4.2 Discourse constituents of two nuclearity classes, projected onto the two-dimensional space using t-SNE (van der Maaten and Hinton, 2008). The distinction among the two classes seems less discernible.

simple sum of pre-trained word embeddings[2] (Hermann and Blunsom, 2014) and concatenate the two vectors to represent each discourse constituent. The discourse-constituent vectors have been projected onto the two-dimensional space using t-SNE (van der Maaten and Hinton, 2008) (perplexity = 10) for visualization.

Figure 4.2 demonstrates that nuclearity classes seem to have less discriminated distributions, and it is hard to believe that they could be separated by a clustering algorithm even in the full space.

4.3 Computing Discourse Irreducibility

In this section, we describe our proposed method for computing discourse irreducibility scores. We call a text span *irreducible*, if the span can not be removed without damaging the text coherence. Our hypothesis is that irreducible text spans are likely to be nuclei more than reducible ones.

[2]We used the 300-dimensional GloVe word embeddings that were trained on the 840 billion tokens (Pennington et al., 2014).

As a similar idea with the deletion test of Carlson and Marcu (2001), Mareček and Žabokrtský (2012) focus on reducibility of phrases for unsupervised dependency parsing. They removed word n-grams from a sentence and checked whether the rest of the sentence appears at least once in a corpus. The number of such reducible occurences is used to determine the reducibility of the n-gram.

However, it is difficult to perform such deletion test at a discourse level. Even after a text span of high reducibility is deleted, the text segment that is left rarely appears anywhere in the corpus.

In this work, we propose to incorporate sentence-importance measures in extractive summarization and a simple recursive algorithm to compute discourse irreducibility. We first compute irreducibility scores of the sentences in a text by leveraging extractive summarization techniques. Then, we compute the irreducibility scores of larger discourse constituents recursively in a bottom-up manner from the sentence irreducibility. The inner-sentence nuclearity statuses are automatically assigned based on our prior knowledge. In this work, we choose sentences as the beginning point of the recursive algorithm. This allows us to utilize publicly available pre-trained sentence encoders, such as Cer et al. (2018), that have been trained usually on large unlabeled corpora.

4.3.1 Sentence Irreducibility

Extractive summarization (Nenkova and McKeown, 2012; Allahyari et al., 2017) is one of the major approaches to text summarization. Extractive summarization produces the summary of an input text by extracting a subset of important sentences in the original text. Conventionally, (1) the summarization systems first extract feature vectors from each sentence in the text, then (2) they assign an importance score to each sentence, and finally (3) the top k most important sentences are selected to generate a summary by using a greedy algorithm or solving an optimization problem.

We use sentence-importance measures in (1)+(2) in extractive summarization for computing sentence irreducibility. In particular, we examine the following five sentence-importance measures: (1) SumBasic (Vanderwende et al., 2007), (2) Latent Semantic Analysis (LSA) (Gong and Liu, 2001; Steinberger et al., 2007), (3) a centroid-based measure (Radev et al., 2004; Rossiello et al., 2017), (4) LexRank (Erkan and Radev, 2004; Mihalcea and Tarau, 2004), and (5) heuristic measures.

SumBasic

SumBasic focuses on frequency of words as the indicators of the sentence importance (Vanderwende et al., 2007). Formally, it uses word probability as the indicators.

First, we compute unigram probabilities of words from an input document d:

$$P(w) = \frac{\text{count}_d(w)}{\sum_{w' \in d} \text{count}_d(w')}, \tag{4.2}$$

where $\text{count}_d(w)$ is the number of occurences of a word w in d. Then, we assign importance scores to each sentence based on the average probability of the words in the sentence:

$$g(s_i) = \frac{1}{|s_i|} \sum_{w \in s_i} P(w), \tag{4.3}$$

where $|i|$ denotes the number of words in the sentence s_i. Then, the highest-scoring sentence is selected. To avoid selecting similar sentences redundantly, SumBasic can discount the probabilities of the words that are already contained in the current summary:

$$P_{\text{new}}(w) = P_{\text{old}}(w) \times P_{\text{old}}(w). \tag{4.4}$$

This selection steps repeat until the summary of desired length is produced.

We use the selected order of each sentence for $g(s_i)$ instead of Eq. (4.3), because we found it yields better performance than the average word probability:

$$g(s_i) = \frac{1}{y_i}, \tag{4.5}$$

where $y_i \in [1, n]$ denotes the selected order of i.

Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is an unsupervised algorithm that extracts hidden representations of words, sentences, or documents via a matrix factorization (Deerwester et al., 1990). (Gong and Liu, 2001; Steinberger et al., 2007) used LSA to detect latent topics in an input document, based on which

importance scores of sentences are calculated. LSA considers that sentences discussing multiple important topics tend to be more important.

More formally, we first build a term-sentence matrix ($m \times n$ matrix) A from the input document, the cell A_{ij} of which represents a TFIDF weight of the term i in the j-th sentence. Then, we run Singular Value Decomposition (SVD) on A to approximate it as follows

$$A \approx U\Sigma V^{\top}. \tag{4.6}$$

The matrix Σ is a diagonal matrix ($k \times k$) where Σ_{kk} represents the weight of the latent topic k. We then build a topic-sentence matrix ($k \times n$ matrix) D:

$$D = \Sigma V^{\top}, \tag{4.7}$$

the cell D_{ij} of which represents the weight of the latent topic i in the j-th sentence. We set k to $n/5$. We follow (Steinberger et al., 2007) and compute a sentence score as follows:

$$g(j) = \sqrt{\sum_{i=1}^{k} D_{ij}^2}. \tag{4.8}$$

Centroid-based measure

Centroid-based methods in extractive summarization first detect topics in the input document and then computes centroids (or *pseudo-documents*) of each topic. Sentences are clustered and then assigned with importance scores according to the distance from the centroids of their corresponding clusters. It then chooses sentences from each topic by a selection algorithm.

