Linguistically-Inspired Neural Coherence Modeling

Wei Liu

Department of Computational Linguistics
Heidelberg University

This dissertation is submitted for the degree of
Doctor of Philosophy

First examiner: Prof. Dr. Michael Strube
Second examiner: Prof. Dr. Anette Frank
Third examiner: Prof. Dr. Amir Zeldes

Submission date: 05.08.2025.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Michael Strube, for giving me the opportunity to pursue my PhD at HITS and for his invaluable guidance over the past four years. I am especially thankful for his encouragement to present more frequently at our colloquium, which significantly helped me to improve my presentation skills and academic confidence.

I am also deeply grateful to Prof. Dr. Anette Frank and Prof. Dr. Amir Zeldes for kindly agreeing to serve as the reviewers of my thesis.

I would like to extend my heartfelt thanks to all my colleagues in the NLP group at HITS: Mehwish Fatima, Sungho Jeon, Haixia Chai, Yi Fan, Souvik Banerjee, Shimei Pan, and Stephen Wan. I will always cherish the fun and insightful conversations, especially with Sungho, Yi, and Souvik.

Special thanks also go to the wonderful staff at HITS. I would like to thank Frauke Bley for her kind assistance with my visa application, and Silvia Galbusera for her continuous support throughout my PhD journey. I will always remember Silvia as a kind, warm-hearted, and elegant Italian lady. I also thank Harald Haas for sharing many interesting stories, especially those about football (although I know little about the sport). I am grateful to Jose Avila for the delicious meals and his unfailing kindness over the years.

I would also like to thank my colleagues from my internship at Amazon Berlin: Adrián Bazaga, Bill Byrne, Dawei Zhu, Felix Hieber, Leonardo F. R. Ribeiro, Luke Ablonczy, Sony Trenous, Tobias Domhan, and Zachary Hille. In particular, I thank Felix, who taught me how to deliver effective presentations in the company and how to collaborate within a team. I am also grateful to Sony for helping me navigate Amazon's platforms and for supporting the design of our annotation guidelines. Bill and Leonardo provided crucial guidance on how to structure our work and present it clearly in the paper. I will always treasure the time I spent at Amazon.

Finally, I owe my deepest thanks to my family. My parents, Caichang Liu and Xinying Zeng, though both farmers with limited formal education, have always supported and encouraged me to pursue my studies and explore the world beyond our hometown. During my PhD, I was away from home for nearly four years. I deeply regret not being able to visit my

parents, and I am sincerely grateful for their love and understanding. I also thank my sisters, Yanqing Liu and Yanmei Liu, and my brother-in-law, Houhua Xiong, for being there with my parents in my absence. Lastly, I thank my girlfriend, Xiyan Fu, for her unwavering love and support throughout my PhD journey.

Abstract

Coherence is an essential property of well-written text, making it easier to read and understand than a sequence of randomly arranged sentences. Assessing text coherence is valuable for many tasks. For example, it can be used to automatically score documents, reducing manual effort, or to provide feedback to students, helping them improve their writing quality. It can also serve as a reward model for training Large Language Models (LLMs) to generate more coherent and natural text. Given the importance of the task, many methods have been proposed for coherence modeling. Among these approaches, the dominant ones are neural network-based models due to their strength in representation learning and feature combination.

In linguistics, many factors contribute to achieving textual coherence. For example, text coherence can be achieved by describing the same set of entities or using discourse relations between sentences. However, existing work on neural coherence modeling focuses on using more powerful encoders or solely entity information, without a systematic analysis of the benefits of different linguistic features. In this thesis, we investigate the importance of entity- and relation-based patterns for coherence assessment and develop novel approaches to utilize these features individually or jointly.

We first investigate the benefits of entity-based patterns for coherence modeling. We analyze previous work that has leveraged entity patterns for coherence assessment. Then, we introduce a novel graph-based approach that captures the similarity of entity transition patterns between documents, rather than limiting the modeling of these patterns within a single document. We evaluate this approach on multiple benchmarks, and the results demonstrate that it outperforms various baselines.

Next, we examine the role of discourse relations in coherence modeling. Existing discourse parsers struggle with implicit discourse relation classification, limiting the use of discourse relations in coherence assessment. To address this, we propose a novel framework that jointly generates a connective between arguments and predicts discourse relations based on both the arguments and the generated connectives. Experiments show that our joint model achieves state-of-the-art performance on the PDTB 2.0, PDTB 3.0, and PCC datasets.

Beyond proposing a novel model for implicit discourse relation classification, we also investigate an unanswered question in the discourse processing community: why do relation classifiers trained on explicit examples (with connectives removed) perform poorly in real implicit scenarios? We identify label shift caused by the removal of connectives as a key factor contributing to this failure. To support this finding, we provide both manual analysis and corpus-level empirical evidence. Additionally, we propose two strategies to mitigate the impact of label shift.

Using the improved discourse parser, we identify discourse relations within documents and empirically demonstrate their correlation with textual coherence. Based on this observation, we develop a novel fusion model that integrates discourse relation-based features into a pre-trained model for coherence modeling.

Finally, we explore combining entity-based and discourse relation-based features for coherence modeling. This approach is motivated by the observation that writers typically employ multiple strategies simultaneously to ensure coherence. To this end, we design two methods to jointly model entities and discourse relations for coherence assessment. Experimental results demonstrate that both approaches significantly outperform models that use either features in isolation, highlighting the importance of considering both types of features simultaneously.

Table of Contents

1 Introduction **1**

1.1 Motivation 1
1.2 Research Questions 3
1.3 Contributions 4
1.4 Thesis Overview 6
1.5 Published Work 8

2 Background **11**

2.1 Coherence 11
2.1.1 Entity-based Coherence 12
2.1.2 Topic-based Coherence 14
2.1.3 Discourse Relation-based Coherence 14
2.2 Tasks and Corpora 21
2.2.1 Coherence Modeling 21
2.2.2 Discourse Relation Classification 26
2.3 Deep Learning in NLP 29
2.3.1 Transformer 29
2.3.2 Graph Neural Networks 33
2.3.3 Pre-trained Language Models & Large Language Models 34
2.3.4 Model Adaptation 36
2.3.5 MASK Stragegy in Transformers 38

3 Related Work **41**

3.1 Coherence Modeling 41
3.1.1 Entity-based Methods 41
3.1.2 Discourse Relation-based Methods 47
3.2 Discourse Relation Classfication 49

4 Document Structure Similarity-Enhanced Coherence Modeling 57
4.1 Why Consider the Structural Similarity? . . . 58
4.2 Graph-based Method . . . 59
4.2.1 Sentence Graph . . . 59
4.2.2 Subgraph Set . . . 61
4.2.3 Doc-subgraph Graph . . . 62
4.2.4 GCN Encoder . . . 63
4.3 Experiments . . . 64
4.3.1 Datasets . . . 64
4.3.2 Experimental Settings . . . 65
4.3.3 Overall Results . . . 66
4.3.4 Performance Analysis . . . 68
4.3.5 Ablation Study . . . 69
4.3.6 Subgraph Analysis . . . 71
4.4 Summary . . . 72

5 Annotation-inspired Implicit Discourse Relation Classification 73
5.1 Why Is Implicit Relation Classification Challenging? . . . 74
5.2 The Annotation Process of Implicit Relations . . . 75
5.3 An Annotation-inspired Model . . . 75
5.3.1 Connective Generation . . . 76
5.3.2 Relation Classification . . . 77
5.3.3 Training and Evaluation . . . 78
5.4 Experiments . . . 79
5.4.1 Experimental Settings . . . 80
5.4.2 Overall Results . . . 82
5.4.3 Performance Analysis . . . 84
5.4.4 Relation Analysis . . . 87
5.4.5 Ablation Study . . . 88
5.5 Summary . . . 89

6 Explicit to Implicit Discourse Relation Classification 91
6.1 Background . . . 92
6.1.1 Task . . . 92
6.1.2 Datasets . . . 93
6.1.3 The Performance Gap . . . 94
6.2 Label Shift in Discourse Relations . . . 94

6.2.1 What Is Label Shift? . . . 94
6.2.2 Do Explicit Examples Suffer from Label Shift? . . . 95
6.2.3 Does Label Shift Exist at the Corpus Level? . . . 96
6.2.4 Can Label Shift Be Measured? . . . 99
6.2.5 Why Does Label Shift Happen? . . . 99
6.3 Strategies to Alleviate Label Shift . . . 101
6.3.1 Filter Out Noisy Examples . . . 102
6.3.2 Joint Learning with Connectives . . . 102
6.4 Experiments . . . 103
6.4.1 Baselines and Upper Bounds . . . 104
6.4.2 Overall Results . . . 105
6.4.3 Results on the GUM Dataset . . . 107
6.5 Summary . . . 107

7 Discourse Relation-Enhanced Coherence Modeling 109
7.1 Discourse Relation and Coherence . . . 110
7.1.1 Discourse Relations . . . 110
7.1.2 Correlation Analysis . . . 111
7.1.3 Text vs. Relations . . . 113
7.2 Discourse Relation-Enhanced Fusion Model . . . 115
7.2.1 Flat Structure with Positions . . . 116
7.2.2 Position-aware Attention . . . 117
7.2.3 Visibility Matrix . . . 118
7.3 Experiments . . . 119
7.3.1 Experimental Settings . . . 119
7.3.2 Overall Results . . . 121
7.3.3 Performance Analysis . . . 123
7.3.4 Ablation Study . . . 124
7.4 Summary . . . 124

8 Coherence Modeling Using Entities and Discourse Relations 125
8.1 Method . . . 126
8.1.1 Method I: Fusion . . . 127
8.1.2 Method II: Prompt . . . 129
8.2 Experiments . . . 131
8.2.1 Experimental Settings . . . 131
8.2.2 Overall Results . . . 132

8.2.3 Analysis 135
8.3 Summary 138

9 Conclusions & Future Work **139**
9.1 Conclusions 139
9.2 Future Work 140

Appendix **143**
A Structural-similarity Enhanced Coherence Modeling 143
A.1 Subgraph Examples 143
B Annotation-Inspired Implicit Relation Classification 143
B.1 Experimental Settings 143
C Explicit to Implicit Discourse Relation Classification 145
C.1 Manual Analysis 145
D Joint Modeling of Entities and Discourse Relations 145
D.1 Prompt with Explanation 145
D.2 Zero-shot Results Using Llama-3.3-70B 146
E Code and Data Used in this Thesis 149

List of Figures **151**

List of Tables **157**

References

Chapter 1

Introduction

1.1 Motivation

Coherence is an important aspect of text quality, which describes how sentences of a text connect to each other. Sentences in a coherent text are usually logically connected rather than randomly assembled (Jurafsky and Martin, 2025). Consider the two examples below, each consisting of three sentences. Example (1.1) is highly coherent because all sentences are about "John" and "piano". In contrast, Example (1.2) lacks coherence, as the sentences shift between unrelated topics. Specifically, while the first sentence discusses "John" and "piano", the second suddenly introduces "dog" and "big house", and the third shifts again to a different subject.

(1.1) [John wanted to buy a piano.]$_{s1}$ [He went to a piano store.]$_{s2}$ [He picked up one and paid for it.]$_{s3}$

(1.2) [John wanted to buy a piano.]$_{s1}$ [The dog lived in a big house.]$_{s2}$ [Mary likes Chinese food.]$_{s3}$

Coherence modeling is the task of assessing the coherence of a given text. It is beneficial for various NLP applications. For instance, it can be used to automatically evaluate document quality (Farag et al., 2018), provide feedback on student writing (Sarzhoska-Georgievska, 2016), or serve as a reward model in training large language models (LLMs) to generate more coherent outputs (Kwon et al., 2023). Given its importance, many methods for coherence modeling have been proposed over the years. Early work in this area is dominated by entity-based approaches. For example, the entity grid (Barzilay and Lapata, 2008) captures the entity transition between adjacent sentences of a text to model local coherence, while the entity graph (Guinaudeau and Strube, 2013) measures coherence using the entity connection

structure of a document. More recently, neural models (Li and Hovy, 2014; Li and Jurafsky, 2017; Mesgar and Strube, 2018; Xu et al., 2019; Jeon and Strube, 2020a; Mesgar et al., 2021) have been applied to the task due to their strength in representation learning and feature combination. Those models learn a document's representation from word embeddings or pre-trained language models, significantly outperforming previous statistical methods. While achieving impressive results, these neural models have paid little attention to linguistic features associated with text coherence.

In linguistics, text coherence can be achieved in multiple ways, with two of the most common being entity-based and discourse relation-based coherence (Jurafsky and Martin, 2025). In the former, coherence is established by discussing a set of related entities throughout the text, as illustrated in Example (1.1). By contrast, the latter relies on discourse relations between sentences to achieve coherence. For instance, Example (1.3) is considered highly coherent due to its well-structured discourse relations: a *Contrast* relation links the first two sentences, an *Instantiation* relation provides additional details about the strike, and a *Cause* relation introduces the final sentence.

(1.3) [Tom was late for the meeting this morning.]$_{s1}$ [However, it was not his fault but rather due to the citywide strike.]$_{s2}$ [All the roads were blocked, and the buses were canceled.]$_{s3}$ [Therefore, he had to walk to the office, which took a lot of time.]$_{s4}$

Existing neural models have attempted to incorporate entity-based features by modeling entity transitions within a document using architectures such as convolutional networks (Tien Nguyen and Joty, 2017) or long short-term memory networks (Mesgar and Strube, 2018). Some subsequent studies have expanded the entity set to include topically related words (Mesgar and Strube, 2016; Jeon and Strube, 2020a, 2022). However, these efforts primarily focus on extracting entity-based patterns within individual documents and do not account for correlations between documents. Another limitation of existing work is the lack of investigation into whether discourse relations are helpful to coherence modeling. This gap is caused by the poor performance of available discourse parsers (Lin et al., 2014), particularly in classifying implicit discourse relations.

This thesis aims to address these limitations by systematically investigating the role of linguistic features in coherence assessment. First, we examine whether similarities in entity transition patterns between documents can enhance coherence modeling. Next, we turn to discourse-based features, proposing a novel approach to improve the classification of implicit discourse relations. In this context, we analyze the influence of discourse connectives on relation classification and evaluate the contribution of discourse relations to coherence assessment. Finally, we adopt a joint modeling perspective, combining both entity-based and discourse relation-based features for coherence assessment.

1.2 Research Questions

For a better overview, we group our research on coherence modeling into three research questions (RQ1-3):

- **RQ1: Does structural similarity between documents contribute to coherence assessment?**
  Centering Theory (Grosz et al., 1995), the most influential theory of entity-based coherence, models local coherence by capturing entity transitions between adjacent sentences in a text. Inspired by this theory, many works (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013; Tien Nguyen and Joty, 2017; Jeon and Strube, 2020a, 2022) have focused on extracting entity-based features for coherence assessment. However, these approaches primarily analyze features within a single document, overlooking potential correlations between documents. Given that coherence describes how sentences within a text are connected (Schwarz, 2001), we hypothesize that texts with similar entity structures should exhibit similar degrees of coherence. This leads us to investigate whether we can develop an enhanced method that explicitly models structural similarities between documents for coherence assessment.

As mentioned before, coherence can be achieved not only by discussing a set of related entities but also by forming meaningful discourse relations between sentences. This brings us to the second research question:

- **RQ2: Can an improved discourse parser assist in coherence modeling?**
  Discourse coherence theories posit relations between text spans as a key feature of coherent texts (Rohde et al., 2018). However, existing work on coherence modeling has paid little attention to discourse relations. One major reason is the limited performance of existing discourse parsers (Lin et al., 2014), particularly in classifying implicit discourse relations. Inaccurate parsing results can lead to misleading conclusions about the role of discourse relations in coherence modeling. To address this issue, it is crucial to develop a better understanding of both explicit and implicit discourse relations, especially the role of discourse connectives, and to design more effective parsers for extracting discourse relations from documents:
    - **RQ2(a)**: How can we design a novel model to enhance the performance of implicit discourse relation classification?
    - **RQ2(b)**: Why do classifiers trained on explicit examples perform poorly in real implicit scenarios?

By addressing the above two subquestions, we can better understand discourse connectives for discourse relation classification and obtain an improved implicit relation classifier. This enables us to investigate:

- **RQ2(c)**: Can we enhance coherence modeling by carefully designing methods to leverage discourse relations parsed from documents?

Once we complete the study on coherence modeling based on entities and discourse relations, we can take it a step further and explore the joint modeling of these two types of features. In other words, we aim to investigate:

- **RQ3: Can the combination of entity and discourse relations further boost the performance of coherence modeling?**
  While entity-based and discourse relation-based methods have proven effective individually, real-world texts often require a more integrated view. In practice, entity and discourse relation cues frequently coexist and interact in complex ways. To illustrate this, consider Example (1.4), which consists of four sentences and is considered highly coherent. Establishing the coherence using entities is not straightforward in this case, as there are no overlapping entities between the second and third sentences. Instead, we must use a more complex linguistic phenomenon, namely bridging (Clark, 1975), to link "city" (in "citywide") and "road". Meanwhile, the connection between these sentences is more readily explained by a discourse relation (e.g., Instantiation), as the third sentence elaborates on the strike mentioned earlier. However, relying solely on discourse relations also has limitations, as it can compromise the smooth tracking of the protagonist if the referents are unclear. For example, if the final sentence were changed to "So, Maria couldn't get to the airport...", the discourse relation might still hold, but the referent switch (i.e., John $\rightarrow$ Maria) would disrupt the overall coherence.