To achieve this goal, it requires feature vectors of the sentences in the input. In this study, we use recent sentence-level embedding techniques proposed in (Rossiello et al., 2017) and (Cer et al., 2018). In (Rossiello et al., 2017), a sentence vector can be computed by using a sum of pre-trained word embeddings[3] in the sentence as follows:

$$f(s_i) = \sum_{w \in s_i} E[w], \tag{4.9}$$

[3]We compare 100/300-dim GloVe embeddings (Pennington et al., 2014) and the 300-dim word2vec embedding (Mikolov et al., 2013a) trained on GoogleNews.

where $f(s_i)$ denotes the vector of the sentence s_i and $E[w]$ returns the embedding of the word w. We also use the *Universal Sentence Encoder* (USE) (Cer et al., 2018), which is trained on multiple tasks and produces an embedding for each sentence. To detect topics in the document, we can use a clustering algorithm such as K-Means on the sentence embeddings, resulting in $c_1, \ldots, c_n$, where $c_i \in [1, K]$ represents the cluster ID of the sentence i. We then compute the centroids of each topic as the sum of sentence vectors that belong to the same topic:

$$\tilde{f}(d_k) = \sum_{s \in d_k} f(s), \tag{4.10}$$

where d_k denotes a set of sentences that belong to the topic k, i.e., $d_k = \{i \mid s_i \in d, c_i = k\}$. We then assign importance scores to each sentence using the cosine similarity to the corresponding centroids:

$$g(s_i) = \frac{f(s_i)^\top \tilde{f}(d_{c_i})}{\|f(s_i)\| \cdot \|\tilde{f}(d_{c_i})\|}. \tag{4.11}$$

In this work, following (Rossiello et al., 2017), we set $K = 1$ and compute a single centroid vector for the whole document.

It is worth noting that this centroid-based measure is different from the clustering-based classifier described in Section 4.2, though at first glance they seem to be similar. The centroid-based measure aims to estimate sentence-level informativeness scores, which is followed by the nuclearity-labeling algorithm introduced in Section 4.4. In contrast, the clustering-based method aims to solve nuclearity labeling as a classification task on discourse constituents, which has a strong assumption that discourse constituents of different nuclearity types are distributed separately in a feature space.

LexRank

LexRank utilizes PageRank algorithm (Mihalcea and Tarau, 2004) on a connected graph where vertices correspond to the sentences and the edges represent the similarity between the two sentences (Erkan and Radev, 2004). LexRank considers that sentences that have many connections with other sentences possibly discuss the center topic in the document and are more likely to be in the summary.

We first embed each sentence in the same way as in the centroid-based method. Then, we calculate the cosine similarity of two sentences using the

embeddings. We connect two sentences if the similarity is greater than a threshold. We set the threshold to 0.1. We then build a transition-probability matrix on the graph by normalizing the weights on the edges. We finally run the PageRank algorithm using the transition matrix and obtain the sentence scores, $\{g(s_i)\}_{i=1}^{n}$.

Heuristic measures

Some heuristic indicators have been used as feature vectors in learning-based models for extractive summarization. In this study, we adopt the following two heuristic measures.

- **Heuristic-Location** assumes that sentences appearing at earlier positions are more essential, i.e.,

$$g(i) = i/|d|, \tag{4.12}$$

  where $|d|$ denotes the number of all sentences in the document d.

- **Heuristic-Length** considers that sentences with more words have higher importances, i.e.,

$$g(i) = |s_i|. \tag{4.13}$$

4.3.2 Discourse Irreducibility

We compute discourse irreducibility recursively in a bottom-up manner from sentence irreducibility scores. A subtree of a discourse constituent, or a subsequence of sentences, $s_{i:j}$, can be denoted as follows:

$$T(s_{i:j}) = \Big(T(s_{i:k}), T(s_{k+1:j})\Big). \tag{4.14}$$

We define irreducibility scores of subtrees by using a recursive function G as follows:

$$G(T(s_{i:j})) = \text{Pool}(G(T(s_{i:k})), G(T(s_{k+1:j}))), \tag{4.15}$$

where Pool denotes the pooling function. In this work, we examined max-pooing and average-pooling functions, i.e.,

$$G(T(s_{i:j})) = \max(G(T(s_{i:k})), G(T(s_{k+1:j}))), \tag{4.16}$$

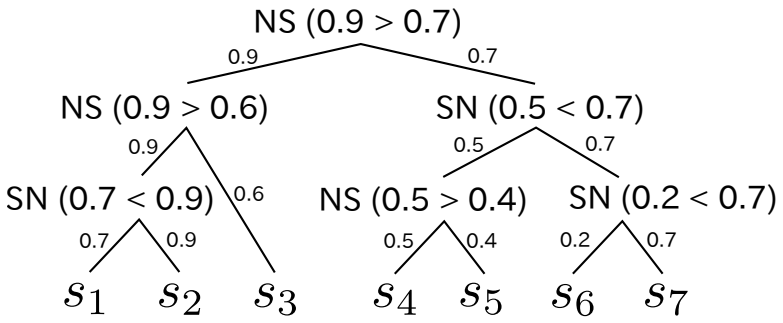


Fig. 4.3 The proposed algorithm for unsupervised nuclearity classification. The leaf nodes, $s_1, \ldots, s_n$, denote sentences in an input text. We first compute irreducibility scores of each sentence. Then, we compute the irreducibility scores of larger discourse constituents recursively in a bottom-up manner. Finally, we assign the nuclearity statuses by comparing the irreducibility scores of connected discourse constituents. Discourse constituents with higher irreducibility are treated as nuclei.

or

$$G(T(s_{i:j})) = \frac{G(T(s_{i:k})) + G(T(s_{k+1:j}))}{2}. \tag{4.17}$$

However, in our preliminary experiments, we found that the max-pooling yields higher accuracy than the average-pooling. Therefore, we use the max-pooling for G throughout our experiments.

For spans of length one (i.e., $i = j$), G is defined as follows:

$$G(T(s_{i:j})) = g(s_i), \tag{4.18}$$

where $g(s_i)$ denotes the importance (i.e., irreducibility) of sentence s_i that can be computed using the sentence-importance measures in Section 4.3.1.

4.4 Unsupervised Nuclearity Classification

In this section, we describe how to decide nuclearity statuses of each internal node after computing irreducibility scores of discourse constituents (including sentences). Figure 4.3 illustrates the process.

Given irreducibility scores of discourse constituents, at each internal node $T(x_{i:j})$, we compare the subtree scores and decide the nuclearity status

Algorithm 2 Nuclearity Classification

```
1:  function F(node, x, B, g)
2:      if node.is_internal() then
3:          node.left_child = F(node.left_child, x, B, g);
4:          node.right_child = F(node.right_child, x, B, g);
5:      end if
6:      i, j = node.span;
7:      if B[i, j] < 0 then
8:          node.nuclearity = NS;
9:      else if B[i, j] = 0 then
10:         node.nuclearity = NS;
11:         node.score = g(x_{i:j});
12:     else
13:         g_l = node.left_child.score;
14:         g_r = node.right_child.score;
15:         if g_l − g_r ≥ 0 then
16:             node.nuclearity = NS;
17:         else
18:             node.nuclearity = SN;
19:         end if
20:         node.score = max(g_l, g_r);
21:     end if
22:     return node;
23: end function
```

$l_{i,j}$:

$$l_{i,j} = \begin{cases} \texttt{NS} & (g_l - g_r \geq 0) \\ \texttt{SN} & (g_l - g_r < 0) \end{cases}, \tag{4.19}$$

where g_l and g_r denote the irreducibility scores of the left and right constituents, respectively. For internal nodes in sentence-level RST trees, we automatically assign the majority label (i.e., NS).

We show a pseudo code for this procedure in Algorithm 2. x_i corresponds to the i-th EDU (not sentence) in the text. B denotes a two-dimensional matrix containing sentence-boundary information: $\mathrm{B}[i, j] < 0$ indicates that the span (i, j) is within a sentence boundary, while $\mathrm{B}[i, j] = 0$ indicates that the span matches the sentence boundary. Otherwise, the span covers discourse constituents beyond a single sentence.

4.5 Experiment Setup

4.5.1 Data

Our approach does not require training. The only exceptions are the centroid-based measure and LexRank, which utilize publicly available word/sentence embeddings pre-trained on very large corpora. For evaluation, we used the RST Discourse Treebank (Carlson et al., 2001), which contains hand-annotated discourse structures for 385 English articles from the Wall Street Journal. RST-DT provides a pre-defined split for training and test sets. We used the training set for building our baseline classifiers. We used the test set for evaluation.

4.5.2 Metrics

Following Marcu (2000a) and Morey et al. (2018), we use the corrected RST-PARSEVAL metrics to evaluate parse trees against gold trees, which is analogous to PARSEVAL (Black et al., 1991), the standard evaluation methodology in syntactic constituency parsing. To avoid unexpected error propagations and focus only on the algorithms for unsupervised nuclearity identification, we use the gold-standard EDU segmentations and the unlabeled constituent structures in RST-DT. Standard Accuracy (SA) is equivalent to the conventional Micro F1 score on nuclearity-labeled spans and defined as the ratio of the number of correctly labeled spans to the total number of spans in the test set. To precisely analyze the performances on each nuclearity class, we show the class-wise recalls (i.e., `NS`-R and `SN`-R). Balanced Accuracy (BA) aims to deal with the class imbalance problem in the test set and is defined as the average of the class-wise recalls (Brodersen et al., 2010).

4.5.3 Baseline

We built clustering-based classifiers as the baselines in this study. Specifically, we developed K-Means ($K = 2$) classifiers using discourse constituents in the training set of the RST-DT corpus. The constituent-level vectors were computed in the same way as in Section 4.2. We used the 300-dim word2vec word embeddings (Mikolov et al., 2013a) trained on GoogleNews and the 100/300-dim GloVe vectors (Pennington et al., 2014). A nuclearity status is assigned to an unseen discourse constituent based on the distance from the

clusters' centroids. We show two results for each K-Means classifier, since the cluster IDs (i.e., {0, 1}) can be mapped either {0: NS, 1: SN} or {0: SN, 1: NS}.

4.5.4 UpperBound

We also examined the upper bound of our approach, since it may be unclear whether it is possible to correctly induce nuclearity statuses from sentence-level scalar values by comparing them in a bottom-up manner. We computed the gold-standard sentence scores using the fully-annotated RST trees. We first mapped a constituent tree to the corresponding dependency tree, following (Ji and Eisenstein, 2014; Li et al., 2014b; Braud et al., 2017). Given the dependency graph, we assign a gold irreducibility score to each EDU x_i according to the distance from the ROOT:

$$g(x_i) = 1/\text{distance}(x_i, \text{ROOT}), \tag{4.20}$$

where the distance is simply defined as the number of hops between x_i and the special-ROOT head. A sentence score is defined as the score of the head EDU of the sentence.

4.6 Results and Discussion

Table 4.1 summarizes the nuclearity-classification performances of our approach and the K-Means classifiers on the test data.

4.6.1 Evaluation of the K-Means Classifiers

The upper block in Table 4.1 shows the scores of the K-Means classifiers using different pre-trained word embeddings for feature extraction. The below block shows performances for our approach with different irreducibility measures. "USE" indicates that we used the publicly available Universal Sentence Encoder (Cer et al., 2018) for sentence embeddings, while "word2vec-k" and "GloVe-k" represent that we used the word2vec/GloVe word embeddings of k dimensions for sentence embeddings.

From the upper block, we can observe that the K-Means classifiers induce highly unbalanced cluster sizes and tend to predict the same label with the majority cluster, which is obvious from the difference between the class-wise recalls (NS-Recall vs. SN-Recall). The recall on the minority cluster

Method	NS-R	SN-R	SA (%)	BA (%)
K-Means (word2vec-300)	91.45	2.35	78.82	46.90
	8.55	97.65	21.18	53.10
K-Means (GloVe-100)	91.25	2.35	78.65	46.80
	8.75	97.65	21.35	53.20
K-Means (GloVe-300)	91.16	2.35	78.57	46.75
	8.84	97.65	21.43	**53.25**
SumBasic	42.95	52.35	44.29	47.65
SumBasic + Update	42.37	50.00	43.45	46.19
LSA	38.68	66.47	42.62	52.57
Centroid-based (word2vec-300)	50.05	54.71	50.71	52.38
Centroid-based (GloVe-100) $\cdots$ (a)	49.08	61.76	51.88	**56.00**
Centroid-based (GloVe-300)	50.83	60.00	52.13	55.41
Centroid-based (USE)	48.10	58.82	49.62	53.46
LexRank (word2vec-300)	49.95	54.12	50.54	52.03
LexRank (GloVe-100)	49.08	61.76	50.88	55.42
LexRank (GloVe-300)	51.02	58.82	52.13	54.92
LexRank (USE)	48.10	60.00	49.79	54.05
Heuristics-Location $\cdots$ (b)	100.00	0.00	85.82	50.00
Heuristics-Length	49.56	64.71	51.71	**57.13**
0.8$\times$ (a) + 0.2$\times$ (b)	57.63	59.42	57.88	**58.52**
UpperBound	100.00	100.00	100.00	100.00
Human	-	-	77.72	-

Table 4.1 Performances on unsupervised nuclearity identification on the RST-DT test set. The evaluation metrics are class-wise recalls (NS-Recall, SN-Recall), Standard Accuracy, and Balanced Accuracy. The upper block shows scores of the K-Means classifiers using different pre-trained word embeddings. The below block shows scores of our approach using different irreducibility measures.

(e.g., 2.35%, 8.55%) is much lower than that on the majority (e.g., 91.45%, 97.65%) with this method, which is not desirable for situations where both nuclearity types are equally important.