  (1.4) [Did you know that John is still in Germany?]$_{s1}$ [He was planning to leave Berlin today but ran into a citywide strike.]$_{s2}$ [All the roads were blocked, and buses and trains were cancelled.]$_{s3}$ [So, he couldn't get to the airport and now has to stay in the city for a few more days.]$_{s4}$

  Therefore, we must consider both entities and discourse relations simultaneously for more effective coherence modeling.

1.3 Contributions

In sum, our main contributions are:

- **To answer RQ1**:
  - We present a graph-based approach for coherence modeling that connects structurally similar documents through a graph and leverages a GCN to learn representations of documents while considering connections between them.
  - We evaluate our method on two tasks, assessing discourse coherence and automated essay scoring, and show that our graph-based approach outperforms strong baselines.
  - We perform detailed analyses to show that structural similarity information helps to mitigate the effects of uneven label distributions in datasets and improve the model's robustness across documents of different lengths.
- **To answer RQ2**, we conduct three studies, each addressing a sub-question:
  - **To answer RQ2(a)**:
    - ▶ We design a joint training approach that leverages discourse connectives for implicit discourse relation classification, inspired by the human annotation of implicit relations.
    - ▶ We evaluate our model on two English corpora and a German corpus, and show that our connective-enhanced model significantly outperforms previous relation classifiers.
    - ▶ We show that the end-to-end training characteristics enable our model to learn a good balance between arguments and connectives for implicit relation classification.
  - **To answer RQ2(b)**:
    - ▶ We identify label shift as one cause for the failure of explicit to implicit discourse relation classification.
    - ▶ We manually analyze 100 examples to show the existence of label shift when removing connectives from explicit examples.
    - ▶ We develop an empirical method to verify the occurrence of label shift at the corpus level.
    - ▶ We investigate various factors contributing to this phenomenon and find that the syntactic role of the connective has the most significant impact.
    - ▶ We propose two strategies to mitigate label shift: filtering out noisy training instances and joint learning with connectives.

        - We show that these strategies achieve significant improvements over strong baselines for explicit to implicit relation classification.
    - **To answer RQ2(c)**:
        - We demonstrate that discourse relations parsed from documents are highly correlated with text coherence.
        - We propose a novel fusion model to combine text- and relation-based features for coherence assessment.
        - Experimental results demonstrate that incorporating discourse relations significantly enhances the model's performance in both in-domain and cross-document evaluations.
- **To answer RQ3**:
    - We propose two methods for jointly modeling entities and discourse relations to assess coherence.
    - We show that models leveraging both entity and discourse relation features consistently outperform those that rely on only one or neither.
    - We demonstrate that models combining entities and discourse relations can learn more robust coherence patterns across different domains.

1.4 Thesis Overview

After discussing the background and related work in Chapters 2 and 3, the main content of this thesis is structured into three parts. Part I (Chapter 4) explores entity-based features for coherence modeling. Part II examines discourse relation-based features for coherence assessment, including improving discourse parsing for implicit relations (Chapter 5), analyzing the role of connectives in explicit and implicit relations (Chapter 6), and evaluating the benefits of discourse relations for coherence modeling (Chapter 7). Part III (Chapter 8) investigates the combination of entity- and discourse relation-based features for coherence modeling.

Preliminaries: Background and Related Work

In **Chapter 2**, we begin by describing key concepts such as coherence, discourse relation, and other relevant terms used throughout this thesis. Next, we introduce neural network techniques used in this thesis, such as the Transformer and Graph Convolutional Networks.

In **Chapter 3**, we review existing works related to the research topic of this thesis. Specifically, we discuss prior studies on coherence modeling and point out their shortcomings.

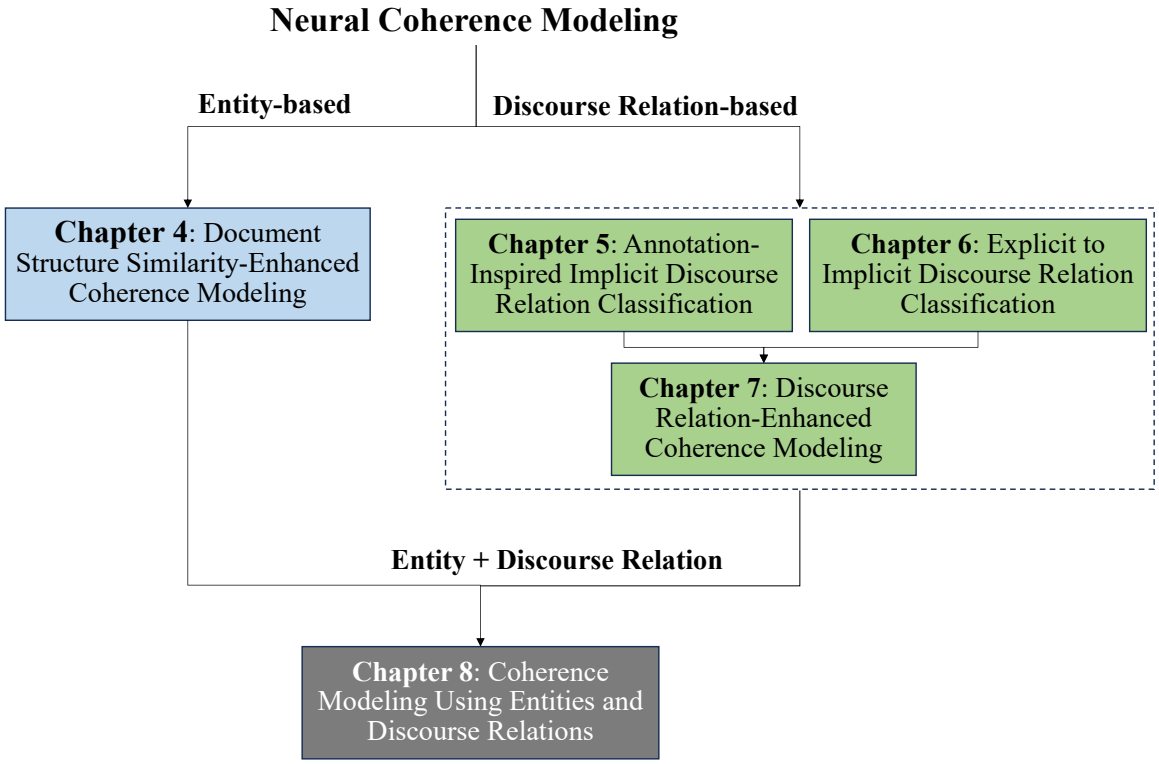


Fig. 1.1 Overview of this thesis. It comprises three components: entity-based coherence modeling, discourse relation-based coherence modeling, and a combined approach that integrates both types of features.

These works will serve as baselines throughout this thesis. We also summarize the key challenges in implicit discourse relation classification and discuss previous efforts to address these challenges.

Part I: Coherence Modeling with Entity-based Features

In **Chapter 4**, we introduce the motivation for using structural similarities between documents for coherence assessment, present a graph-based approach, and validate its effectiveness through extensive experiments and analyses.

Part II: Coherence Modeling with Discourse Relation-based Features

In **Chapter 5**, we first describe why implicit discourse relation classification is challenging and present how humans annotate such relations in the Penn Discourse Treebank. We then propose a novel approach that jointly learns to generate a connective between arguments and predicts a discourse relation based on both arguments and the generated connective. Finally, we evaluate our model on two English corpora and one German corpus.

In **Chapter 6**, we identify label shift as one cause for the poor performance of explicit to implicit discourse relation classification. We provide manual and empirical evidence to demonstrate the existence of such a shift when removing connectives from explicit examples, analyze contributing factors, and propose two mitigation strategies, which we validate through experiments on various corpora.

In **Chapter 7**, we present an enhanced PDTB parser that incorporates our annotation-inspired implicit discourse relation classifier to extract discourse relations from texts. With this parser, we show that text coherence is correlated with the sequence of parsed discourse relations. Building on this insight, we propose a novel fusion model that leverages these relations for coherence modeling. Finally, we assess the effectiveness of our model across multiple corpora.

Part III: Coherence Modeling Enhanced with Entities and Discourse Relations
In **Chapter 8**, we introduce two methods, a Fusion Transformer and a Graph Prompt, that jointly model entities and discourse relations for coherence assessment. Our results show that models leveraging both types of features consistently outperform those that use only one or neither.

Conclusions
Finally, we conclude this thesis in **Chapter 9**, where we first summarize our findings and contributions, and then discuss potential directions for future research.

1.5 Published Work

This dissertation expands on the following publications:

- Wei Liu, Xiyan Fu, Michael Strube. Modeling Structural Similarities between Documents for Coherence Assessment with Graph Convolutional Networks. In: ACL 2023, pages 7792-7808. Code: `https://github.com/liuwei1206/StruSim`. C.f.: Chapter 4.

- Wei Liu, Michael Strube. Annotation-Inspired Implicit Discourse Relation Classification with Auxiliary Discourse Connective Generation. In: ACL 2023, pages 15696-15712. Code: `https://github.com/liuwei1206/ConnRel`. C.f.: Chapter 5.

- Wei Liu, Yi Fan, Michael Strube. HITS at DISRPT 2023: Discourse Segmentation, Connective Detection, and Relation Classification. In: DISRPT 2023, pages 43-49. Code: `https://github.com/liuwei1206/disrpt2023`. C.f.: Chapter 5.

- Wei Liu, Stephen Wan, Michael Strube. What Causes the Failure of Explicit to Implicit Discourse Relation Recognition? In: NAACL 2024, pages 2738–2753. Code: `https://github.com/liuwei1206/Exp2Imp`. C.f.: Chapter 6.

- Wei Liu, Michael Strube. Discourse Relation-Enhanced Neural Coherence Modeling. In: ACL 2025, pages 4748–4762.. Code: `https://github.com/liuwei1206/Relcoh`. C.f.: Chapter 7.

- Wei Liu, Michael Strube. Joint Modeling of Entities and Discourse Relations for Coherence Assessment. In: EMNLP 2025. Code: `https://github.com/liuwei1206/EntyRel`. C.f.: Chapter 8.

Additional publications from my PhD period that are not covered in this thesis:

- Wei Liu, Sony Trenous, Leonardo F. R. Ribeiro, Bill Byrne, Felix Hieber. XRAG: Cross-lingual Retrieval-Augmented Generation. In: EMNLP 2025 Findings. Code: `https://github.com/amazon-science/XRAG`.

Chapter 2

Background

This chapter introduces fundamental concepts essential for understanding the topics and methods developed later in this thesis. We begin with the definitions of coherence and discourse relations, then discuss commonly used corpora for coherence modeling and discourse relation classification. Finally, we present the deep learning techniques applied in this thesis.

2.1 Coherence

Coherence describes the relationship between sentences that makes a group of sentences logically connected rather than just a random collection of them (Jurafsky and Martin, 2025). A coherent text presents topics in a structured manner, enabling readers to recognize their relationships and perceive the text as a unified whole. For instance, Example (2.1) is highly coherent because it consistently focuses on Mike's efforts to secure a faculty position. Each sentence logically follows the previous one, detailing the steps he has taken to improve his profile. In contrast, Example (2.2) lacks coherence, as its sentences introduce unrelated information without a clear thematic connection.

(2.1) [Mike wants to find a position in academia, but his profile is not good enough.]$_{s1}$ [So, he has been doing a lot to improve it.]$_{s2}$ [For example, he has volunteered to teach more classes to gain teaching experience.]$_{s3}$ [He has also been working hard to get more publications.]$_{s4}$

(2.2) [Mike wants to find a job in academia, but his profile is not good enough.]$_{s1}$ [He eats a lot at lunch every day, so he's getting fat.]$_{s2}$ [Mike likes traveling and trying new things.]$_{s3}$ [He also has a dog named Bob.]$_{s4}$

In linguistics, coherence can be established through various means, namely entity-based, topic-based, and discourse relation-based approaches (Jurafsky and Martin, 2025).

2.1.1 Entity-based Coherence

Entity-based coherence is achieved when all sentences describe common entities and maintain continuity by consistently tracking and organizing references to them (such as people, objects, or concepts) throughout a text. An example is shown in Example (2.3), where all four sentences are related to AI: the first two introduce the topic of AI, and the last two describe its advantages and disadvantages.

(2.3) [**Artificial intelligence (AI)** has rapidly transformed various industries, from healthcare to finance.]$_{s1}$ [**This technology** enables machines to learn from data and make decisions without human intervention.]$_{s2}$ [As **AI systems** become more advanced, they can analyze vast amounts of information in real-time.]$_{s3}$ [However, **these intelligent models** also raise ethical concerns, such as bias in decision-making.]$_{s4}$

The most influential framework for explaining entity-based coherence is **Centering Theory**, introduced by Grosz et al. (1995). This theory identifies the most salient entity in each sentence (referred to as the "center" or "focus") and describes the coherence by tracking how these centers are referenced across sentences. Specifically, Centering Theory proposes that discourses in which consecutive sentences consistently focus on the same salient entity are more coherent than those that frequently switch between multiple entities. Below are two examples from Grosz et al. (1995), which convey the same information but exhibit different levels of coherence:

(2.4) [John went to his favorite music store to buy a piano.]$_{s1}$ [He had frequented the store for many years.]$_{s2}$ [He was excited that he could finally buy a piano.]$_{s3}$ [He arrived just as the store was closing for the day.]$_{s4}$

(2.5) [John went to his favorite music store to buy a piano.]$_{s1}$ [It was a store John had frequented for many years.]$_{s2}$ [He was excited that he could finally buy a piano.]$_{s3}$ [It was closing just as John arrived.]$_{s4}$

The first text is more coherent than the second because, as Grosz et al. (1995) pointed out, it maintains a clear focus on John, describing his actions and feelings. In contrast, the second text shifts between John and the store multiple times (first focuses on John, then on the store, next back to John, and finally returns to the store again), resulting in a lack of consistency.

To implement this idea, Centering Theory maintains two representations for each sentence $\mathrm{S_i}$: **the backward-looking center**, $\mathrm{C_b(S_i)}$, and **the forward-looking centers**, $\mathrm{C_f(S_i)}$. The backward-looking center of a sentence is the most salient entity regarding the previous context. The forward-looking centers are a list of entities within the sentence that might

become the focus of the next sentence. The elements of the forward-looking centers are ordered based on factors such as grammatical roles to reflect their relative prominence in the sentence. The highest-ranked element among the forward-looking centers is called **the preferred center**, $\mathrm{C_p(S_i)}$.

The theory also defines several types of transitions between pairs of sentences $\mathrm{S_i}$ and $\mathrm{S_{i+1}}$, including **Continue**, **Retain**, and **Shift**, based on the relationships among $\mathrm{C_b(S_{i+1})}$, $\mathrm{C_b(S_i)}$, and $\mathrm{C_p(S_{i+1})}$. Brennan et al. (1987) further subdivide the Shift transition into **Smooth Shift** and **Rough Shift**, a distinction that has since been widely adopted in the literature. Table 2.1 summarizes the definitions of these transitions.

In Centering Theory, transitions are ordered by preference, with Continue preferred over Retain, Retain over Smooth Shift, and Smooth Shift over Rough Shift (i.e., Continue > Retain > Smooth Shift > Rough Shift). We focus here on these canonical transition types, which assume the presence of a well-defined backward-looking center. Extensions of the framework have proposed additional transition types, such as **Establishment**, **Null**, and **Zero**, to account for discourse-initial sentences and cases where no plausible backward-looking center can be identified (see Poesio et al. (2004) for an overview). In Example (2.4), the backward-looking center, the forward-looking centers, and the preferred center of the first two sentences are:
$\mathrm{C_b(S_1)}$ = undefined
$\mathrm{C_f(S_1)}$ = {John, music store, piano}
$\mathrm{C_p(S_1)}$ = John
$\mathrm{C_b(S_2)}$ = John
$\mathrm{C_f(S_2)}$ = {John, music store}
$\mathrm{C_p(S_2)}$ = John
Since $\mathrm{C_b(S_1)}$ = NIL and $\mathrm{C_b(S_2)} = \mathrm{C_p(S_2)}$, the transition between the first two sentences is Continue. Similarly, for the first two sentences of Example (2.5), we have:
$\mathrm{C_b(S_1)}$ = undefined
$\mathrm{C_f(S_1)}$ = {John, music store, piano}
$\mathrm{C_p(S_1)}$ = John
$\mathrm{C_b(S_2)}$ = John
$\mathrm{C_f(S_2)}$ = {John, music store}
$\mathrm{C_p(S_2)}$ = music store (referent "it")
Here, $\mathrm{C_b(S_2)} \neq \mathrm{C_p(S_2)}$, so the transition is Retain. This explains why Example (2.4) is more coherent than Example (2.5) because Continue is preferred to Retain.

	$C_b(S_{i+1}) = C_b(S_i)$ or $C_b(S_i) = \text{NIL}$	$C_b(S_{i+1}) \neq C_b(S_i)$
$C_b(S_{i+1}) = C_p(S_{i+1})$	Continue	Smooth-Shift
$C_b(S_{i+1}) \neq C_p(S_{i+1})$	Retain	Rough-Shift

Table 2.1 Four types of transitions in Certering Theory, from Brennan et al. (1987).

2.1.2 Topic-based Coherence

In addition to describing common entities, coherence can also be established by discussing topically or semantically related words across sentences, which is called topic- or lexicon-based coherence (Jurafsky and Martin, 2025). An instance can be found in Example (2.6), where no common entities are shared between sentences, but the text is still coherent because the sentences are linked by sports-related vocabulary.

(2.6) [Different **excercise** have different benifits for our body.]$_{s1}$ [**Jogging** can increase your breathing and heart rate.]$_{s2}$ [**Table tennis** keeps you away from shortsightedness.]$_{s3}$ [**Playing basketball** can stengthen your muscles.]$_{s4}$ [**Yoga** helps to relieve your back pain.]$_{s5}$ [So, pick the **one** your body needs the most.]$_{s6}$

Topically coherent texts usually draw from a single semantic field or topic, which often leads to lexical cohesion (Halliday and Hasan, 1976), a surface-level property where related words link sentences together. There are two primary forms of lexical cohesion: reiteration and collocation. Reiteration can be accomplished by repeating lexical items or by using synonymy, antonymy, hyponymy, taxonomy, etc. Example (2.6) belongs to this category, where taxonomy (i.e., different types of exercise) is applied to create ties between sentences. By contrast, collocation is a form of lexical cohesion that depends on the tendency of some words to co-occur in texts. For example, when one sees the noun *bicycle* in a sentence, it is more probable that the verb *ride* will also appear.

2.1.3 Discourse Relation-based Coherence

Ultimately, a text can maintain coherence by systematically using logical or semantic relations to connect clauses or sentences. Below, we show two examples:

(2.7) [Hagen took a flight from Berlin to Seattle.]$_{s1}$ [He had to attend a conference about an AI product.]$_{s2}$ [The product will be used to facilitate remote education.]$_{s3}$

(2.8) [Hagen took a flight from Berlin to Seattle.]$_{s1}$ [He likes spinach.]$_{s2}$ [Spinach is a very common vegetable in China.]$_{s3}$

The first example is very coherent because there is a strong connection between the sentences. Specifically, the second sentence explains Hagen's actions in the first, and the third provides more detailed information about the product mentioned in the second. By contrast, the second example is less coherent because it's unclear to the reader why the second sentence follows the first: what does liking spinach have to do with flight trips? Similarly, it's hard for readers to understand why "Spinach in China" relates to the first two sentences about Hagen. We call the logical connection between sentences **discourse relations** (or **coherence relations**). Many discourse relation theories analyze how sentences and clauses are connected to create coherent text. Here, we introduce the two most common theories: **Rhetorical Structure Theory** (RST) and **the Penn Discourse Treebank** (PDTB) Framework.

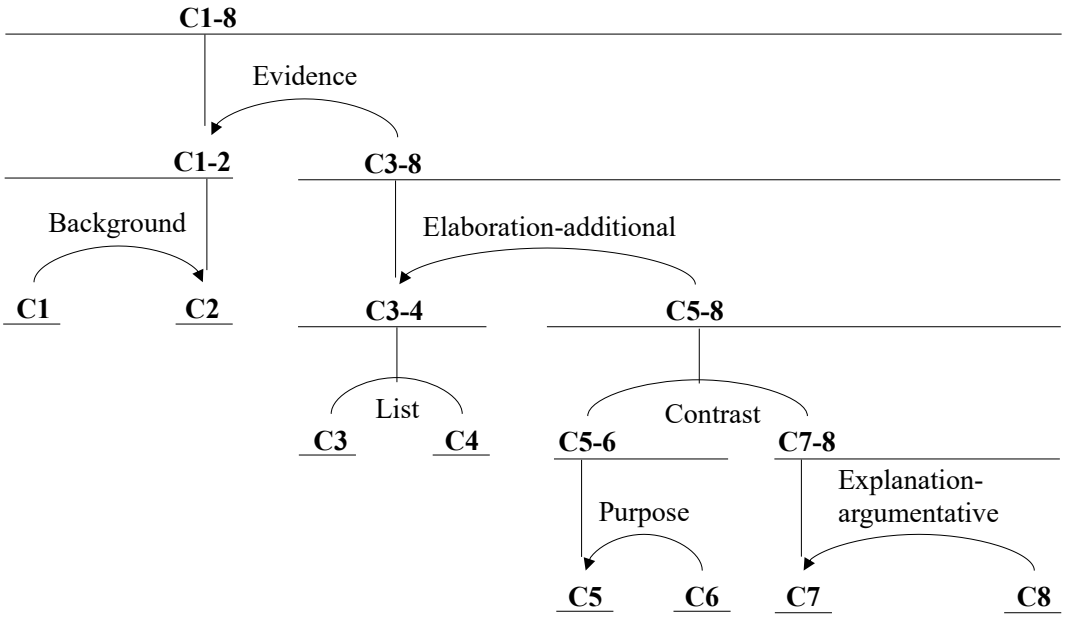
Rhetorical Structure Theory

Rhetorical Structure Theory (RST) was developed by Mann and Thompson (1988) for analyzing the coherence of written and spoken discourse. It describes how parts of a text relate to each other structurally and functionally to form a meaningful whole. Specifically, in RST, texts are structured in a tree-like structure, with nodes representing specific text spans connected by discourse relations. Below is an example from Marcu (2000a). Figure 2.1 shows the rhetorical structure representation of the text in Example (2.9).

(2.9) With its distant orbit–50 percent farther from the sun than Earth–and slim atmospheric blanket, Mars experiences frigid weather conditions. Surface temperatures typically average about -60 degrees Celsius (-76 degrees Fahrenheit) at the equator and can dip to -123 degrees C near the poles. Only the midday sun at tropical latitudes is warm enough to thaw ice on occasion, but any liquid water formed in this way would evaporate almost instantly because of the low atmospheric pressure.

In the RST tree, the smallest unit corresponds to a sentence, clause, or phrase, referred to as an **Elementary Discourse Unit** (EDU). These units can be connected to form larger text spans, which can, in turn, be linked to even larger spans until they encompass the entire text (see Figure 2.1). The edges in the tree represent discourse relations, typically connecting a nucleus and a satellite, though they can also link two nuclei. The nucleus is the central unit of a discourse relation, carrying the core meaning that remains coherent even on its own. In contrast, the satellite provides supporting information that enhances, explains, or modifies the nucleus but is not essential for understanding the main message.

The discourse relations in RST can be broadly categorized into two types: nucleus-satellite relations and multi-nuclear relations. In the nucleus-satellite relations, one unit (the nucleus) is more central, while the other (the satellite) provides supporting information. The satellite depends on the nucleus, whereas the nucleus can stand alone. Taking the text



C1: With its distant orbit–50 percent farther from the sun than Earth–and slim atmospheric blanket,

C2: Mars experiences frigid weather conditions.

C3: Surface temperatures typically average about -60 degrees Celsius (-76 degrees Fahrenheit) at the equator

C4: and can dip to -123 degrees C near the poles.

C5: Only the midday sun at tropical latitudes is warm enough

C6: to thaw ice on occasion,

C7: but any liquid water formed in this way would evaporate almost instantly

C8: because of the low atmospheric pressure.

Fig. 2.1 A rhetorical structure representation of the text in Example (2.9).

spans **C1** and **C2** in Figure 2.1 as an example, **C2** is a nucleus unit, expressing the core idea that it is very cold on Mars, while **C1** is a satellite unit, providing background information explaining why this happens. Below are a few examples of nucleus-satellite relations:

- Elaboration: The satellite expands on the nucleus by adding details.

- ▶ [City, in Sweden, will be the site of the 1969 International Conference on Computational Linguistics, September 1-4.]$_{\text{nucleus}}$ [It is expected that some 250 linguists will attend from Asia, Western Europe, Eastern Europe, including Russia, and the United States.]$_{\text{satellite}}$

- Cause: The satellite explains the reason for the nucleus.

  - ▶ [Hagen failed to pass the examination]$_{\text{nucleus}}$ [because he didn't spend much time on study.]$_{\text{satellite}}$

- Evidence: The satellite supports the nucleus with evidence.

  - ▶ [Sun Yusha is a highly skilled table tennis player.]$_{\text{nucleus}}$ [She has won several international championships, including two last year.]$_{\text{satellite}}$

In multi-nuclear relations, both units have equal importance, and removing one unit would significantly alter the overall meaning. The Contrast relation between the text spans **C5-6** and **C7-8** in Figure 2.1 falls into this category, in which removing any part would result in incomplete information for the reader. Below are a few examples of multi-nuclear relations:

- Contrast: Two units present opposing ideas.

  - ▶ [She likes sunny days,]$_{\text{nucleus}}$ [but he likes rainy days.]$_{\text{nucleus}}$

- List: Multiple units contribute to a common topic

  - ▶ [I am 17 years old.]$_{\text{nucleus}}$ [It is summer, and football practice is about to begin.]$_{\text{nucleus}}$

Penn Discourse TreeBank

In the early stages of computational discourse research, the study of discourse relations is closely linked to discourse structure. As a result, theories such as RST implicitly assume a tree structure. However, many studies (Mann and Thompson, 1988; Knott et al., 2000) have identified this as a drawback, as annotating discourse relations requires an understanding of the overall coherence of a given text, and annotators often disagree on this. This has motivated efforts to annotate discourse relations independently of discourse structure. Such a shallow model of discourse coherence can be annotated based solely on local context. The Penn Discourse TreeBank (PDTB, Miltsakaki et al., 2004) is the most prominent framework in this category.

The Penn Discourse TreeBank is a corpus that uses a lexically grounded framework to annotate discourse relations. It adopts a shallow discourse annotation approach, focusing on

discourse connectives (such as *because*, *however*, and *as a result*) and the two text spans they connect, known as **arguments**. Specifically, rather than having annotators identify discourse relations between text spans directly, they are provided with a list of discourse connectives. Annotators then recognize the discourse connectives in the text along with the two arguments linked by each connective and finally mark a discourse relation for each connective. An instance is shown in Example (2.10), where the connective *because* signals a *Cause* relation between two arguments. Discourse relations signaled by connectives present in the text are referred to as **Explicit** discourse relations.

(2.10) [They may feel emotionally secure now]$_{\text{Arg1}}$ **because** [they are not heavily in the stock market]$_{\text{Arg2}}$

(2.11) [He has not changed, but those around him have.]$_{\text{Arg1}}$ (**implicit=Because**) [Many of his views on the protection of wilderness areas are now embraced by the mainstream.]$_{\text{Arg2}}$

(2.12) [After training at an average discount of more than 20% in late 1987 and part of last year, country funds currently trade at an average premium of 6%.]$_{\text{Arg1}}$ (**AltLex**) [Share prices of many of these funds this year have climbed much more sharply than the foreign stocks they hold.]$_{\text{Arg2}}$

(2.13) [Pierre Vinken, 61 years old, will join the board as a non-executive director on Nov. 29.]$_{\text{Arg1}}$ (**EntRel**) [Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.]$_{\text{Arg2}}$

(2.14) [Mr. Rapanelli met in August with U.S. Assistant Treasury Secretary David Mulford.]$_{\text{Arg1}}$ (**NoRel**) [Argentine negotiator Carlos Carballo was in Washington and New York this week to meet with banks.]$_{\text{Arg2}}$

However, not all text spans are connected by a connective. Therefore, the PDTB also annotates adjacent sentence pairs with no explicit signal. In some cases, a discourse connective can be inserted between paragraph-internal adjacent sentence pairs despite not being related by any explicit connectives, as shown in Example (2.11). In this example, we can insert an implicit connective *because* between the two arguments without causing redundancy. Discourse relations inferred from such implicit connectives are called **Implicit** discourse relations. In other instances, when annotators determine that no implicit connective is suitable between adjacent sentence pairs, they are further categorized as follows: (a) **AltLex**, where a discourse relation is inferred, but inserting an implicit connective would be *redundant*, as the relation is already expressed through an alternative lexicalization; (b) **EntRel**, where no clear discourse relation is inferred, and the second sentence elaborates on or continues discussing

Level-1	Level-2	Level-3
TEMPORAL	Synchronous	-
	Asynchronous	precedence
		succession
CONTINGENCY	Cause	reason
		result
		negresult
	Cause+Belief	reason+belief
		result+belief
	Cause+SpeechAct	reason+speechact
		result+speechact
	Condition	arg1-as-cond
		arg2-as-cond
	Condition+SpeechAct	-
	NegActive-Condition	arg1-as-negcond arg2-as-negcond
	NegActive-Condition+SpeechAct	-
	Purpose	arg1-as-goal
		arg2-as-goal
COMPARISON	Concession	arg1-as-denier
		arg2-as-denier
	Concession+SpeechAct	arg2-as-denier+SpeechAct
	Contrast	-
	Similarity	-
EXPANSION	Conjunction	-
	Disjunction	-
	Equivalence	-
	Exception	arg1-as-excpt
		arg2-as-excpt
	Instantiation	arg1-as-instance
		arg2-as-instance
	Level-of-Detail	arg1-as-detail
		arg2-as-detail
	Manner	arg1-as-manner
		arg2-as-manner
	Substitution	arg1-as-subst
		arg2-as-subst

Table 2.2 PDTB 3.0 Sense Hierarchy. The leftmost column contains the Level-1 senses, and the middle column, the Level-2 senses. For asymmetric relations, Level-3 senses are located in the rightmost column.

an entity mentioned in the first, aligning with entity-based coherence (Knott et al., 2000); and (c) **NoRel**, where neither a discourse relation nor entity-based coherence is present between the sentences. Examples (2.12), (2.13), and (2.14) illustrate these three cases, respectively.

In the PDTB framework, discourse relations are organized hierarchically into three levels: class, type, and subtype (cf. Table 2.2). At the top level, the framework defines four broad semantic classes: TEMPORAL, CONTINGENCY, COMPARISON, and EXPANSION. Each class is subdivided into types that further specify its meaning. For instance, TEMPORAL includes two types: Synchronous and Asynchronous. At the third level, subtypes clarify the semantic role of each argument. Within TEMPORAL, the Asynchronous type is further divided into two subtypes: precedence and succession.

Other Theories of Discourse Coherence

In addition to RST and PDTB, several other well-known theories of discourse coherence have been proposed, such as Segmented Discourse Representation Theory and the Cognitive approach to Coherence Relations.

Segmented Discourse Representation Theory (**SDRT**; Asher and Lascarides, 2003) is a discourse interpretation theory that combines formal semantic representation with discourse structure and pragmatic reasoning. Building on dynamic semantics, SDRT models how the interpretation of a sentence depends on the evolving discourse context, while also incorporating insights from AI-based approaches that emphasize discourse structure and commonsense inference.

SDRT extends prior work on discourse structure by assigning rhetorical relations a precise dynamic semantic interpretation, which explains how the content of a discourse augments the compositional semantics of its clauses. In addition, SDRT incorporates commonsense reasoning with linguistic and non-linguistic information to determine rhetorical relations and resolve discourse-level phenomena such as pronoun interpretation, presupposition resolution, and bridging inferences. In this way, SDRT provides a unified account of how discourse structure and semantic interpretation jointly contribute to discourse coherence.

Another well-known framework for discourse coherence is the **Cognitive approach to Coherence Relations** (**CCR**; Sanders et al., 1992). Unlike formalisms that primarily focus on structural representations or formal semantics, CCR adopts a cognitive perspective, viewing coherence relations as mental constructs that reflect how language users establish meaningful connections between discourse segments during comprehension and production. It decomposes discourse relations into a small set of cognitively motivated dimensions: (I) Basic Operation: Causal vs. Additive. This dimension captures whether the coherence relation involves a causal dependency between discourse segments or merely adds informa-

tion without implying causation. (II) Source of Coherence: Semantic vs. Pragmatic. This dimension distinguishes whether the coherence relation is grounded in objective, real-world states of affairs (semantic) or arises from the speaker's reasoning, evaluation, or communicative intention (pragmatic). (III) Polarity: Positive vs. Negative. Positive relations reinforce expectations or alignments, while negative relations signal denial, contrast, or violation of an expected causal or logical connection. (IV) Basic Order: Basic vs. Non-basic. In basic order, the segments follow the canonical sequence (e.g., cause preceding effect), whereas non-basic order involves a reversal of this sequence, such as presenting an effect before its cause.

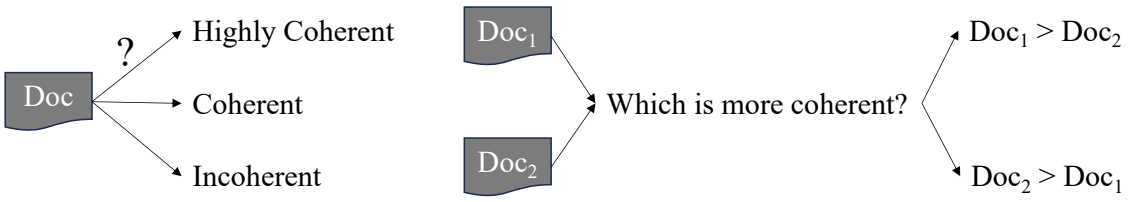
The strength of CCR lies in its parsimony. By using these four dimensions, CCR can generate a taxonomy of relations that aims to capture psychologically plausible distinctions in discourse processing. These abstract dimensions have also made CCR a useful "interlingua" for mapping and comparing different annotation schemes, such as aligning the disparate relation labels used in RST and PDTB (Sanders et al., 2021).

2.2 Tasks and Corpora

This thesis focuses on neural coherence modeling and includes discourse relation classification; therefore, we introduce both tasks and the corpora used for evaluation.

2.2.1 Coherence Modeling

Coherence modeling is the task of assessing how coherent a given text is (Lapata and Barzilay, 2005). It can be formulated as a classification task when applied to a single document (cf. Figure 2.2a), or as a ranking problem when applied to a pair of documents (cf. Figure 2.2b).