4.6.2 Evaluation of the Proposed Method

From the below block in Table 4.1, we can observe that the centroid-based measure and LexRank yield higher Balanced Accuracies than SumBasic and LSA. Contrary to the former two measures, SumBasic and LSA do not utilize pre-trained word/sentence embeddings and rely only on count-based statistics in the input text. Thus, this result indicates that using embeddings trained

on larger corpora is effective to estimate discourse irreducibility and detect the nuclei, even though the embeddings have not been trained for nuclearity identification. Heuristic-Length achieves the best Balanced Accuracy among all the single measures. This result supports our hypothesis that a discourse constituent containing more words tends to be more irreducible and is more likely to be a nucleus.

We also explored various combinations of multiple measures with different weights through grid search. We fuse multiple measures by simply computing a weighted sum of the sentence irreducibility scores as follows:

$$\lambda_{m_1} g_1(s_i) + \lambda_{m_2} g_{m_2}(s_i). \tag{4.21}$$

$g_{m_1}(s_i)$ and $g_{m_2}(s_i)$ denote the sentence irreducibility scores predicted based on the measures m_1 and m_2, respectively. λ_{m_1} and λ_{m_2} are data-agnostic weights for each measure.

We found that the combination of the centroid-based measure (using the 100-dim GloVe embeddings; $\lambda = 0.8$) and Heuristic-Location ($\lambda = 0.2$) outperforms all the other methods and achieves the best Balanced Accuracy of 58.52% (See the bottom row of the second block), which is greater than the original scores of each measure, i.e., 56.00% and 50.00%. This result indicates that these measures estimate irreducibility of sentences in the different perspectives and that certain measures can be combined complementarily for further improvements. Actually, it is hard for the centroid-based measure to capture location-based irreducibility because it does not use localtional features in sentence scoring, while Heuristic-Location considers the locational information of sentences. The combination of these measures can thus accelerate performance. Interestingly, combining the centroid-based measure with Heuristic-Length fails to achieve a higher score, though Heuristic-Length yields higher accuracy than Heuristic-Location. We conjecture it happens because the centroid-based measure and Heuristic-Length are not complemental to each other, since the aggregated word embeddings of the centroid-based measure already contain information on sentence length.

The table also shows a classification upper bound of our approach. We observe that our approach has the potential to induce nuclearity statuses correctly only from sentence-level scalar values using the simple recursive algorithm. This result also suggests that it is possible to exploit irreducibility in supervised nuclearity classification, although it is out of the scope of the present work.

4.7 Summary

In this chapter, we introduced an unsupervised algorithm for discourse nuclearity classification, which aims to classify nuclearity statuses for an unlabeled discourse tree structure. To perform the deletion test of Carlson and Marcu (2001) automatically, we proposed discourse irreducibility scores and computed them by incorporating sentence-importance measures in extractive summarization with a simple recursive algorithm. We showed that the proposed irreducibility-based method outperforms the clustering-based methods. We also found that multiple complementary sentence-importance measures can be combined in our method to improve performance.

Chapter 5

Unsupervised Pre-training for Discourse Relation Classification

5.1 Motivation

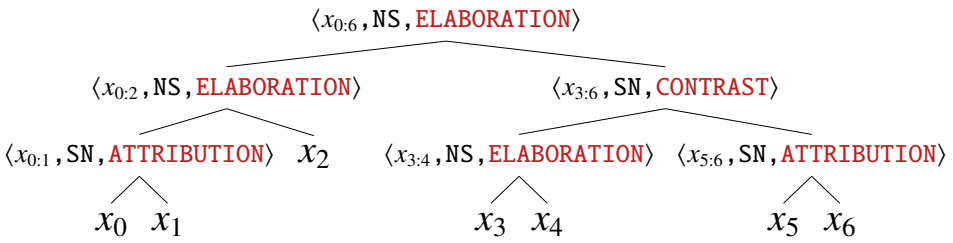
So far, we have proposed unsupervised algorithms for the first and second subtasks in our roadmap in Section 2.3. In this chapter[1], we tackle the last subtask, i.e., discourse relation classification. We again show a RST-based discourse tree structure we assume in this thesis in Figure 5.1.

When connectives such as *however* explicitly appear, discourse relations are relatively easy to classify, as connectives provide strong cues (Pitler et al., 2008). In contrast, it remains challenging to identify discourse relations across text spans that have no connectives.

One reason for this inferior performance is a shortage of labeled instances, despite the diversity of natural language discourses. Collecting annotations about implicit relations is highly expensive because it requires linguistic expertise. [2] A variety of semi-supervised or unsupervised methods have been explored to alleviate this issue. Marcu and Echihabi (2002) proposed generating synthetic instances by removing connectives from sentence pairs. This idea has been extended in many works and remains a core approach in the field (Zhou et al., 2010; Patterson and Kehler, 2013; Lan et al., 2013; Rutherford and Xue, 2015; Ji et al., 2015; Liu et al., 2016; Braud and Denis, 2016; Lan et al., 2017; Wu et al., 2017). However, these methods rely on automatically detecting connectives in unlabeled corpora beforehand,

[1] This chapter is mainly based on Nishida and Nakayama (2018).

[2] The Penn Discourse Treebank (PDTB) 2.0 corpus (Prasad et al., 2008), which is the current largest corpus for discourse relation recognition, contains only about 16K annotated instances in total.



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

Fig. 5.1 In this chapter, we assume that coherent text can be represented as a RST-like tree structure. Leaf nodes x_i correspond to clause-level segments, while internal nodes consists of three orthogonal elements: discourse constituents $x_{i:j}$, discourse nuclearities (e.g., NS), and discourse relations (e.g., ELABORATION).

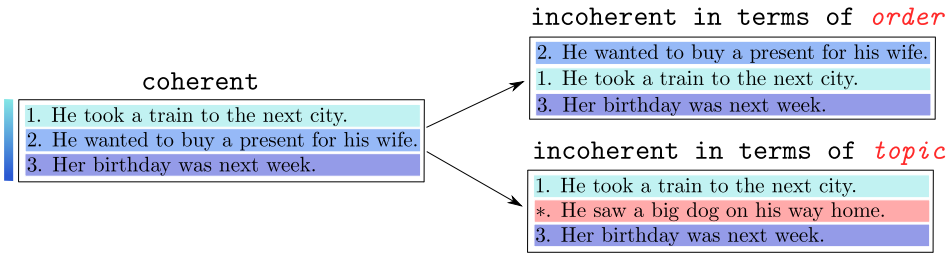


Fig. 5.2 An example of order-oriented and topic-oriented negative sampling in coherence modeling.

which makes it almost impossible to utilize parts of unlabeled corpora in which no connectives appear.[3] In addition, as Sporleder and Lascarides (2008) discovered, it is difficult to obtain a generalized model by training on synthetic data due to domain shifts. Though several semi-supervised methods do not depend on detecting connectives (Hernault et al., 2010a, 2011; Braud and Denis, 2015), these methods are restricted to manually selected features, linear models, or word-level knowledge transfer.

[3]For example, nearly half of the sentences in the British National Corpus hold implicit discourse relations and do not contain connectives (Sporleder and Lascarides, 2008).

In this chapter, our research question is how to exploit unlabeled corpora without explicitly detecting connectives to learn linguistic knowledge associated with implicit discourse relations.

Our core hypothesis is that unsupervised learning about text coherence could produce numerical representations related to discourse relations. Sentences that compose a coherent document should be connected with syntactic or semantic relations (Hobbs, 1985; Grosz et al., 1995). In particular, we expect that there should be latent relations among local sentences. In this study, we hypothesize that parameters learned through coherence modeling could contain useful information for identifying (implicit) discourse relations. To verify this hypothesis, we develop a semi-supervised system whose parameters are first optimized for coherence modeling and then transferred to implicit discourse relation recognition. We also empirically examine two variants of coherence modeling: (1) *order-oriented* negative sampling and (2) *topic-oriented* negative sampling. An example is shown in Figure 5.2.

Our experimental results demonstrate that coherence modeling improves Macro F_1 on implicit discourse relation recognition by about 3 points on first-level relation *classes* and by about 5 points on second-level relation *types*. Coherence modeling is particularly effective for relation categories with few labeled instances, such as temporal relations. In addition, we find that topic-oriented negative sampling tends to be more effective than the order-oriented counterpart, especially on first-level relation classes.

5.2 Coherence Modeling

In this study, we adopt the sliding-window approach of Li and Hovy (2014) to form a conditional probability that a document is coherent. That is, we define the probability that a given document X is coherent as a product of probabilities at all possible local windows, i.e.,

$$P(\text{coherent}|X,\theta) = \prod_{x \in X} P(\text{coherent}|x,\theta), \tag{5.1}$$

where $P(\text{coherent}|x,\theta)$ denotes the conditional probability that the local clique x is coherent and θ denotes parameters. Clique x is a tuple of a central sentence and its left and right sentences, (s_-, s, s_+). Though larger window sizes may allow the model to learn linguistic properties and inter-sentence

dependencies over broader contexts, it increases computational complexity during training and suffers from data sparsity problem.

We automatically build a dataset $\mathcal{D} = \mathcal{P} \cup \mathcal{N}$ for coherence modeling from an unlabeled corpus. Here, $\mathcal{P}$ and $\mathcal{N}$ denote sets of positive and negative instances, respectively. Given a source corpus C of $|C|$ sentences $s_1, s_2, \ldots, s_{|C|}$, we collect positive instances as follows:

$$\mathcal{P} = \{(s_{i-1}, s_i, s_{i+1}) \mid i = 2, \ldots, |C| - 1\}. \tag{5.2}$$

Text coherence can be corrupted by two aspects, which correspond to how to build negative set $\mathcal{N}$.

The first variant is ***order**-oriented negative sampling*, i.e.,

$$\mathcal{N} = \{x' \mid x' \in \phi(x) \wedge x \in \mathcal{P}\} \tag{5.3}$$

where $\phi(x)$ denotes the set of possible permutations of x, excluding x itself.

The second variant is ***topic**-oriented negative sampling*, i.e.,

$$\mathcal{N} = \{(s_-, s', s_+) \mid s' \in C \wedge \ (s_-, s, s_+) \in \mathcal{P}\} \tag{5.4}$$

where s' denotes a sentence randomly sampled from a uniform distribution over the entire corpus C. We call this method *topic-oriented* because topic consistency shared across a clique (s_-, s, s_+) is expected to be corrupted by replacing s with s'.

5.3 Model Architecture

We develop a simple semi-supervised model with neural networks. An overall view is shown in Figure 5.3. Our model mainly consists of three components: sentence encoder E, coherence classifier F_c, and implicit discourse relation classifier F_r. The parameters of E are shared across the two tasks: coherence modeling and implicit discourse relation recognition. In contrast, F_c and F_r are optimized separately. Though it is possible to develop more complex architectures (such as with word-level matching (Chen et al., 2016), a soft-attention mechanism (Liu and Li, 2016; Rönnqvist et al., 2017), or highway connections (Qin et al., 2016)), such architectures are outside the scope of this study, since the effectiveness of incorporating coherence-based knowledge would be broadly orthogonal to the model's complexity.

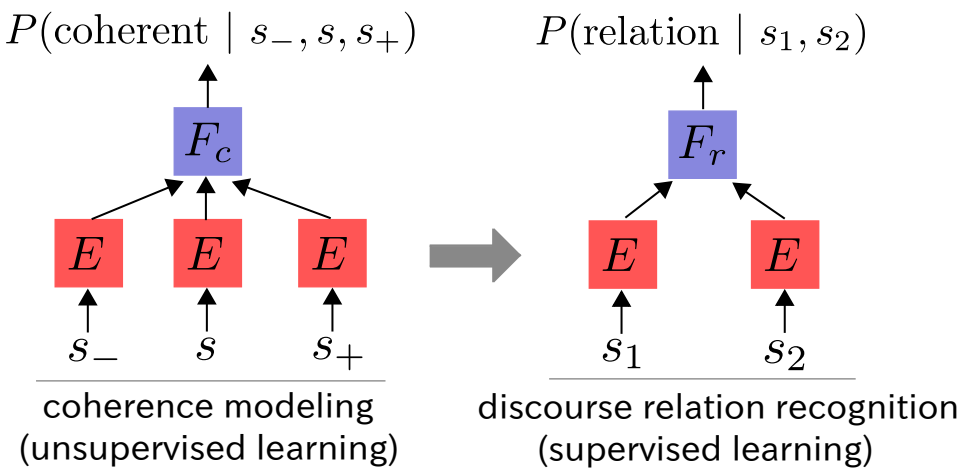


Fig. 5.3 The semi-supervised system we developed. The model consists of sentence encoder E, coherence classifier F_c, and implicit discourse relation classifier F_r.

5.3.1 Sentence Encoder

Sentence encoder E transforms a symbol sequence (i.e., a sentence) into a continuous vector. First, a bidirectional LSTM (BiLSTM) is applied to a given sentence of n tokens $w_1, \ldots, w_n$, i.e.,

$$\overrightarrow{\boldsymbol{h}}_i = \text{FwdLSTM}(\overrightarrow{\boldsymbol{h}}_{i-1}, w_i) \in \mathbb{R}^D, \tag{5.5}$$

$$\overleftarrow{\boldsymbol{h}}_i = \text{BwdLSTM}(\overleftarrow{\boldsymbol{h}}_{i+1}, w_i) \in \mathbb{R}^D \tag{5.6}$$

where FwdLSTM and BwdLSTM denote forward and backward LSTMs, respectively. We initialize the hidden states to zero vectors, i.e., $\overrightarrow{\boldsymbol{h}}_0 = \overleftarrow{\boldsymbol{h}}_{n+1} = \mathbf{0}$. In our preliminary experiments, we tested conventional pooling functions (e.g., summation, average, or maximum pooling); we found that the following concatenation tends to yield higher performances:

$$\boldsymbol{h} = \left(\overrightarrow{\boldsymbol{h}}_L^\top, \overleftarrow{\boldsymbol{h}}_1^\top\right)^\top \in \mathbb{R}^{2D}. \tag{5.7}$$

We use Eq. 5.7 as the aggregation function throughout our experiments.