(a) Classification Task (b) Ranking Task

Fig. 2.2 Two forms of the task of Coherence Modeling.

Manually annotating text coherence is costly and time-consuming, as it requires linguistic expertise. As a result, prior work mainly evaluates coherence models on synthetic tasks

Original
S1: The Justice Department is conducting an anti-trust trial against Microsoft Corp. with evidence that the company is increasingly attempting to crush competitors.
S2: Microsoft is accused of trying to forcefully buy into markets where its own products are not competitive enough to unseat established brands.
S3: The case revolves around evidence of Microsoft aggressively pressuring Netscape into merging browser software.
S4: Microsoft claims its tactics are commonplace and good economically.
S5: The government may file a civil suit ruling that conspiracy to curb competition through collusion is a violation of the Sherman Act.
S6: Microsoft continues to show increased earnings despite the trial.
Shuffled
S5: The government may file a civil suit ruling that conspiracy to curb competition through collusion is a violation of the Sherman Act.
S1: The Justice Department is conducting an anti-trust trial against Microsoft Corp. with evidence that the company is increasingly attempting to crush competitors.
S4: Microsoft claims its tactics are commonplace and good economically.
S3: The case revolves around evidence of Microsoft aggressively pressuring Netscape into merging browser software.
S2: Microsoft is accused of trying to forcefully buy into markets where its own products are not competitive enough to unseat established brands.
S6: Microsoft continues to show increased earnings despite the trial.

Table 2.3 An example of shuffle test from Barzilay and Lapata (2008), where the first is the original document and the second is the shuffled one.

(e.g., the shuffle test) or proxy tasks (automatic essay scoring). More recently, recognizing the importance of coherence modeling for NLP applications, researchers have developed higher-quality datasets specifically designed to assess discourse coherence.

The Shuffle Test

In early work, the **shuffling test**, introduced by Barzilay and Lapata (2005, 2008), was the most widely used task for evaluating coherence models. The task is a ranking problem (but in the form of binary classification): Given a pair of documents, one original and one generated by randomly shuffling the sentences of the original, the model must predict which document is more coherent. The underlying assumption is that the original document is inherently more

	Sections	# Doc.	# Pairs	Avg. # Sent.
Train	00-13	1378	26422	21.5
Test	14-24	1053	20411	22.3

Table 2.4 Statistics of the shuffle test dataset created from the Wall Stree Journal portion of Penn TreeBank.

coherent than its shuffled counterpart. Ideally, a robust coherence model should consistently rank the original document higher. Table 2.3 provides an example from Barzilay and Lapata (2008), showing both the original and its shuffled version.

The shuffle test dataset has multiple versions. The original version comprises documents from two distinct genres: newspaper articles about earthquakes (Earthquake) and government-written accident reports (Accidents). A more widely used version, introduced by Elsner and Charniak (2011), replaces these with articles from the Penn Treebank (specifically, the Wall Street Journal). In this version, articles from Sections 00 to 13 are used for training, while articles from Sections 14 to 24 are reserved for evaluation. Table 2.4 summarizes the number of <original, permuted> pairs included in the training and evaluation sets.

However, many studies have raised concerns about the suitability of this artificial task for coherence modeling. For instance, Lai and Tetreault (2018) create a high-quality corpus for coherence assessment using expert annotators (see Section 2.3), and finds that models trained on <original permuted> pairs, despite achieving near-perfect accuracy on the shuffle test, perform poorly on their dataset, even falling below a random baseline. Similarly, Mohiuddin et al. (2021) examine whether models trained on the shuffle test learn features that generalize to real-world tasks. They train models on the shuffle test training set and evaluate them on downstream tasks such as machine translation and text summarization. The underlying hypothesis is that models capturing genuine coherence should prefer outputs that align with human judgments. However, their results show that models performing well on the artificial task often perform poorly on these downstream tasks. These findings have prompted recent work to explore alternative evaluation strategies for coherence modeling, including automatic essay scoring and assessing discourse coherence.

Automatic Essay Scoring

Automated Essay Scoring (AES) is the task of automatically assigning a holistic score to an essay, summarizing its overall quality (Ke and Ng, 2019). Since coherence is a key factor in determining the quality of an essay, early work has investigated the relationship between this task and coherence modeling. For example, Miltsakaki and Kukich (2000) annotate

ID	Content
1	Agree or Disagree: It is better to have broad knowledge of many academic subjects than to specialize in one specific subject.
2	Agree or Disagree: Young people enjoy life more than older people do.
3	Agree or Disagree: Young people nowadays do not give enough time to helping their communities.
4	Agree or Disagree: Most advertisements make products seem much better than they really are.
5	Agree or Disagree: In twenty years, there will be fewer cars in use than there are today.
6	Agree or Disagree: The best way to travel is in a group led by a tour guide.
7	Agree or Disagree: It is more important for students to understand ideas and concepts than it is for them to learn facts.
8	Agree or Disagree: Successful people try new things and take risks rather than only doing what they already know how to do well.

Table 2.5 Topic prompts in the TOEFL dataset.

Prompt ID	# Doc.	Avg # Word.	Max # Word.	Avg # Sent.
1	1656	339.1	806	13.7
2	1562	357.8	770	15.7
3	1396	343.5	731	14.7
4	1509	338.0	699	15.1
5	1648	358.4	876	15.2
6	960	358.3	784	15.3
7	1686	336.6	638	14.0
8	1683	340.9	659	14.7

Table 2.6 The statistics of the TOEFL dataset.

centering transitions in student essays and examines how these transitions relate to various levels of writing quality. Their findings reveal a strong correlation between essay scores and coherence scores derived from centering transitions. Building on these insights, automatic essay scoring has become a widely used method for evaluating coherence models (Farag et al., 2018; Jeon and Strube, 2020b; Liu et al., 2023a).

	Split	# Doc.	Avg # Word.	Max # Word.	Avg # Sent.
Yahoo	Train	1000	157.2	339	7.8
	Test	200	162.7	314	7.8
Clinton	Train	1000	182.9	346	8.9
	Test	200	186.0	352	8.8
Enron	Train	1000	185.1	353	9.2
	Test	200	191.1	348	9.3
Yelp	Train	1000	178.2	347	10.4
	Test	200	179.1	340	10.1

Table 2.7 The statistics of the GCDC dataset.

Coherence	GCDC				TOEFL							
	Yahoo	Clinton	Enron	Yelp	P1	P2	P3	P4	P5	P6	P7	P8
low	45.56	28.22	29.89	27.00	09.58	08.33	13.16	12.82	09.92	10.63	09.80	11.20
medium	17.54	20.67	19.33	21.89	54.23	54.92	50.24	53.26	53.24	54.53	54.95	55.78
high	37.00	51.11	50.78	51.11	36.19	36.75	36.66	33.92	36.84	34.84	35.25	33.00

Table 2.8 Label distribution in TOEFL and GCDC (%).

The **TOEFL** dataset (Blanchard et al., 2014) is a widely used resource for coherence modeling in automatic essay scoring. It consists of essays written by students from various countries, covering **eight prompts** (see Table 2.5). Each essay is labeled with a readability level: **low**, **medium**, or **high**. Previous studies (Jeon and Strube, 2020b, 2021) have shown that the TOEFL dataset generally contains higher-quality essays compared to those in other corpora, such as ASAP (Hamner et al., 2012). Therefore, this dataset will be used to evaluate our models in this thesis. Table 2.6 presents the statistics of the TOEFL dataset and Table 2.8 shows the label distribution of the corpus.

Assessing Discourse Coherence

Assessing Discourse Coherence (ADC) is the task of measuring the coherence of a text. It aligns closely with the goal of coherence modeling and is therefore widely used in recent studies (Farag and Yannakoudakis, 2019; Sheng et al., 2024). The benchmark dataset for this task is the **Grammarly Corpus of Discourse Coherence** (**GCDC**) dataset, introduced by Lai and Tetreault (2018). This dataset is valuable because it includes coherence annotations for texts, a process that, as previously mentioned, is both costly and time-consuming.

The GCDC dataset includes texts from **four domains**: (1) online forum posts from the Yahoo Answers L6 corpus (**Yahoo**); (2) emails from the U.S. State Department's release of Hillary Clinton's office emails (**Clinton**); (3) emails from the Enron Corpus (**Enron**); and (4) business reviews from the Yelp Open Dataset (**Yelp**). Expert raters with prior experience in linguistic annotation assigned each text a coherence score from 1 to 3, corresponding to **low**, **medium**, and **high** coherence, respectively. Tables 2.7 and 2.8 provide the statistics and label distribution for this corpus, respectively.

2.2.2 Discourse Relation Classification

Discourse relations, such as *Cause* and *Contrast*, describe the logical relationship between two text spans (called arguments in the PDTB). Discourse connectives, such as *because* and *as a result*, are words or phrases that signal the presence of a discourse relation (Pitler et al., 2009). In the two examples below, Example (2.15) contains a *Cause* relation between its two arguments, while Example (2.16) demonstrates an *Instantiation* relation. Here, the word *because* and the phrase *for example* serve as discourse connectives.

(2.15) [People love Heidelberg]$_{\text{Arg1}}$ [**because** the city is beautiful]$_{\text{Arg1}}$

—- Contingency.Cause

(2.16) [Heidelberg is attractive for many reasons.]$_{\text{Arg1}}$ [**For example**, the city has rivers and mountains.]$_{\text{Arg1}}$

—- Expansion.Instantiation

(2.17) [Tom was selected as the president to handle the issue.]$_{\text{Arg1}}$ [Many people doubt if he has the ability.]$_{\text{Arg1}}$

—- Comparison.Contrast

Discourse Relation Classification is the task of identifying the type of discourse relation between two given text spans (arguments). It is typically divided into two forms, depending on whether a connective is present in the input text:

- **Explicit Discourse Relation Classification** is performed when a connective (e.g., *because* in Example 2.15) is present.
- **Implicit Discourse Relation Classification** is required when no connective is explicitly stated (as in Example 2.17), and the relation must be inferred from context.

Explicit discourse relation classification is relatively straightforward, as discourse connectives serve as clear indicators of the relationship between arguments. For example, in most

	PDTB 2.0	PDTB 3.0
L1	Comparison	Comparison
	Contingency	Contingency
	Expansion	Expansion
	Temporal	Temporal
L2	Comparison.Concession	Comparison.Concession
	Comparison.Contrast	Comparison.Contrast
	Contingency.Cause	Contingency.Cause
	Contingency.Pragmatic cause	Contingency.Cause+Belief
	Expansion.Conjunction	Contingency.Condition
	Expansion.Instantiation	Contingency.Purpose
	Expansion.Alternative	Expansion.Conjunction
	Expansion.List	Expansion.Equivalence
	Expansion.Restatement	Expansion.Instantiation
	Temporal.Asynchronous	Expansion.Level-of-detail
	Temporal.Synchrony	Expansion.Manner
		Expansion.Substitution
		Temporal.Asynchronous
		Temporal.Synchronous

Table 2.9 The top-level (L1) and second-level (L2) discourse relations in PDTB 2.0 and PDTB 3.0 commonly used in the literature.

cases, the connective *because* signals a *Cause* relation. In contrast, implicit discourse relation classification is more challenging, as it requires inferring the relation solely from the arguments without explicit connective cues.

Discourse relation classification is a key component of various discourse theories, such as Rhetorical Structure Theory (RST) and the Penn Discourse TreeBank (PDTB) framework. Consequently, many corpora have been developed for training and evaluating discourse relation classifiers. Here, we introduce two such corpora used in this thesis: the Penn Discourse TreeBank and the Georgetown University Multilayer Corpus.

Penn Discourse TreeBank

The Penn Discourse TreeBank (PDTB) is a large-scale corpus for discourse analysis that provides annotated discourse relations between two adjacent text spans. It provides a clear distinction between explicit examples (where connectives are present in the text) and implicit examples (where connectives are absent), and annotates suitable connectives for the implicit

Dataset	Type	Train	Dev	Test
PDTB 2.0	Explicit	14117	1462	1285
	Implicit	12632	1183	1046
PDTB 3.0	Explicit	18626	1944	1767
	Implicit	17085	1653	1474

Table 2.10 The statistics of PDTB 2.0 and PDTB 3.0.

Type	Train	Dev	Test
Explicit	5185	813	796
Implicit	3136	519	541

Table 2.11 The statistics of the GUM corpus.

cases. The corpus has two versions, **PDTB 2.0** (Prasad et al., 2008) and **PDTB 3.0** (Webber et al., 2019b), with PDTB 3.0 being an improved and extended version of PDTB 2.0. Each version includes annotations using a hierarchical discourse relation schema.

Most existing studies consider all four relation types for **top-level** (**L1**) relation prediction. In contrast, **second-level** (**L2**) relation recognition is often restricted to a smaller set of relation types, mainly because some relations (e.g., *NegActive–Condition*) have very few training instances. Consequently, prior work typically evaluates 11 L2 relation types in PDTB 2.0 and 14 in PDTB 3.0, as shown in Table 2.9.

Two data splitting strategies are widely used for PDTB 2.0 and PDTB 3.0. The first, proposed by Ji and Eisenstein (2015), uses Sections 2–20 for training, Sections 0–1 for development, and Sections 21–22 for testing, as presented in Table 2.10. The second strategy, referred to as section-level cross-validation (Kim et al., 2020), partitions the 25 sections into 12 folds, where each fold consists of 21 training sections, 2 validation sections, and 2 test sections.

The Georgetown University Multilayer Corpus

The Georgetown University Multilayer (GUM) corpus is an open-source, richly annotated corpus of English text created by the Computational Linguistics and Information Processing (CLIP) group at Georgetown University (Zeldes, 2017a). It is designed to facilitate research in a variety of linguistic domains by offering multiple layers of annotation, including tokenization and lemmatization, part-of-speech (POS) tagging, dependency syntax, discourse

parsing, and more. For discourse annotation, the GUM corpus adopts the framework of Rhetorical Structure Theory (RST).

The original GUM corpus is annotated with a constituent tree structure that contains both structural and relational information. In its latest version, [1] the authors of this corpus extend it to include annotations in the style of the Penn Discourse Treebank (PDTB). Specifically, they convert constituent trees into dependency trees, and map the resulting triples (EDU_i, Relation, EDU_j) to PDTB-style tuples (Arg1, Relation, Arg2). The discourse units (EDUs) are referred to as *text units*, and discourse connectives within these units are annotated accordingly. Each instance is categorized as explicit or implicit depending on whether a connective is present in the text unit. Due to the imbalance in label distribution across the full set of discourse relations, we focus on a subset comprising the following relation types: Causal, Concession, Conjunction, Contrast, Elaboration, Purpose, and Temporal. The statistics for this filtered subset of the corpus are presented in Table 2.11.

2.3 Deep Learning in NLP

Over the past few decades, research in NLP has developed rapidly. Different techniques have dominated the field over time, ranging from rule-based methods in the 1980s to statistical models in the 1990s and, more recently, neural networks. Neural models have been applied to almost every NLP task due to their strength in representation learning and feature combination. In this thesis, we primarily introduce several key techniques, including Transformers, Graph Neural Networks, Pre-trained Language Models, Large Language Models, and Model Adaptation.

2.3.1 Transformer

The Transformer is a deep learning architecture introduced by Google researchers in Vaswani et al. (2017). It first splits a text into tokens and converts them into vectors. Then, multiple layers of multi-head attention are applied to learn contextualized representations of the text. The attention mechanism allows key tokens to be emphasized while reducing the influence of less relevant ones. Compared to other widely used neural models in NLP, such as Recurrent Neural Networks (RNN, Elman, 1990) and Long Short-Term Memory Networks (LSTM, Hochreiter and Schmidhuber, 1997), it has several distinct advantages:

- **Parallel Processing**. Unlike RNNs or LSTMs, which process data sequentially, Transformers process entire sequences at once, allowing for faster training and inference.

[1] https://github.com/disrpt/latest

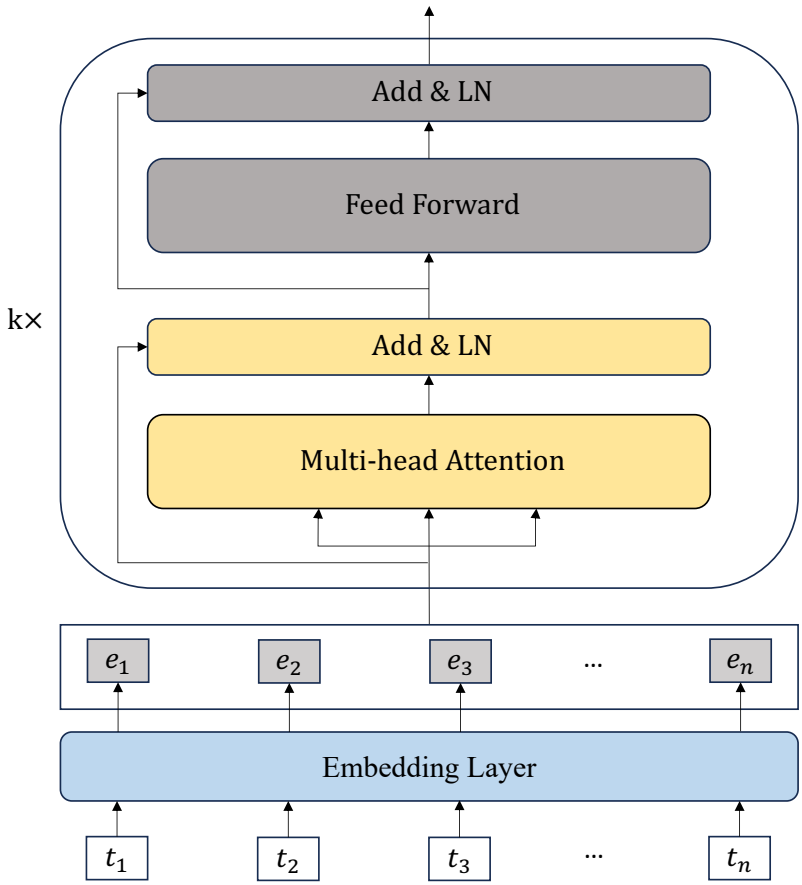


Fig. 2.3 An overview of the Transformer.

- **Better Handling of Long-Range Dependencies**. The attention mechanism in the Transformer allows it to consider relationships between all words in a sequence, making it better at capturing long-term dependencies.
- **Scalability**. Transformers scale well with large datasets and computational power (e.g., GPUs and TPUs).

Because of these advantages, Transformers have revolutionized AI and dominated modern AI research and applications. Figure 2.3 shows an overview of the Transformer. It contains two key components: an Embedding Layer and multiple Transformer Layers.

Embedding Layer. The embedding layer in a Transformer is the first step in processing input sequences. It converts discrete tokens into dense numerical representations that the model can understand. This step is crucial for capturing semantic relationships between words before going through attention mechanisms.

The embedding layer typically consists of two key components:

- **Token Embeddings**. It contains a learnable matrix E of size $V \times d_{model}$ that maps each token w_t to a high-dimensional vector:

$$x_t = \mathrm{E}[w_t] \tag{2.1}$$

  where V is the vocabulary size and d_{model} is the embedding dimension.

- **Positional Encoding**. Since the Transformer does not have a recurrence (like RNNs) or a convolution (like CNNs), it requires a way to incorporate information about the order of words in a sequence. This is achieved through Positional Encoding (PE), which is added to the token embeddings:

$$h_t = x_t + p_t \tag{2.2}$$

  where x_t is the embedding of the t-th token, p_t is the positional encoding for position t, and h_t is the final input representation of the t-th token, passed into the Transformer layers.

Transformer Layer. The transformation layer is responsible for converting the input representation sequence into a richer and more meaningful representation by capturing the relationships between tokens, regardless of their positions in the sequence. It consists of Multi-Head Self-Attention, Add & LayerNorm, and Feedforward Network (FFN).

- **Multi-Head Self-Attention**. Multi-head self-attention is a key mechanism in the Transformer model, which is inspired by self-attention. Self-attention allows a model to weigh the importance of different tokens within a sequence when encoding information. Unlike traditional sequence models, such as LSTM, that process inputs sequentially, self-attention considers all positions in the input simultaneously. Specifically, given an input X with length L, we first project it into queries Q, keys K, and values V:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{2.3}$$

  where $Q, K, V \in \mathbb{R}^{L \times D}$, and D represents the dimension of hidden states. Then, we compute the dot product of the queries and keys, divide by a scaling factor $\sqrt{d_k}$ (d_k is the dimension of query vectors), convert the scores into attention weights using softmax, and calculate the weighted sum of value vectors:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.4}$$

The result is a new sequence of vectors, each representing a position in the input attended by all other positions.

The multi-head variant of self-attention extends this concept by employing multiple attention heads in parallel. Each head operated independently, projecting the input into distinct subspaces and learning different attention distributions. This allows the model to jointly attend to information from different representation subspaces at various positions, thus enhancing its expressive power. The outputs of these heads are then concatenated and linearly transformed to produce the final output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O \tag{2.5}$$

and each head is:

$$\text{head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^K) \tag{2.6}$$

where h is the number of heads and W^O is the output projection matrix.

- **Feedforward Neural Network**. In addition to the self-attention mechanism, a Transformer Layer includes a feedforward network (FFN). This component is crucial for introducing non-linearity and learning complex transformations independently at each position in the input sequence. Specifically, the FFN comprises two linear (fully connected) layers that transform the input data. The first layer expands the input dimension d_{model} to a larger dimension $4d_{model}$, and the second layer projects it back to d_{model}. A Rectified Linear Unit (ReLU) activation function $\text{ReLU}(x) = \max(0, x)$is applied between these two linear layers to introduce non-linearity into the model, helping it to learn more complex patterns:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.7}$$

where W_1, b_1, W_2, b_2 are learnable parameters of the two linear layers.

- **Add & LayerNorm**. In the Transformer, each sub-layer, such as multi-head self-attention and feedforward networks, is followed by a residual connection and a layer normalization operation, often referred to as "Add & LayerNorm":

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer(x)}) \tag{2.8}$$

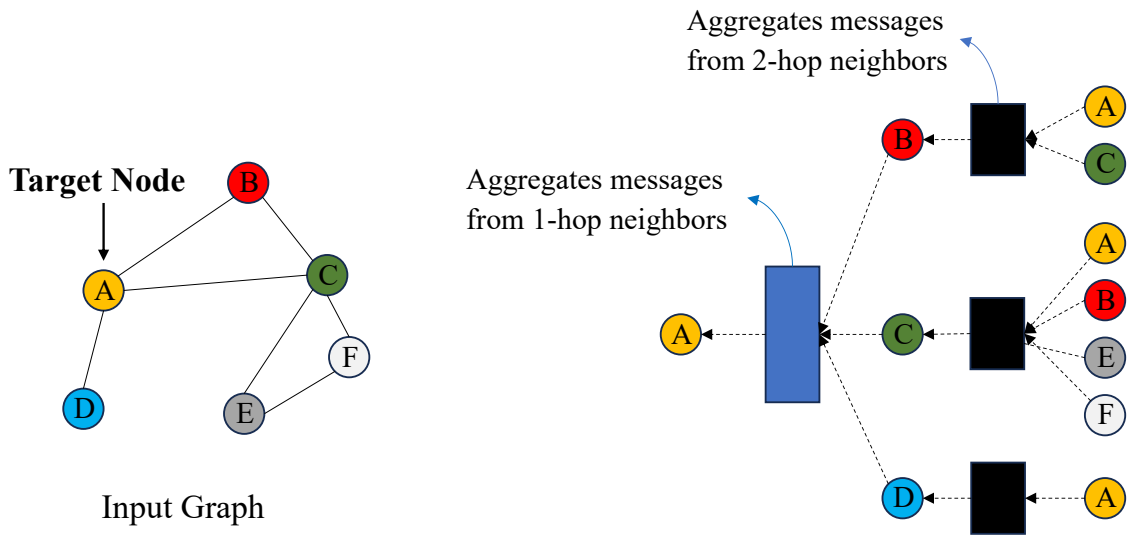


Fig. 2.4 An overview of Message Passing.

where x is the input to a $\mathrm{Sublayer}$ and the $\mathrm{Sublayer}$ is a Multi-Head Self-Attention or a Feedforward Network. This step plays a crucial role in stabilizing and improving the training of deep Transformer networks.

In summary, given the input $H^{l-1} = \{h_1^{l-1}, h_2^{l-1}, ..., h_n^{l-1}\}$ with n tokens, the L-th Transformer Layer compute the Multi-head Self-attention, Add & Layernorm, Feedforward Network, and another Add & LayerNorm, and output $H^l = \{h_1^l, h_2^l, ..., h_n^l\}$:

$$\begin{aligned} G &= \mathrm{LN}(H^{l-1} + \mathrm{MHAttn}(H^{l-1})) \\ H^L &= \mathrm{LN}(G + \mathrm{FFN}(G)) \end{aligned} \tag{2.9}$$

Figure 2.3 shows an overview of a Transformer Layer.

2.3.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a class of neural architectures designed to perform inference on data represented as graphs (Wu et al., 2021a). Unlike traditional neural networks that operate on grid-like structures such as sequences or images, GNNs are specifically designed to capture the complex relationships and interdependencies between nodes in arbitrary graph topologies.

In a typical GNN, each node iteratively updates its representation by aggregating and transforming information from its neighbors (see Figure 2.4). This process, often referred to as message passing, enables the model to learn node embeddings that encode both local

structure and feature information. The general formulation can be described as:

$$h_v^k = \mathrm{UPDATE}^k\left(h_v^{k-1}, \mathrm{AGGREGATE}^k\left(\{h_u^{k-1} : u \in \mathcal{N}(v)\}\right)\right) \tag{2.10}$$

where h_v^k denotes the representation of node v at k-th GNN layer and $\mathcal{N}(v)$ means the set of neighbors of node v. $\mathrm{AGGREGATE}$ represents a permutation-invariant function, such as mean and sum, and UPDATE is a learnable function, such as MLP.

Graph Convolutional Networks (GCNs)

Graph Convolutional Networks (GCNs) are among the most widely used variants of Graph Neural Networks (GNNs) (Kipf and Welling, 2017). They generalize the concept of convolution from grid-structured data (such as images) to graph-structured data. In a GCN, each node updates its representation by aggregating normalized features from its neighbors and itself, effectively performing a form of feature smoothing over the graph. The update rule for a single GCN layer is defined as:

$$H^{k+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^k W^k) \tag{2.11}$$

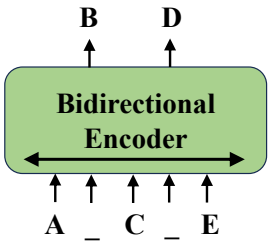
where $\tilde{A} = A + I$ is the adjacency matrix of the graph with added self-loops, $\tilde{D}$ is the degree matrix of $\tilde{A}$, H^k is the node feature matrix at layer k, and W^l is a learnable weight matrix.

2.3.3 Pre-trained Language Models & Large Language Models

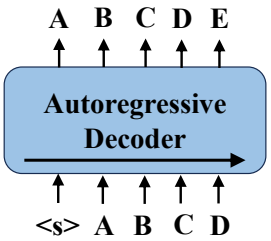
Traditionally, NLP systems relied heavily on task-specific models and manual feature engineering, which often led to fragmented solutions with limited generalization capabilities. To address this, researchers have long sought to develop general-purpose language representations by pre-training on large-scale unlabeled text (Mikolov et al., 2013). However, progress has been constrained by the limited efficiency of sequence models such as LSTMs. The emergence of Transformers has effectively addressed this issue, as they not only model dependencies between words effectively, but also support parallelized training. This has led to the development of Pre-trained Language Models (PLMs), which are typically first trained on massive amounts of raw text using self-supervised learning, and then fine-tuned for specific NLP tasks.

Pre-trained Language Models

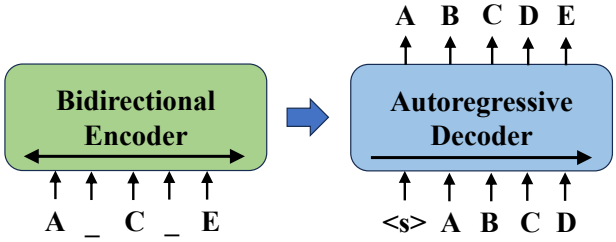
The first successful Transformer-based pre-trained language model (PLM) is Generative Pre-trained Transformer (GPT), introduced by Radford et al. (2018). It is a *decoder-only* model



(a) **BERT**: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) **GPT**: Tokens are predicted autoregressively, meaning GPT can be used for generation. However, words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) **BART**: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitrary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Fig. 2.5 A schematic comparison between GPT (decoder-only), BERT (encoder-only) and BART (encoder-decoder).

that uses a causal language modeling (CLM) objective for pretraining, in which the model is trained to predict the next word in a sequence using only the preceding words, never those that follow. Due to its left-to-right training approach, GPT cannot leverage bidirectional context, which is crucial for many natural language understanding tasks. The *encoder-only* model BERT (Bidirectional Encoder Representations from Transformers), proposed by Devlin et al. (2019), addresses this limitation with a different training objective called masked language modeling (MLM). Specifically, BERT randomly masks a portion of the input tokens and trains the model to predict these masked tokens based on the surrounding unmasked context. While effective for understanding tasks, BERT is not suitable for text generation because it is trained to predict randomly masked tokens rather than generating sequences left-to-right. Therefore, some works propose using an *encoder-decoder* architecture that combines both

	Model	Architecture	Size
PLMs	Bert	Encoder-only	110M
	RoBERTa	Encoder-only	125M
	XLNet	Decoder-only	340M
LLMs	LLaMA 2	Decoder-only	7B, 13B, 70B
	LLaMA 3.1	Decoder-only	8B, 70B, 405B
	GPT-4o	Decoder-only	>200B

Table 2.12 Examples of Pre-trained Language Models (PLMs) and Large Language Models (LLMs) used in this thesis.

types of training. For example, T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) input masked text into an encoder and then use a decoder to reconstruct the original text by generating words sequentially. Figure 2.5 shows a comparison between encoder-only, decoder-only, and encoder-decoder PLMs.

Large Language Models

Large Language Models (LLMs) are a significant evolution of PLMs, characterized by scaling up both the model architecture and the volume of training data. LLMs, such as GPT-4 (OpenAI, 2023), Gemini (DeepMind, 2024), and LLaMa (Touvron et al., 2023), are trained as autoregressive models (i.e., decoder-only) to predict the next token in a sequence, enabling them to generate coherent and contextually rich text across diverse domains. The central idea behind this transition is the hypothesis that increasing model size, data diversity, and training duration leads to emergent abilities not observed in smaller models (Kaplan et al., 2020), such as in-context learning, reasoning over long contexts, and multilingual generalization. Unlike earlier PLMs, which required task-specific fine-tuning, many LLMs demonstrate strong zero-shot and few-shot capabilities using prompt-based learning. This shift reflects a broader trend in NLP toward general-purpose models that can perform a wide range of tasks with minimal adaptation.

Table 2.12 provides a summary of both Pretrained Language Models (PLMs) and Large Language Models (LLMs) used in the experiments presented in this thesis, including details on their architecture and size.

2.3.4 Model Adaptation

The adaptation of pre-trained language models (PLMs) and large language models (LLMs) to specific natural language processing (NLP) tasks has emerged as a central theme in recent

	Model Type	Parameter Updates	Data Requirement
Fine-tuning	PLMs or LLMs	Yes	Large amounts of labeled data
Zero-shot Prompting	LLMs	No	No labeled data needed
In-context Learning	LLMs	No	Few labeled examples

Table 2.13 Comparison of model adaptation strategies, including fine-tuning, zero-shot prompting, and in-context learning.

research (Han et al., 2024). In this context, we present the three most common approaches to model adaptation: fine-tuning, zero-shot prompting, and in-context learning (see Table 2.13 for a comparison between them).

Fine-tuning

Fine-tuning is a widely adopted approach for adapting PLMs and LLMs to specific downstream NLP tasks (Peters et al., 2019). In this paradigm, a model that has been pre-trained on a large-scale general-purpose corpus is further trained on a smaller, task-specific dataset. This additional training allows the model to adjust its parameters to better capture the nuances and requirements of the target task, such as sentiment analysis or question answering. Fine-tuning typically results in improved performance compared to using the pre-trained model alone, as it leverages both the general linguistic knowledge acquired during pre-training and task-specific patterns learned during adaptation.

Zero-shot Prompting

Zero-shot prompting is a technique used with LLMs (not for PLMs) where the model is given a task instruction or question without any examples of how to perform the task (Kojima et al., 2022). Instead of training or fine-tuning the model on task-specific data, zero-shot prompting relies entirely on the model's pre-existing knowledge, acquired during pre-training, to understand and complete the task based on the prompt alone. Here is an example:
Task*: Translate English to German*
Prompt*: "Translate the following sentence to German: 'Heidelberg is a very beautiful city.'"*
Output*: "Heidelberg ist eine sehr schöne Stadt."*

In-context Learning

In-context learning is an adaptation method for large language models (LLMs) that enables them to perform specific tasks without modifying their underlying parameters (Brown et al., 2020). Instead of traditional training or fine-tuning, the model is provided with a

sequence of task demonstrations, examples of input-output pairs, directly within the prompt at inference time. By conditioning on these examples, the model can infer the task and generate appropriate outputs for new inputs. This approach leverages the model's ability to recognize patterns and generalize from context, making it highly flexible and suitable for scenarios where labeled training data is scarce or rapid deployment is needed. Here is an example:

Prompt:
Translate the following English sentences to German:

English: The cat is sleeping.
German: Die Katze schläft.

English: I would like a cup of coffee.
German: Ich hätte gerne eine Tasse Kaffee.

English: Heidelberg is a very beautiful city.
German:

Output:
Heidelberg ist eine sehr schöne Stadt.

2.3.5 MASK Stragegy in Transformers

Mask strategy is a widely used approach in Transformer-based models for controlling information flow during training and inference (Radford et al., 2018; Devlin et al., 2019; Dong et al., 2019). By selectively masking input tokens or attention connections, the model is constrained to access only a subset of available information.

In pre-trained models and large language models, masking is mainly used to define training objectives and prediction settings. For example, masked language models, such as BERT (Devlin et al., 2019), mask a portion of the input words and train the model to recover those words, thus achieving self-supervised learning of contextual representations (see Figure 2.5a). In contrast, causal language models, such as GPT (Radford et al., 2018), apply causal attention masks to prevent words from focusing on future positions, so each word is generated using only its left-hand context (see Figure 2.5b).