5.3.2 Classifiers

We develop two multi-layer perceptrons (MLPs) with ReLU nonlinearities followed by softmax normalization each for F_c and F_r. The MLP inputs are the concatenation of sentence vectors. Thus, the dimensionalities of the

input layers are $2D \times 3$ and $2D \times 2$ respectively. The MLPs consist of `input`, `hidden`, and `output` layers.

5.4 Experiment Setup

We used the Penn Discourse Treebank (PDTB) 2.0 corpus (Prasad et al., 2008) as a dataset for implicit discourse relation recognition. We followed the standard section partition, which is to use Sections 2–20 for training, Sections 0-1 for development, and Sections 21–22 for testing. We evaluate multi-class classifications with first-level relation *classes* (4 classes) and second-level relation *types* (11 classes).

We used the Wall Street Journal (WSJ) articles (Marcus et al., 1993)[4] or the BLLIP North American News Text (Complete) (McClosky et al., 2008)[5] to build a coherence modeling dataset, resulting in about 48K (WSJ) or 23M (BLLIP) positive instances. We inserted a special symbol "⟨ARTICLE_BOUNDARY⟩" to each article boundary. For the WSJ corpus, we split the sections into training/development/test sets in the same way with the implicit relation recognition. For the BLLIP corpus, we randomly sampled 10,000 articles each for the development and test sets. Negative instances are generated following the procedure described in Section 5.2. Note that this procedure requires neither human annotation nor special connective detection.

We set the dimensionalities of the word embeddings, hidden states of the BiLSTM, and hidden layers of the MLPs to 100, 200, and 100, respectively. GloVe (Pennington et al., 2014) was used to produce pre-trained word embeddings on the BLLIP corpus. To avoid overfitting, we fixed the word embeddings during training in both coherence modeling and implicit relation recognition. Dropout (ratio 0.2) was applied to word embeddings and MLPs's layers. At every iteration during training in both tasks, we configured class-balanced batches by resampling.

5.5 Results and Discussion

To verify whether unsupervised learning on coherence modeling could improve implicit discourse relation recognition, we compared the semi-

[4]We used the raw texts in LDC99T42 Treebank-3: https://catalog.ldc.upenn.edu/LDC99T42

[5]https://catalog.ldc.upenn.edu/LDC2008T13

[6]The values are taken from Wu et al. (2017).

	4 Classes		11 Classes		Coherence
	Acc.	Macro F_1	Acc.	Macro F_1	Acc.
IRel only	51.49	42.29	37.49	24.81	N/A
IRel + *O-Coh* (Small)	52.16	41.39	37.77	25.46	57.96
IRel + *O-Coh* (Large)	52.29	42.48	41.29	**30.70**	**64.24**
IRel + *T-Coh* (Small)	51.70	40.84	37.91	25.35	83.04
IRel + *T-Coh* (Large)	**53.54**	**45.03**	**41.39**	29.67	**91.53**

Table 5.1 The results of implicit discourse relation recognition (multi-class classification) and coherence modeling (binary classification). *IRel* and *O/T-Coh* denote that the model is trained on implicit discourse relation recognition and order/topic-oriented coherence modeling respectively. "Small" and "large" correspond to the relative size of the used unlabeled corpus: 39K (WSJ) and 22M (BLLIP) positive instances, respectively.

	Acc. (%)	Macro F_1 (%)
Rutherford and Xue (2015)	57.10	40.50
Liu et al. (2016)	57.27	44.98
Braud and Denis (2016)[6]	52.81	42.27
Wu et al. (2017)	**58.85**	44.84
IRel only	51.49	42.29
IRel only*	52.72	42.61
IRel + *T-Coh* (Large)	53.54	**45.03**
IRel + *T-Coh* (Large)*	56.60	**46.90**

Table 5.2 Comparison with previous works that exploit unlabeled corpora on first-level relation *classes*. An asterisk indicates that word embeddings are fine-tuned (which slightly decreases performance on second-level relation *types* due to overfitting).

supervised model (i.e., implicit discourse relation recognition (***IRel***) + coherence modeling with order/topic-oriented negative sampling (***O/T-Coh***)) with the baseline model (i.e., *IRel* only). The evaluation metrics are accuracy (%) and Macro F_1 (%). We report the mean scores over 10 trials. Table 5.1 shows that coherence modeling improves Macro F_1 by about 3 points in first-level relation *classes* and by about 5 points in second-level relation *types*. Coherence modeling also outperforms the baseline in accuracy. We observed that the higher the coherence modeling performance (see Small vs. Large), the higher the implicit relation recognition score. This indicates that utilizing unlabeled data via coherence modeling improves the classification performance, which becomes possible thanks to our unsupervised proposal.

	Exp.	Cont.	Comp.	Temp.
# of training data	6,673	3,235	1,855	582
IRel only	66.40	53.49	39.48	32.31
IRel + *T-Coh*	**67.48**	**54.94**	**40.41**	**35.60**

Table 5.3 Results on one-vs.-others binary classification in implicit discourse relation recognition. The evaluation metric is Macro F_1 (%). We evaluate on the first-level relation *classes*: `Expansion`, `Contingency`, `Comparison`, and `Temporal`.

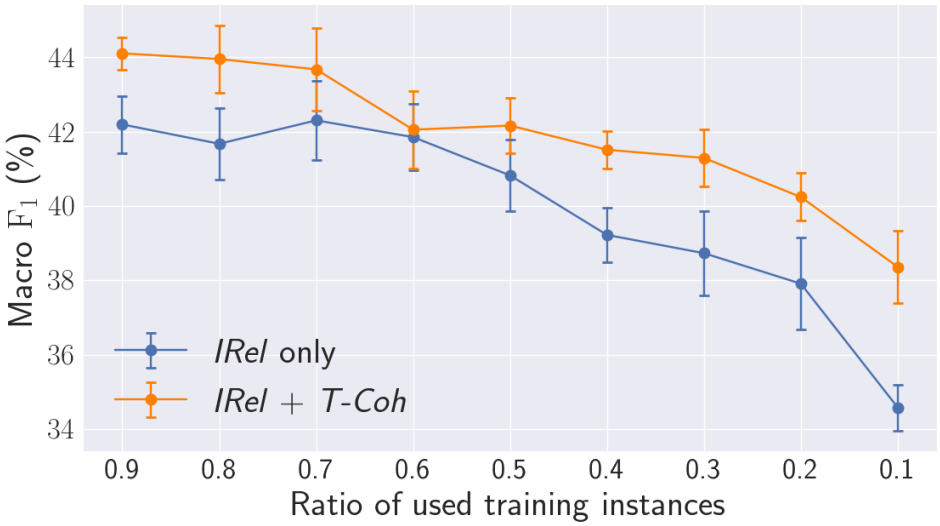


Fig. 5.4 Results on implicit discourse relation recognition (first-level *classes*), with different numbers of training instances. The error bars show one standard deviation over 10 trials.