Beyond general information control, mask strategies can also be used to inject prior knowledge into Transformer models. By constructing attention masks based on linguistic or

task-related constraints, such as syntactic structures (Li et al., 2021), discourse relations (Mihaylov and Frank, 2019), or knowledge graph (Liu et al., 2020a), the model's attention can be restricted to meaningful token-to-token interactions. Under such designs, only predefined token pairs are allowed to attend to each other, while irrelevant connections are blocked. This form of prior-guided masking enables the model to better utilize structured knowledge without modifying the core Transformer architecture.

Chapter 3

Related Work

This chapter reviews prior research on coherence modeling and discourse relation classification, which together serve as the foundation for the present study.

3.1 Coherence Modeling

3.1.1 Entity-based Methods

Coherence modeling has long been a central topic in discourse analysis and natural language processing (NLP), with researchers aiming to capture how texts flow logically and meaningfully. Early coherence modeling approaches are grounded in linguistic theories, such as Centering Theory (Grosz et al., 1995), which describes how entities shift focus across different discourse segments to maintain coherence. These models provide the foundation for more structured representations of text organization.

Entity Grid

A significant milestone in coherence modeling is the entity grid model, proposed by Barzilay and Lapata (2008). Given a text, this approach first identifies the entities and their grammatical roles: subjects (S), objects (O), or others roles (X). It then represents the text as a two-dimensional grid, where each row corresponds to a sentence and each column to an entity. Each cell (s_i, e_j) in the grid indicates the grammatical role of the j-th entity e_j in the i-th sentence s_i. An example is shown in Figure 3.1. Finally, the entity grid method counts the occurrences of all N-gram transitions among the syntactic categories S, O, X, and –, normalizes these counts, and uses them as coherence patterns. Taking the grid in Figure 3.1 as an example, all 2-gram transitions among S, O, X, and - include: "S S", "S O", "S X", "S –", "O S", "O O", "O X", "O –", "X S", "X O", "X X", "X –", "– S", "– O", "– X", and "–

1. [The Justice Department]$_S$ is conducting an [anti-trust trial]$_O$ against [Microsoft Corp.]$_X$ with [evidence]$_X$ that [the company]$_S$ is increasingly attempting to crush [competitors]$_O$.
2. [Microsoft]$_O$ is accused of trying to forcefully buy into [markets]$_X$ where [its own products]$_S$ are not competitive enough to unseat [established brands]$_O$.
3. [The case]$_S$ revolves around [evidence]$_O$ of [Microsoft]$_S$ aggressively pressuring [Netscape]$_O$ into merging [browser software]$_O$.
4. [Microsoft]$_S$ claims [its tactics]$_S$ are commonplace and good economically.
5. [The government]$_S$ may file [a civil suit]$_O$ ruling that [conspiracy]$_S$ to curb [competition]$_O$ through [collusion]$_X$ is [a violation of the Sherman Act]$_O$.
6. [Microsoft]$_S$ continues to show [increased earnings]$_O$ despite [the trial]$_X$.

(a) A text with recognized entities and their grammatical roles.

	Department	Trial	Microsoft	Evidence	Competitions	Markets	Products	Brands	Case	Netscape	Software	Tactics	Government	Suit	Earnings
1	S	O	S	X	O	-	-	-	-	-	-	-	-	-	-
2	-	-	O	-	-	X	S	O	-	-	-	-	-	-	-
3	-	-	S	O	-	-	-	-	S	O	O	-	-	-	-
4	-	-	S	-	-	-	-	-	-	-	-	S	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	S	O	-
6	-	X	S	-	-	-	-	-	-	-	-	-	-	-	O

(b) The entity grid of the text.

Fig. 3.1 An illustrative example of the entity grid. Given a text, all entities and their grammatical roles, subject (S), object (O), or other (X), are identified. The text is then represented as a two-dimensional grid, where each row corresponds to a sentence and each column to an entity. Each cell (s_i, e_j) in the grid denotes the grammatical role of the j-th entity e_j in the i-th sentence s_i.

–". The number of occurrence of "S O" between adjacent sentences is 1, and the normalized value is 0.013 (1/75, where 75 is the total number of 2-grams in the grid).

The entity grid model has been enhanced by numerous subsequent studies. For instance, Filippova and Strube (2007) extend the original model by accounting not only for transitions involving the same entities but also for those between semantically related entities. To determine semantic relatedness, they employ the WikiRelate! API (Strube and Ponzetto, 2006) to compute a relatedness score between entities; if this score exceeds a predefined threshold, the entities are considered related. Elsner and Charniak (2011) propose an extension to the entity grid model that differentiates between various types of entities. In the standard entity grid, no information about the nature or importance of the entity is considered, i.e., each entity is treated equally in terms of transition probability. To address this limitation, their approach incorporates additional information derived from syntactic structure, named entity recognition, and statistical data from an external coreference corpus into the entity grid representation.

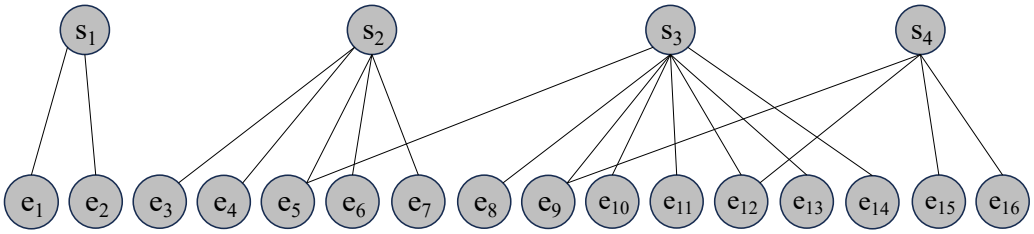
Entity Graph

Another prominent early approach to coherence modeling is the Entity Graph, proposed by Guinaudeau and Strube (2013), which aims to refine the entity grid model. Unlike the entity grid, which represents coherence through discrete grammatical role transitions of entities across sentences, the entity graph models sentence-entity relationships as a bipartite graph and assesses coherence based on the structural properties of the resulting graph. As with the entity grid, the entity graph approach begins by identifying all entities in a given text. It then constructs a bipartite graph linking entities and sentences, where an edge is established between a sentence and an entity if the sentence contains that entity. This bipartite graph is subsequently transformed into a sentence graph through a one-mode projection onto the sentence nodes, such that an edge is created between two sentence nodes if they share at least one common entity. Finally, the coherence of the text is assessed by calculating the average outdegree of the sentence graph. Figure 3.2 shows an example of the entity graph.

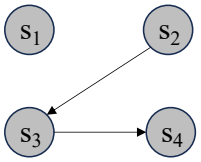
Several studies have aimed to improve the entity graph for coherence modeling. Mesgar and Strube (2015) argue that relying solely on the out-degree is inadequate for capturing the structural properties of the sentence graph. Motivated by the functional sentence perspective of text coherence (Danes, 1974), they enhance the entity graph with graph-based features extracted from text structures. Specifically, they extract subgraphs from the sentence graph that represent the local structure of the text, and use these subgraphs as patterns to evaluate local coherence. Building on a similar idea, Mesgar and Strube (2016) use subgraph patterns for coherence assessment, but utilize word embeddings to construct the sentence graph. In this approach, two sentences are connected if they contain entities with a similarity

1. The Turkish [government]$_{e_1}$ fell after mob-tie [allegations]$_{e_2}$.
2. [Turkey's]$_{e_3}$ [constitution]$_{e_4}$ mandates a [secular]$_{e_5}$ [republic]$_{e_6}$ despite its Muslim [majority]$_{e_7}$.
3. [Military]$_{e_8}$ and [secular]$_{e_5}$ [leaders]$_{e_9}$ pressured [President]$_{e_{10}}$ [Demirel]$_{e_{11}}$ to keep the Islamic-oriented [Virtue]$_{e_{12}}$ [Party]$_{e_{13}}$ on the [fringe]$_{e_{14}}$.
4. [Business]$_{e_{15}}$ [leaders]$_{e_9}$ feared [Virtue]$_{e_{12}}$ would alienate the [EU]$_{e_{16}}$.

(a) A text with recognized entities.



(b) The bipartite graph between sentences and entities of the text.



(c) The sentence graph of the text.

Fig. 3.2 An illustrative example of the entity graph. Given a text with identified entities, a bipartite graph is constructed linking sentences and entities. This bipartite graph is subsequently transformed into a sentence graph via a one-mode projection onto the sentence nodes.

score exceeding a predefined threshold, determined by the cosine similarity between the embeddings of those entities.

Neural Coherence Models

With the advent of deep learning, neural models have become dominant in coherence research. Early neural models for coherence modeling primarily focus on learning improved sentence representations. For instance, Li and Hovy (2014) and Xu et al. (2019) develop models that assess coherence by training neural encoders to distinguish coherent texts from incoherent ones. Other studies have aimed to extend the traditional entity grid model by incorporating

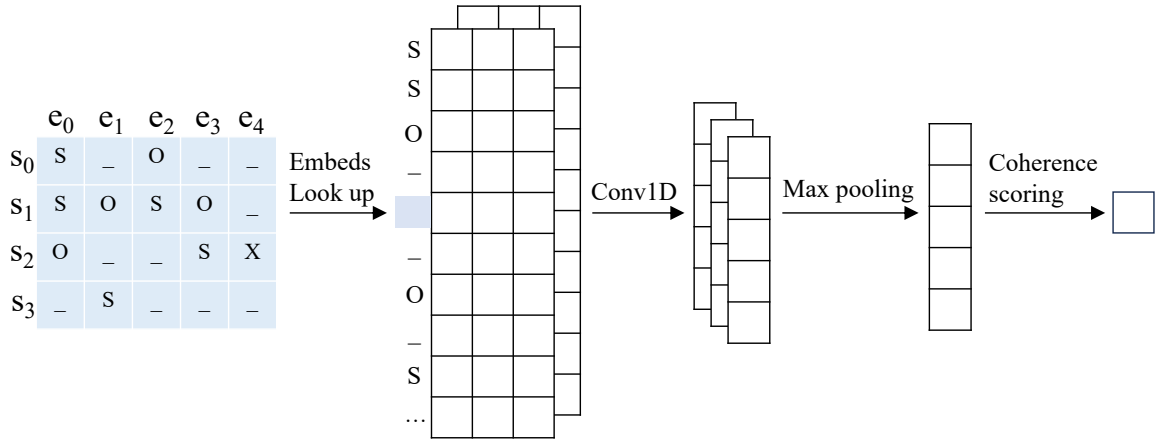


Fig. 3.3 Neural entity grid model proposed by Tien Nguyen and Joty (2017). The model is trained using a pairwise ranking approach with shared parameters for positive and negative documents.

neural components. Tien Nguyen and Joty (2017) and Joty et al. (2018), for example, propose using a convolutional neural network to capture transitions between entities, replacing the n-gram patterns used in the standard entity grid approach. An example of this approach is shown in Figure 3.3.

More recently, several studies have explored whether neural models that mimic Centering Theory can lead to improved coherence modeling. Mesgar and Strube (2018) propose a model designed to capture the most relevant elements between adjacent sentences, analogous to the focus in Centering Theory. Specifically, they employ an LSTM to obtain hidden states for the words in each sentence and then identify the words with the highest similarity across adjacent sentences. The hidden states of these most similar word pairs are treated as salient information. A CNN is subsequently used to extract patterns from the changes in this salient information across the text; these patterns are then utilized for coherence evaluation.

Jeon and Strube (2020a) propose a coherence model designed to approximate Centering Theory for tracking shifts in discourse focus across segments. This model identifies the focus of each sentence within its contextual setting, consistent with Centering Theory's emphasis on monitoring discourse entities to maintain coherence. By capturing shifts in sentence-level focus, the model constructs hierarchical discourse structures that reflect the relationships among different segments of the text. These structures are then integrated into a structure-aware transformer model, improving its capacity to evaluate coherence by incorporating both local and global connectivity information, as illustrated in Figure 3.4. One shortcoming of this approach is that it computes coherence based on connections between any words, including sub-words or function words. This can lead the model to focus on

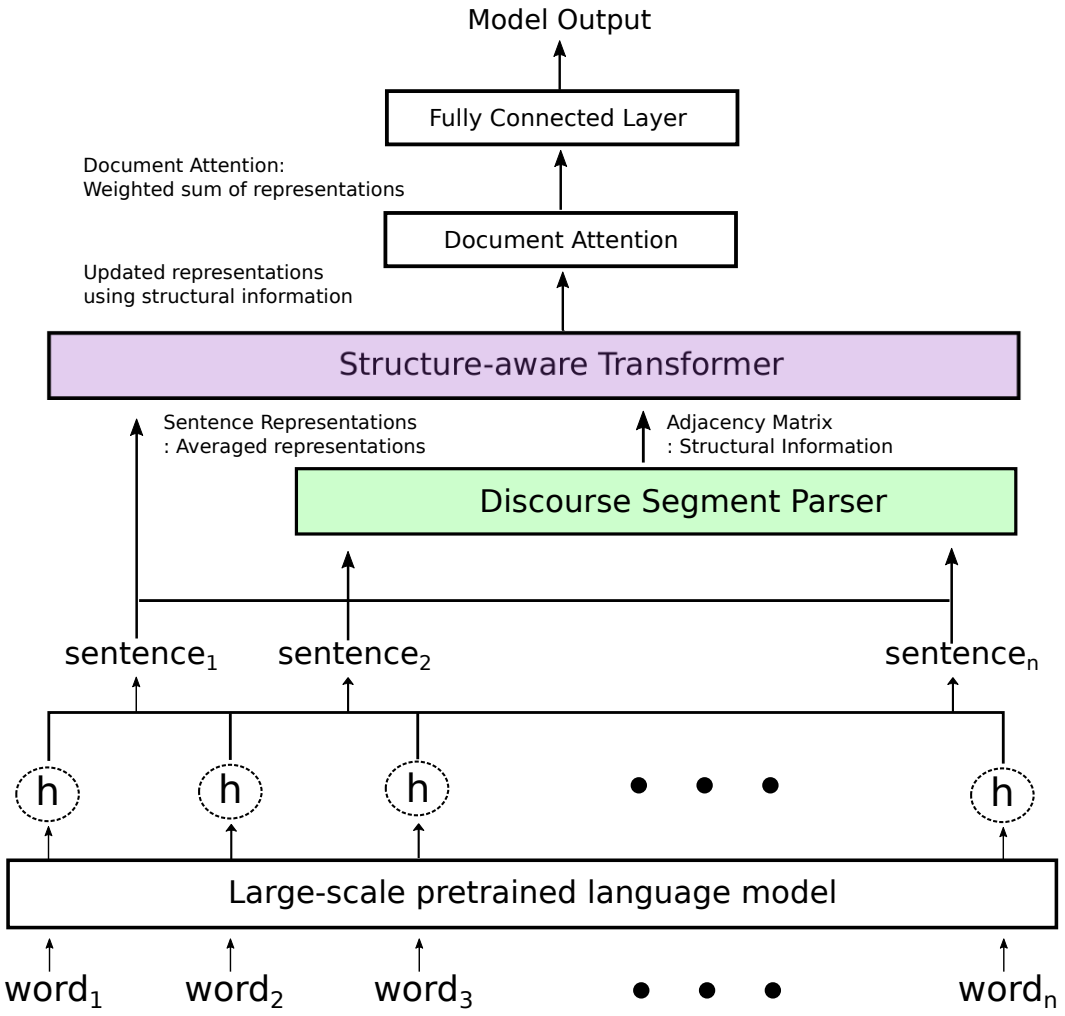


Fig. 3.4 An overview of the model proposed by Jeon and Strube (2020a). The approach approximates Centering Theory to track shifts in discourse focus across segments and constructs hierarchical discourse structures that represent relationships between different segments of the text (see the Discourse Segment Parser). These structures are then utilized by a structure-aware Transformer for coherence assessment.

irrelevant or spurious information, and it is not linguistically sound, as coherence theories typically emphasize entities. To address this, Jeon and Strube (2022) refine the method by restricting the focus to noun phrases and proper names.

While the entity-based models discussed above have demonstrated strong performance, they focus exclusively on identifying entity-based patterns within individual documents, overlooking the underlying relationships between documents. Coherence refers to how sentences within a text are connected. Theoretically, documents with similar entity structures are likely to exhibit comparable levels of coherence, which can serve as valuable prior

knowledge for coherence modeling. In Chapter 4, we demonstrate that explicitly modeling structural similarity between documents contributes to a more effective coherence assessment.

3.1.2 Discourse Relation-based Methods

Discourse relations, such as *Cause* and *Contrast*, describe the logical relation between two text spans. In discourse coherence theory (Crothers, 1978; van Dijk and Kintsch, 1983), discourse relations between text spans play a key role in establishing the coherence of texts. This has inspired several studies utilizing discourse relations for coherence modeling.

Lin et al. (2011) is among the few studies that utilizes discourse relations for coherence assessment. Their approach is inspired by the entity grid model. Similar to that model, they construct a matrix where the rows represent sentences and the columns represent terms, respectively (see Figure 3.5b). However, instead of simply marking the syntactic roles of terms, they populate the matrix with information about discourse relations. Specifically, given a text, the method first identifies all the discourse arguments and determines the discourse relations between each pair of arguments (see Figure 3.5a). It then extracts, for each sentence, the discourse arguments in which a given term is involved, along with the associated discourse relations. For example, consider the term "Cananea" in Figure 3.5. It occurs in three locations: the first sentence (S_1), the first clause of the third sentence ($C_{3.1}$), and the third clause of the fourth sentence ($C_{4.3}$). Sentence S_1 serves as the first argument in a Comparison relation between S_1 and S_2; therefore, the cell corresponding to (S_1, Cananea) is assigned the value "Comp.Arg1". Similarly, clause $C_{3.1}$ functions as the second argument in a Comparison relation between S_2 and S_3, the first argument in a Temporal relation between $C_{3.1}$ and $C_{3.2}$, and the first argument in an Expansion relation between S_3 and S_4. As a result, the cell (S_3, Cananea) is assigned the combined value "Comp.Arg2, Temp.Arg1, Exp.Arg1". The method then computes the frequencies of n-gram transitions between discourse roles (i.e., relation types combined with argument positions), normalizes these counts, and treats them as coherence patterns, in a manner analogous to the entity grid model. This approach is later extended by Feng et al. (2014), who replace the PDTB discourse relations with those defined in the RST framework.

However, Mesgar and Strube (2015) argue that these methods are conceptually flawed, as they treat discourse relations as properties of entities, which contradicts the established understanding of discourse relations as operating between sentences or elementary discourse units. Additionally, the effectiveness of these approaches was limited by the poor performance of discourse parsers at the time. For example, the PDTB parser used by Lin et al. (2011) achieve an F1-score of only 25.46 in identifying top-level implicit discourse relations. This limitation also discouraged subsequent research from incorporating discourse relations into

[Japan normally depends heavily on the Highland Valley and **Cananea** mines as well as the Bougainville mine in Papua New Guinea.]$_{S_1}$ [Recently, Japan has been buying copper elsewhere.]$_{S_2}$ [[But as Highland Valley and **Cananea** begin operating,]$_{C_{3.1}}$ [they are expected to resume their roles as Japan's suppliers.]$_{C_{3.2}}$]$_{S_3}$ [[According to Fred Demler, metals economist for Drexel Burnham Lambert, New York,]$_{C_{4.1}}$ ["Highland Valley has already started operating]$_{C_{4.2}}$ [and **Cananea** is expected to do so soon."]$_{C_{4.3}}$]$_{S_4}$

5 discourse relations are present in the above text:

1. Implicit Comparison between S_1 as Arg_1, and S_2 as Arg_2
2. Explicit Comparison using "but" between S_2 as Arg_1, and S_3 as Arg_2
3. Explicit Temporal using "as" within S_3 (Clause $C_{3.1}$ as Arg_1, and $C_{3.2}$ as Arg_2)
4. Implicit Expansion between S_3 as Arg_1, and S_4 as Arg_2
5. Explicit Expansion using "and" within S_4 (Clause $C_{4.2}$ as Arg_1, and $C_{4.3}$ as Arg_2)

(a) An excerpt with four contiguous sentences from wsj 0437, showing five gold standard discourse relations. "Cananea" is highlighted for illustration.

	copper	cananea	operat	depend	…
S_1	nil	Comp.Arg1	nil	Comp.Arg1	
S_2	Comp.Arg2 Comp.Arg2	nil	nil	nil	
S_3	nil	Comp.Arg2 Temp.Arg1 Exp.Arg1	Comp.Arg2 Temp.Arg1 Exp.Arg1	nil	
S_4	nil	Exp.Arg2	Exp.Arg1 Exp.Arg2	nil	

(b) Discourse role matrix for the text above. Rows correspond to sentences, columns to stemmed terms, and cells contain extracted discourse roles.

Fig. 3.5 An illustrative example of the method proposed by Lin et al. (2011). Given a text, the method first identifies all discourse arguments and determines the discourse relations between each pair of arguments. It then extracts, for each sentence, the discourse arguments involving a given term, along with their associated discourse relations.

coherence modeling, prompting the question: Can discourse relations contribute to neural coherence modeling?

In Chapter 5, we demonstrate that the performance of the PDTB discourse parser can be substantially improved by leveraging pre-trained language models such as RoBERTa, combined with a carefully designed approach inspired by human annotation practices. Building on this enhanced parser, Chapter 7 shows that when discourse relations are employed in a more linguistically grounded manner, they provide significant benefits for coherence assessment.

The aforementioned studies enhance coherence models either from an entity-based perspective or a discourse-based perspective, but none consider both features simultaneously. In practice, entity cues and discourse relations often coexist and interact in complex ways. Therefore, integrating both types of information has the potential to further improve performance. In Chapter 8, we explore two approaches that jointly model entities and discourse relations for coherence assessment and demonstrate that they significantly outperform strong baselines that consider only one of these features, or neither.

3.2 Discourse Relation Classfication

The task of Discourse Relation Classification (DRC), identifying the logical or rhetorical relations between textual units, has undergone significant evolution over the past two decades.

Early work in discourse analysis is predominantly rule-based, heavily influenced by Rhetorical Structure Theory (RST, Mann and Thompson, 1988). These works aim to construct full hierarchical discourse trees, which requires segmenting text into elementary discourse units, identifying discourse relations, and recursively building a coherent hierarchical structure. A notable contribution in this line is Marcu (2000b), who develops one of the first end-to-end discourse parsers based on RST, utilizing decision trees and syntactic templates. Building on this foundation, subsequent studies apply increasingly powerful learning models, ranging from early statistical classifiers, such as support vector machines (Hernault et al., 2010), to recent neural architectures (Nguyen et al., 2021; Kobayashi et al., 2022; Yu et al., 2022b; Maekawa et al., 2024), which yield more robust and accurate discourse trees. These prasers generally adopt either a bottom-up or a top-down strategy (Kobayashi et al., 2022; Maekawa et al., 2024). Bottom-up approaches (Feng and Hirst, 2014; Ji and Eisenstein, 2014) recursively merge adjacent text spans, typically starting from individual EDUs. At each step, a classifier determines whether to combine spans and assigns the corresponding nuclearity roles and discourse relations until the full tree is rooted. Top-down

approaches (Lin et al., 2019; Zhang et al., 2020) treat the entire document as a single span and recursively partition it into smaller constituent spans. These models identify the optimal split point within a span and simultaneously predict the nuclearity and relation labels for the resulting sub-spans.

The release of the Penn Discourse Treebank (PDTB) by Prasad et al. (2008) marks a significant shift toward data-driven approaches. The PDTB annotates discourse relations between pairs of text spans, distinguishing between explicit relations (signaled by discourse connectives) and implicit ones (inferred without connectives). This resource enables the application of supervised learning techniques to discourse relation classification. Crucially, as established by Pitler et al. (2008, 2009), discourse connectives serve as the most reliable surface indicators of discourse coherence, allowing for straightforward mapping to relation classes. However, they also highlight that the problem becomes substantially more complex for implicit units, where the underlying relations are not lexically signaled but must be inferred by interlocutors or readers. This inferential process involves a broader field of linguistic analysis than mere structural parsing, encompassing pragmatic reasoning (Torabi Asr and Demberg, 2012), lexical semantics (Pitler et al., 2009), and the synthesis of world knowledge (Kishimoto et al., 2018) to reconstruct the intended coherence. Their findings highlight the particular challenge of classifying implicit relations, which soon became a central focus in the field.

Implicit Discourse Relation Classification

In response to this challenge, much of the subsequent research focuses on directly predicting implicit discourse relations from the input arguments. Early efforts treat implicit relation classification as a supervised learning problem, employing lexical features, syntactic cues, and shallow semantic indicators. Notably, Lin et al. (2009) incorporat word pairs extracted from argument spans into the model to model lexical associations that indicate discourse relations, thus establishing a strong baseline for implicit relation classification. Rutherford and Xue (2014) has expanded this direction by introducing richer semantic representations, including polarity, modality, and semantic role information. As the field progressed, researchers increasingly adopted neural networks for their ability to automatically learn rich semantic and syntactic representations from raw text, thereby reducing dependence on manual feature engineering. Ji and Eisenstein (2015) pioneer neural models for discourse relation classification by introducing distributed representations of argument spans and an attention-based fusion mechanism. Qin et al. (2016a,b) further advance the field with bi-directional LSTM models that incorporate pairwise interaction features, surpassing traditional feature-based methods.

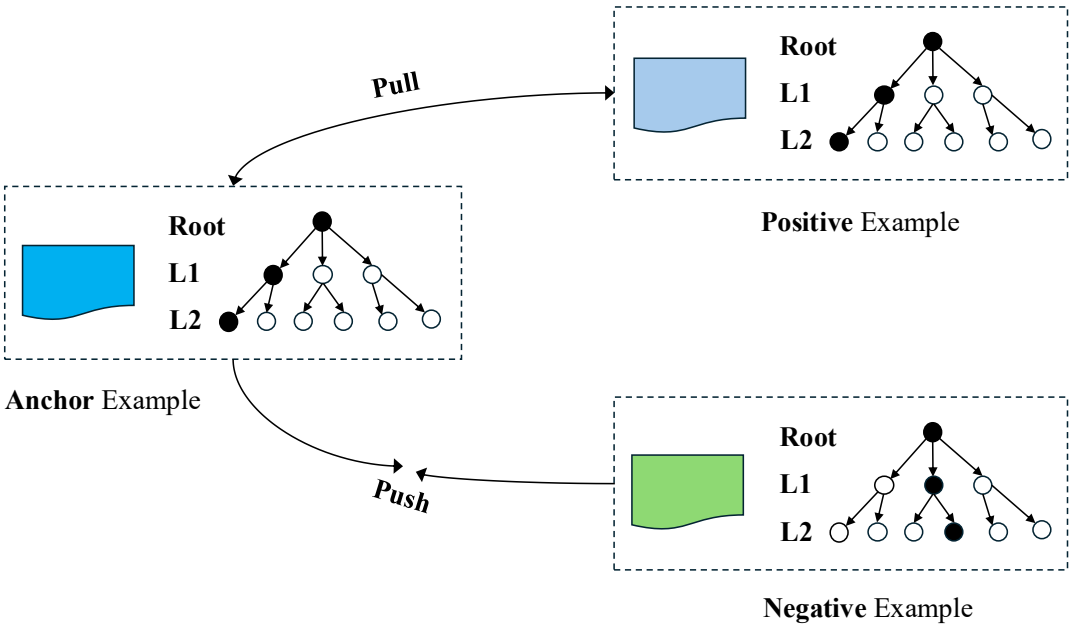


Fig. 3.6 The overall idea of Long and Webber (2022). Given an anchor instance, positive and negative examples are identified within a training batch according to the PDTB sense hierarchy. The contrastive objective encourages instances sharing higher-level discourse senses to be closer in the representation space, while pushing apart instances belonging to different branches of the sense hierarchy.

Beyond purely lexical or compositional semantics, some studies argue that the classification of implicit discourse relations inherently involves pragmatic reasoning and world knowledge. Torabi Asr and Demberg (2012) analyze the cognitive basis of implicit relations, showing that readers rely on expectations of coherence and causality. Complementary computational studies have incorporated external knowledge sources or underlying semantic abstractions to model such reasoning processes (Kishimoto et al., 2018). These findings suggest that successful implicit relation classification extends beyond structural parsing, requiring the integration of lexical semantics, pragmatic inference, and background knowledge.

More recently, pre-trained language models (PLMs) have become the state-of-the-art for implicit discourse relation classification. Shi and Demberg (2019b) highlight that BERT's next-sentence prediction task benefits cross-domain classification of implicit relations. Long and Webber (2022) propose to incorporate the hierarchical structure of discourse relation senses into contrastive learning, encouraging instances that share higher-level senses to have similar representations while pushing apart instances belonging to different branches of the sense hierarchy. As illustrated in Figure 3.6, the framework pulls anchor and positive

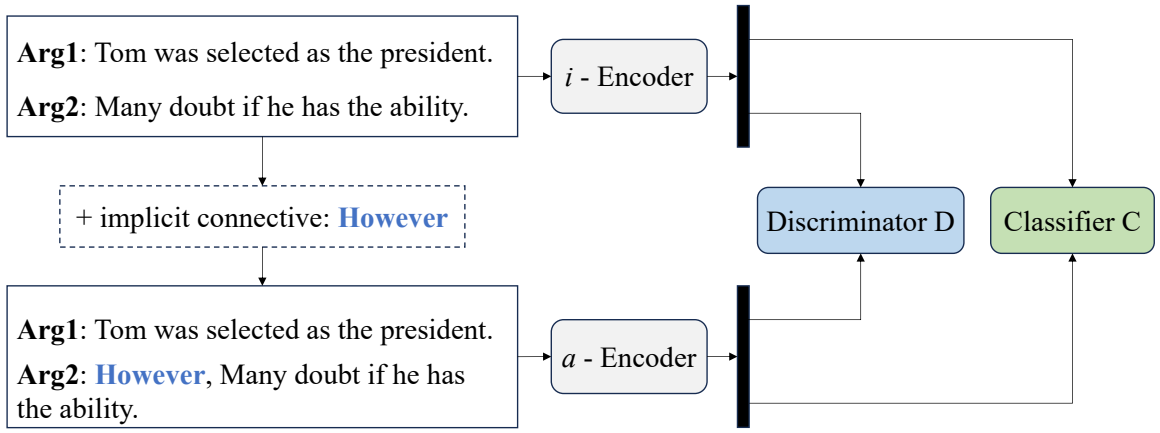


Fig. 3.7 The architecture of the adversarial model proposed by Qin et al. (2017). The framework contains three main components: 1) an implicit relation network *i*-encoder over raw sentence arguments, 2) a connective-augmented relation network *a*-encoder whose inputs are augmented with implicit connectives, and 3) a discriminator distinguishing between the features by the two networks. The features are fed to the final classifier for relation classification. The discriminator and *i*-encoder form an adversarial pair for feature imitation. At test time, the implicit network *i*-encoder with the classifier is used for prediction.

examples closer in the representation space, while separating negative examples from the anchor.

The significant performance gap between explicit and implicit discourse relation classification has motivated a line of work exploring the use of discourse connectives to address the implicit case (refer to as **connective-enhanced methods**). Zhou et al. (2010) introduce a pipeline approach that leverages connectives recovered from an n-gram language model to aid in recognizing implicit relations. Their findings demonstrate that incorporating these recovered connectives as features can achieve performance comparable to a strong baseline. This pipeline-based strategy has been refined through the use of pre-trained language models (Kurfalı and Östling, 2021; Jiang et al., 2021) and prompt-based techniques (Xiang et al., 2022; Zhou et al., 2022). However, some studies (Qin et al., 2017; Xiang and Wang, 2023) have highlighted a key limitation of pipeline methods: the accumulation of cascading errors. In response, recent work has shifted toward end-to-end neural architectures. For instance, Qin et al. (2017) propose a feature imitation framework designed to transfer information from explicit to implicit discourse relation classification. Their method introduces two encoders: one is trained on the original implicit discourse instances (without connectives in the input), while the other is trained on the same set of examples with implicit connectives included in the input. The implicit encoder is encouraged to learn representations that resemble those produced by the connective-aware encoder through an adversarial training objective. By

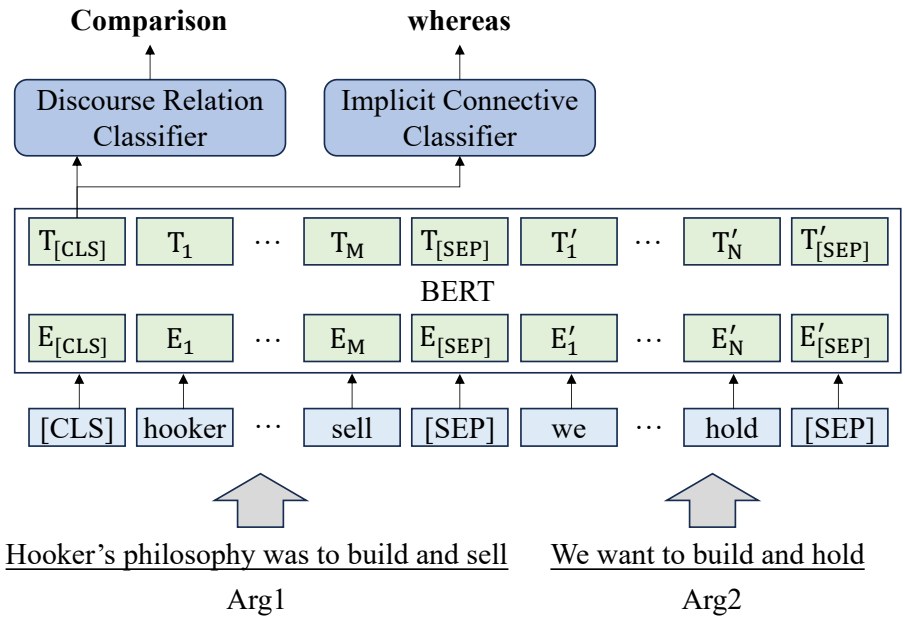


Fig. 3.8 Overview of the multi-task model proposed by Kishimoto et al. (2020). The input is an implicit argument pair randomly selected from the training data, where annotators have provided an implicit connective for each pair. BERT is trained to predict the implicit connective and the discourse relation.

aligning implicit representations with connective-enhanced ones, the model leverages the strong signaling effect of discourse connectives while remaining applicable to implicit relation classification at test time (see Figure 3.7). Similarly, Shi and Demberg (2019a) propose a sequence-to-sequence encoder-decoder model that generates implicit discourse connectives from text. By treating connective generation as an auxiliary task, the encoder learns richer representations of discourse arguments, which are then used for implicit discourse relation classification. Kishimoto et al. (2020) further explores a BERT-based multi-task learning framework, in which connective prediction is jointly learned with implicit discourse relation classification, allowing the model to capture discourse-level information better and improving overall performance (see Figure 3.8).