Actually, coherence modeling allows us to use 39K (WSJ) or 22M (BLLIP) sentences for training in addition to the 1.2K instances in PDTB. These results support our claim that coherence modeling could learn linguistic knowledge that is useful for identifying discourse relations.

We also found that topic-oriented negative sampling tends to outperform its order-oriented counterpart, especially on first-level relation *classes*. We suspect that this is because order-oriented coherence modeling is more fine-grained and challenging than topic-oriented identification, resulting in poor generalization. For example, there could be order-invariant cliques that still hold coherence relations after random shuffling, whereas topic-invariant cliques hardly exist. Indeed, training on order-oriented negative sampling

converged to lower scores than that of topic-oriented negative sampling (see coherence accuracy).

Next, for reference, we compared our system with previous work that exploits unlabeled corpora. As shown in Table 5.2, we found our model to outperform previous systems in Macro F_1. In this task, Macro F_1 is more important than accuracy because the class balance in the test set is highly skewed. Note that these previous models rely on previously detected connectives in the unlabeled corpus, whereas our system is free from such detection procedures.

To assess the effectiveness of coherence modeling on different relation classes, we trained and evaluated the models on one-vs-others binary classification. That is, we treated each of the first-level relation *classes* (4 classes) as the positive class and others as the negative class. Table 5.3 shows that coherence modeling is effective, especially for the `Temporal` relation which has relatively fewer labeled instances than others, indicating that coherence modeling could compensate for the shortage of labeled data.

We also performed an ablation study to discover the performance contribution from coherence modeling by changing the number of training instances used in implicit relation recognition. Here, we assume that in real-world situations, we do not have sufficient labeled data. We downsampled from the original training set and maintained the balance of classes as much as possible. As shown in Figure 5.4, coherence modeling robustly yields improvements, even if we reduced the labeled instances to 10%.

5.6 Summary

In this chapter, we showed that unsupervised learning on coherence modeling improves implicit discourse relation recognition in a semi-supervised manner. Our approach does not require detecting explicit connectives, which makes it possible to exploit entire unlabeled corpora. We empirically examined two variants of coherence modeling and show that topic-oriented negative sampling tends to be more effective than the order-oriented counterpart on first-level relation *classes*.

It still remains unclear whether the coherence-based knowledge is complemental to those by previous work. It is also interesting to qualitatively inspect the differences of learned properties between order-oriented and

topic-oriented negative sampling. We will examine this line of research in future.

Chapter 6

Conclusions

6.1 Summary of the Thesis

In this thesis, we introduced unsupervised discourse parsing algorithms that aim to induce discourse structures without relying on hand-annotated structures. To the best of our knowledge, this thesis is the first attempt to develop a fully unsupervised discourse parser. The only assumption of this thesis is that coherent text can be represented as a RST-based discourse tree structure, which contains three orthogonal elements at its internal nodes: discourse constituents, discourse nuclearities, and discourse relations.

We first broke down the discourse structures into smaller subtasks, each corresponding to one of the three orthogonal elements. Then, we proposed unsupervised algorithms for the tree subtasks based on our prior knowledge.

In the first subtask, we found that our unsupervised parser using Viterbi EM and the margin-based criterion induces more accurate discourse constituents than fully supervised parsers. We also confirmed the importance of the initialization methods we proposed based on our prior knowledge of document structures. In the second subtask, we found that the proposed method using discourse irreducibility identifies relative importance between text spans more accurately than the straight-forward baselines. We also found that multiple sentence-importance measures can be complementarily combined to improve performance. In the third subtask, we found that our approach using topic-oriented coherence modeling identifies implicit relations more accurately than the existing methods.

Totally, our unsupervised algorithms outperform and improve the unsupervised (or even fully supervised) baselines in all the three subtasks. These results indicate that it is possible to develop unsupervised algorithms based

on prior knowledge of discourse structures and that the prior knowledge used in this thesis captures somewhat reasonable aspects of discourse structures.

However, these results simultaneously indicate that our algorithms are strongly dependent on the prior knowledge, which is empirically shown in our analytic experiments. To develop unsupervised algorithms that induce discourse elements more accurately and diversely, we need to observe the characteristics of the target discourse elements more carefully. For example, to improve unsupervised discourse constituency parsing (i.e., the first subtask) for left-heavy discourse constituents, we need to explore more detailed but general prior knowledge of left-heavy discourse constituents and to implement the knowledge as an initial-tree sampling procedure in Viterbi EM. To improve unsupervised discourse nuclearity classification (i.e., the second subtask), we need to introduce new prior knowledge that correlates with nuclearity in a different way with discourse irreducibility. We believe that breaking down the discourse parsing problem into smaller subtasks (focusing on orthogonal discourse elements) is helpful to consider suitable prior knowledge of each discourse element.

6.2 Limitations and Future Work

We have two limitations in this research.

First, this thesis assumes that one tree structure can be derived from coherent text. However, although tree structures are widely used in a variety of discourse theories (Hobbs, 1985; Grosz and Sidner, 1986; Mann and Thompson, 1988; Asher and Lascarides, 2003; Webber, 2004) and have been proven to be useful in many applications, there are criticisms to RST-style tree structures (Moore and Pollack, 1992; Wolf and Gibson, 2005). For example, Wolf and Gibson (2005) argue that graph structures are more suitable than tree structures to represent discourse coherence.

Another limitation of this thesis is that our study uses only English texts for experiments. However, different languages could have different discourse tendencies. For example, Shimmura and Beteson (1999) discussed the discourse pattern differences between English and Japanese. Hence, it would be interesting to investigate whether our prior knowledge and the proposed methods work uniformally across different languages, which we believe gives suggestions on discourse-level universals.

We leave these issues as a future work.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth Trippe, Juan Gutierrez, and Krys Kochut. 2017. Text summarization techniques: A brief survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 8:397–405.

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

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

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

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

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

Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, Mitch Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the Feburary 1991 DARPA Speech and Natural Language Workshop*.

Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Chloé Braud and Pascal Denis. 2016. Learning connective-based word representations for implicit discourse relation identification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Chloé Braud, Ophelie Lacroix, and Anders Søgaard. 2017. Does syntax help discourse segmentation? Not so much. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Kay H. Brodersen, Cheng Soon Ong, Klaas E. Stephan, and Joachim M. Buhmann. 2010. The balanced accuracy and its posterior distribution. In *ICPR*.

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

Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. In *Technical Report ISI-TR-545*. University California Information Sciences Institute.

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

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Eugene Charniak. 1993. Statistical language learning.

Jifan Chen, Qi Zhang, Pengfei Liu, and Xuanjing Huang. 2016. Discourse relations detection via a mixed generative-discriminative framework. In *Proceedings of the 30th Conference on Artificial Intelligence*.

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

Norm Chomsky. 1975. *Reflections on Language*. Random House.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the 2001 workshop on Computational Natural Language Learning*.

Edward Crothers. 1979. *Paragraph Structure Inference*. Ablex Publishing Corporation.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391.

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

Güneş Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

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

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Charles Fillmore. 1974. Pragmatics and the description of discourse. *Berkeley Studies in Syntax and Semantics*, 1:V–1–V–21.

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

Kevin Gimpel and Noah A. Smith. 2012. Concavity and initialization for unsupervised dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

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

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *SIGIR*.

Joseph Grimes. 1975. *The Thread of Discourse*. Mouton and Company.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistis*, 21(2):203–225.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistis*, 12(3):175–204.

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

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Hermann Hendriks. 2002. *Information Packaging: From Cards to Boxes ser. Information Sharing: Reference and Presupposition in Language Generation and Interpretation*. CSLI.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2010a. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2011. Semi-supervised discourse relation classification with structure learning. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing*.

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

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference of Empirical Methods in Natural Language Processing*.

Jerry R. Hobbs. 1985. On the coherence and structure of discourse. Technical Report CSLI-85-37, Center for the Study of Language and Information (CSLI), Stanford University.

Eduard Hovy and Elisabeth Maier. 1995. Parsimonious or profligate: How many and which discourse relations? In *Technical Report*. University of Southern California.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

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

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Yangfeng Ji, Gongbo Zhang, and Jacob Eisenstein. 2015. Closing the gap: Domain adaptation from explicit to implicit discourse relations. In *Proceedings of the 53st Annual Meeting of the Association for Computational Linguistics*.

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

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

Aravind K. Joshi and Yves Schabes. 1997. *Tree-Adjoining Grammars*. Springer.

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

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

Hans Kamp and Uwe Reyle. 1993. *From Discourse to the Lexicon: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Springer Netherlands.

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

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*.

Dan Klein. 2005. The unsupervised learning of natural language structure. *Ph.D. Thesis*.

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

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

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

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

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

Man Lan, Jianxiang Wang, Yuanbin Wu, Zheng-yu Niu, and Haifeng Wang. 2017. Multi-task attention-based neural networks for implicit discourse relationship representation and identification. In *Proceedings of the 2017 Conference of Empirical Methods in Natural Language Processing*.

Man Lan, Yu Xu, and Zhengyu Niu. 2013. Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

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

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014a. Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference of Empirical Methods in Natural Language Processing*.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference of Empirical Methods in Natural Language Processing*.

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

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptation of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yang Liu and Sujian Li. 2016. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. In *Proceedings of the 30th Conference on Artificial Intelligence*.

Robert Longacre. 1976. *An Anatomy of Speech Notions*. The Peter de Ridder Press.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605.

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

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

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

Daniel Marcu. 1999. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

Daniel Marcu. 2000a. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3):395–448.

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

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

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

David Mareček and Zdeněk Žabokrtský. 2012. Exploiting reducibility in unsupervised dependency parsing. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing*.

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

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

David McClosky, Eugene Charniak, and Mark Johnson. 2008. BLLIP north american news text, complete. *Linguistic Data Consortium*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

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

Johanna D. Moore and Martha E Pollack. 1992. A problem for RST: The need for multiple-level discourse analysis. *Computational Linguistics*, 18(4):537–544.

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

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

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

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

Ani Nenkova and Kathleen McKeown. 2012. *A Survey of Text Summarization Techniques*. Springer.

Noriki Nishida and Hideki Nakayama. 2017. Word ordering as unsupervised learning towards syntactically plausible word representations. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*.

Noriki Nishida and Hideki Nakayama. 2018. Coherence modeling improves implicit discourse relation recognition. In *Proceedings of the 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Noriki Nishida and Hideki Nakayama. to appear. Unsupervised discourse constituency parsing using Viterbi EM. *Transactions of the Association for Computational Linguistics*.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*.

Gary Patterson and Andrew Kehler. 2013. Predicting the presence of discourse connectives. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

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

Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind Joshi. 2008. Easily identifiable discourse relations. In *Proceedings of the 24th International Conference on Computational Linguistics*.

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

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

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*.

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016. A stacking gated neural architecture for implicit discourse relation classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938.

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

E. Reiter and R. Dale. 1997. Building applied natural language generation system. *Natural Language Engineering*, 3(1):57–88.

Samuel Rönnqvist, Niko Schenk, and Christian Chiarcos. 2017. A recurrent neural model with attention for the recognition of chinese implicit discourse relations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compotitionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*.

Attapol T. Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the 9th International Workshop on Parsing Technology*.

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

Tomoko Shimmura and Gordon Beteson. 1999. Comparatively speaking: Extended oral comparisons in english and japanese. *Journal of the Department of Literature at Kanazawa Gakuin University*.

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

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

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Radu Soricut and Daniel 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*.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010. Viterbi training improves unsupervised dependency parsing. In *Proceedings of the 14th Conference on Computational Natural Language Learning*.

Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(03).

Josef Steinberger, Massimo Poesio, Mijail A. Kabadjov, and Karel Ježek. 2007. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6).

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

Simone Teufel and Marc Moens. 2001. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4).

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Manegement*, 43(6):1606–1618.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *SIGIR'07*.

Kimberly Voll and Maite Taboada. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In *Proceedings of Australian Conference on Artificial Intelligence*.

Yizhong Wang, Sujian Li, and Honfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

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

Bonnie Webber. 2004. D-LTAG: Extending lexicalized TAG to discourse. *Cognitive Science*, 28(5):751–779.

Florian Wolf and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational, Linguistics*, 31(2):249–287.

Changxing Wu, Xiaodong Shi, Yidong Chen, Jinsong Su, and Boli Wang. 2017. Improving implicit discourse relation recognition with discourse-specific word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

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

Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics*.