However, we argue that these connective-enhanced methods remain suboptimal because connectives are still not explicitly present in the input text. This limitation is underscored by the findings of Kishimoto et al. (2020), who show that incorporating implicit connective prediction as an auxiliary training objective yields only marginal improvements in classifying implicit relations on the PDTB 2.0 dataset. In Chapter 5, we propose an end-to-end approach that addresses this limitation by explicitly generating a connective between two arguments and incorporating it into the input for relation classification. This method is inspired by

the human annotation process used for implicit discourse relation labeling. Our results demonstrate that this approach significantly outperforms previous connective-enhanced methods.

Explicit to Implicit Discourse Relation Classification

Corpora of explicit discourse relations are relatively easy to construct, both manually and automatically, because connectives serve as clear indicators of the underlying relations (Pitler and Nenkova, 2009). In contrast, annotating implicit relations is far more challenging and costly, as it requires inferring the relation from context without explicit markers. This difficulty has led many early studies to leverage explicit examples to train models for classifying implicit relations, a strategy often referred to as *explicit to implicit relation recognition*.

Marcu and Echihabi (2002) train the first classifier for implicit intra-sentential discourse relations using explicitly marked examples from a raw English corpus, BLIPP (Charniak, 2000), and the RST Treebank (Carlson et al., 2001). Lapata and Lascarides (2004) present a similar approach using BLIPP but focus on sentence-internal temporal relations. Blair-Goldensohn et al. (2007) extend this line of work by refining the training process using parameter optimization, topic segmentation, and syntactic parsing on the Gigaword (Graff and Cieri, 2003) and PDTB (Prasad et al., 2004). These three works are evaluated on test sets constructed in the same manner as the training set and show good performance. Sporleder and Lascarides (2008a) and Lin et al. (2009) investigate the applicability of this approach to real implicit scenarios and find that performance degrades substantially. They claim, based on a manual analysis of a few instances, that the linguistic dissimilarities between explicit and implicit examples may be the cause. However, a corpus-level empirical analysis is not provided to establish how widespread the problem is.

More recent work has focused on improving the performance in *explicit to implicit discourse relation recognition*. Wang et al. (2012) propose to use typical examples with linguistic structure shared between explicit and implicit relations for training. Ji et al. (2015) adopt techniques such as resampling and transfer learning to handle the mismatched label distribution between explicit and implicit corpora. Huang and Li (2019) follow a similar domain adaptation idea but focus on minimizing the distance between representations of explicit and implicit examples using an adversarial training framework. Kurfalı and Östling (2021) tackle this task from a distant supervision perspective. However, little attention has been paid to the underlying causes of the poor results.

In Chapter 6, we show that one cause for this failure is a label shift after connectives are eliminated. We present both manual and empirical evidence to demonstrate the existence of such a shift in the explicit corpus and investigate two strategies to mitigate it.

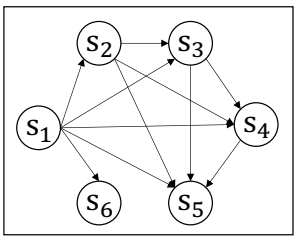
Chapter 4

Document Structure Similarity-Enhanced Coherence Modeling

Entity-based methods have been extensively developed for coherence modeling. Early approaches, such as the Entity Grid (Barzilay and Lapata, 2008), assess coherence by capturing entity transitions between adjacent sentences. Other methods, like the Entity Graph (Guinaudeau and Strube, 2013), evaluate coherence by leveraging structural properties of the sentence graph of a document. More recently, researchers have explored the use of neural network architectures, including Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and Transformers, for entity-based coherence modeling (Tien Nguyen and Joty, 2017; Mesgar and Strube, 2018; Farag and Yannakoudakis, 2019; Jeon and Strube, 2020a, 2022). However, these methods primarily focus on feature extraction within individual documents, overlooking potential correlations between documents.

In this chapter, we first present the motivation of leveraging structural similarity between documents for coherence modeling. Next, we introduce a graph-based approach that explicitly connects structurally similar documents and employs Graph Convolutional Networks (GCNs) to capture inter-document connectivity. Finally, we report experimental results comparing our model with previous methods on two widely used benchmark corpora, followed by in-depth analyses demonstrating the effectiveness of leveraging structural similarity information for this task.

1. The **Internet** is changing **Africa**.
2. In **South Africa**, **people** can look for **jobs** without leaving **home**.
3. **Movies** from **Nigeria** can easily spread around the **world**.
4. Playing music on mobile **phones** is becoming popular in **Senegal**.
5. **Farmers** in **Tanzania** can learn to grow **vegetables** from **videos**.
6. These **results** show the **power** of the **Internet**.



1. Different **exercise** have different **benefits** for the **body**.
2. **Jogging** can increase your **breathing** and **heart rate**.
3. **Table tennis** keeps you away from **shortsightedness**.
4. Playing **basketball** can strengthen your **muscles**.
5. **Yoga** helps to relieve your **back pain**.
6. So, pick the **one** your **body** needs the most.

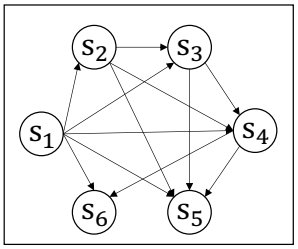


Fig. 4.1 An example of two highly coherent texts exhibiting similar entity connectivity structures. Recognized nouns are highlighted in bold.

4.1 Why Consider the Structural Similarity?

Coherence describes how sentences of a text connect to each other (Reinhart, 1980; Foltz et al., 1998; Schwarz, 2001). Theoretically, documents with similar structures should tend to have a similar degree of coherence. Figure 4.1 illustrates this idea using two texts with entirely different content but quite similar structural patterns. The first text discusses the internet in Africa, beginning with a general overview, followed by an examination of specific African countries, and concluding with a summary. The second text introduces the topic of exercise, then discusses various daily sports, and also ends with a summary. According to linguistic theories of textual coherence, these two texts should exhibit a similar degree of coherence due to their analogous organizational structures. This observation suggests that structural similarity can serve as valuable prior knowledge in coherence assessment. For instance, given the structural alignment between the two texts in Figure 4.1, one could reasonably estimate the coherence of one text by referencing the coherence label of the other.

Although structural similarity between documents holds potential for improving coherence assessment, it has not been explored in previous work. To address this gap, we propose a graph-based approach, which will be described in detail in the following section.

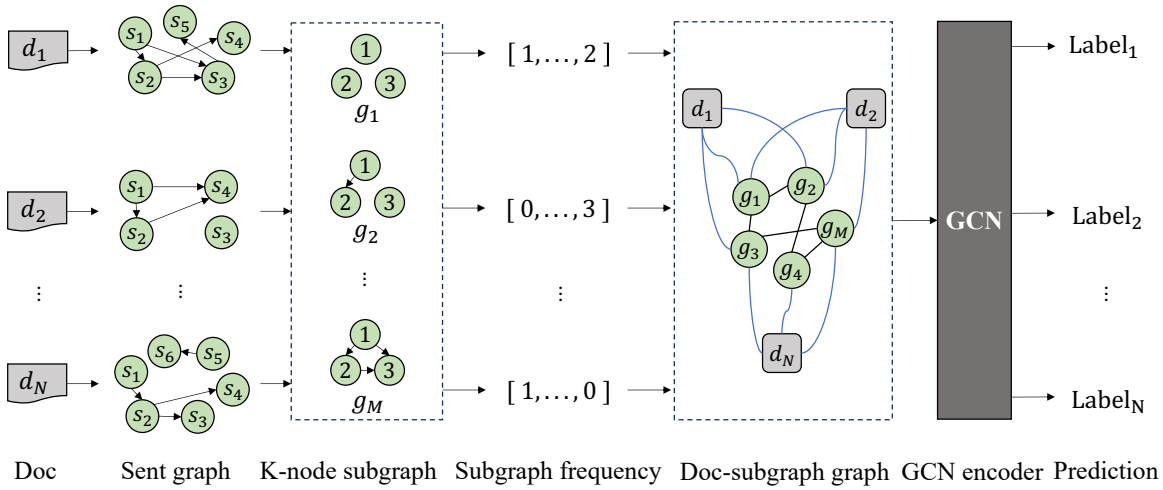


Fig. 4.2 Overview of the proposed graph-based approach. Our method identifies the graph structure of each document, converts the graph into a set of subgraphs, constructs a corpus-level graph based on the shared subgraphs between structurally similar documents, and finally encodes those connections using a Graph Convolutional Network (GCN). For simplicity, we illustrate this process with only three documents and five subgraphs, limiting the number of sentences per document. s_u, d_i, and g_j denote the u-th sentence in a document, the i-th document in the training corpus, and the j-th defined subgraph.

4.2 Graph-based Method

In this section, we present a graph-based approach to modeling the structural similarity between documents for coherence assessment. The main idea is to connect structurally similar documents through a graph and capture their connectivity relationships using Graph Convolutional Networks (GCN). Figure 4.2 provides an overview of our proposed method. We describe step-by-step how to capture the structural similarities between documents, including i) identifying the structure of a document (Section 4.2.1); ii) representing the sentence graph of a document as a set of subgraphs (Section 4.2.2); iii) constructing a corpus-level heterogeneous graph to connect structurally similar documents based on the shared subgraphs (Section 4.2.3); iv) applying a GCN encoder to capture connectivity relationships between document nodes (Section 4.2.4).

4.2.1 Sentence Graph

To model structural similarities between documents, it is first necessary to identify the structural representation of each document. Following the approach of Guinaudeau and Strube (2013), we represent each document as a directed sentence graph, with several modifications

Algorithm 1 Constructing sentence graph

Input: Document d, threshold δ
Output: Sentence graph G

$S, NS \leftarrow$ stanza(d) ▷ Sentences and nouns
$L \leftarrow$ len(S)
$G \leftarrow$ zeros(L, L) ▷ Init adjacency matrix
for $u \leftarrow 1$ **to** $L - 1$ **do**
    for $v \leftarrow u + 1$ **to** L **do**
        $un, vn \leftarrow$ len(NS_u), len(NS_v)
        $sim\_scores \leftarrow [\,]$
        for $a \leftarrow 1$ **to** un **do**
            for $b \leftarrow 1$ **to** vn **do**
                $e_a \leftarrow$ embed($NS_{u,a}$)
                $e_b \leftarrow$ embed($NS_{v,b}$)
                $score \leftarrow$ cos_sim(e_a, e_b)
                Append($score$, sim_scores)
            end for
        end for
        $max\_score \leftarrow$ max(sim_scores)
        if $max\_score > \delta$ **then**
            $G_{u,v} \leftarrow 1$
        end if
    end for
end for

to the original graph construction process. Specifically, in our implementation, two sentences are considered semantically connected if there exists a strong semantic relationship between the nouns in the two sentences. We use nouns instead of entities, as suggested in the original work, because previous studies have shown that nouns are more effective in capturing semantic connections between sentences (Elsner and Charniak, 2011; Tien Nguyen and Joty, 2017).

Given a document, we use the Stanza toolkit (Qi et al., 2020) to segment it into sentences $\{s_1, s_2, \ldots, s_L\}$ and identify all nouns in each sentence. For a pair of sentences s_u and s_v ($u < v$), we compute the similarity score for each pair of nouns from the two sentences (one noun from s_u and the other from s_v) and use the maximum similarity score to measure their semantic connection. The score between two nouns is computed as the cosine similarity between their embeddings. If the maximum similarity score exceeds a preset threshold δ, then the two sentences are considered semantically connected, and we add a directed edge

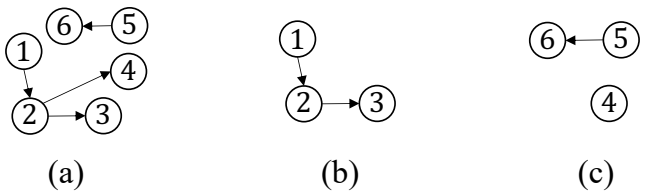


Fig. 4.3 An example of subgraphs, in which graph (b) and graph (c) are 3-node subgraphs of graph (a).

between them (from s_u to s_v). By iterating over all sentence pairs (s_u, s_v) where $u < v$ within the document, we construct a directed graph in which nodes correspond to sentences and edges represent semantic links. The construction process is outlined in Algorithm 1.

4.2.2 Subgraph Set

After constructing the graph representation of each document, we represent each sentence graph as a set of subgraphs. This subgraph set provides an efficient means of comparing the topological structures of sentence graphs (Shervashidze et al., 2009), enabling structural comparison across documents.

Formally, a graph g is considered a subgraph of a graph G if there exists a mapping from the nodes of g to a subset of nodes in G, such that the edge relationships are preserved. When a subgraph contains k nodes, we refer to it as a *k-node subgraph*. In our method, we only consider subgraphs without backward edges. This is because when constructing the sentence graph, we process the document from left to right and never look back. We include both weakly connected and disconnected subgraphs (illustrated in Figure 4.3), as we empirically find they both capture important aspects of document coherence.

Given a sentence graph G_i of a document d_i, we extract all k-node subgraphs by enumerating every possible combination of k nodes and their corresponding edges in G_i. To reduce noise and computational complexity, we discard subgraphs in which the inter-sentence distance between any two nodes exceeds a predefined threshold w, based on the assumption that distant sentences are less likely to be semantically related. Among the remaining subgraphs, structurally identical ones that differ only in node IDs are treated as equivalent, as they are isomorphic in graph theory. We identify such isomorphic subgraphs using the pynauty library and count the frequency of each unique k-node subgraph. Consequently, a sentence graph is represented as a k-node subgraph set. Implementation details are provided in Algorithm 2.

Algorithm 2 Counting Subgraph Frequency

Input: Sentence graph G, subgraph size k, max sentence distance w
Output: subgraph set $freq$

$freq \leftarrow \{\}$ ▷ frequency of each subgraph
$nodes \leftarrow G.nodes()$
$i,\ n \leftarrow 0,\ \text{len}(nodes)$
while $i < (n - k + 1)$ **do**
  $w\_n \leftarrow nodes[i : i + w]$ ▷ distance $< w$
  $k\_node\_combs \leftarrow$ combinations(w_n, k)
  for k_nodes **in** k_node_combs **do**
    $subgraph \leftarrow$ extract(G, k_nodes)
    $signature \leftarrow$ pynauty($subgraph$)
    Add($freq[signature]$, 1)
  end for
  $i \leftarrow i + (w - k + 1)$
end while

4.2.3 Doc-subgraph Graph

A graph is an efficient way to model the correlation between items and has been widely used in various domains, such as knowledge graphs (Carlson et al., 2010) and social networks (Tang and Liu, 2009). We build a corpus-level undirected graph (on the training dataset), named *doc-subgraph graph*, to explicitly connect structurally similar documents through their shared subgraphs (shown in Figure 4.2). The graph contains document nodes and subgraph nodes, and the total number of nodes is the sum of the number of documents (N) and the number of k-node subgraph types (M) mined in Section 4.2.2. We design two types of edges in the graph: (i) edges between documents and subgraphs, and (ii) edges between subgraphs. The first type of edge is added when a document's subgraph set contains a given subgraph, and its weight is computed as the product of the subgraph's normalized frequency within the document and its inverse document frequency in the corpus. The definition of inverse document frequency follows that of TF-IDF, but in this context, it reflects how common a subgraph is across all documents' subgraph sets. The second type of edge is constructed between two subgraphs that co-occur in the same subgraph set of a document, with edge weight corresponding to their co-occurrence probability. We model the co-occurrence information between subgraphs because it has been shown to be useful for comparing similar structures between graphs (Kondor et al., 2009).

Formally, we denote the documents in a training corpus as $\mathbf{D} = \{d_1, d_2, \ldots, d_N\}$ and all types of k-node subgraphs mined from the corpus as $\mathbf{SubG} = \{g_1, g_2, ..., g_M\}$. We use G_i to denote the sentence graph of document d_i and $F_i = \{f_{i1}, f_{i2}, ..., f_{iM}\}$ to denote the k-node

subgraph set mined from G_i, where f_{ij} denotes the frequency of subgraph g_j in G_i. We represent nodes in the doc-subgraph graph as $\mathbf{V} = \{v_1, ..., v_N, v_{N+1}, ..., v_{N+M}\}$, in which $\{v_1, ..., v_N\}$ are documents $\mathbf{D}$ and $\{v_{N+1}, ..., v_{N+M}\}$ are k-node subgraphs $\mathbf{SubG}$.

For any pair of document node v_i $(i \leq N)$ and subgraph node v_{N+j} $(j \leq M)$, we build an edge between them if g_j appears in the subgraph set of d_i, i.e., $f_{ij} > 0$, and define the edge weight as:

$$A_{i,N+j} = \frac{f_{ij}}{\sum_{j'=1}^{M} f_{ij'}} \cdot \log \frac{N}{|\{d \in \mathbf{D} : g_j \in d\}|} \tag{4.1}$$

where the first term is the normalized frequency of subgraph g_j in the subgraph set F_i, and the second term is an inverse document frequency factor, which diminishes the weight of subgraphs that occur frequently in subgraph sets and increases the weight of subgraphs that occur rarely. $|\{d \in \mathbf{D} : g_j \in d\}|$ represents the number of documents whose subgraph set contains subgraph g_j. A denotes the adjacency matrix of the doc-subgraph graph with shape $(N + M) \times (N + M)$ and is initialized as a zero matrix. To make the graph symmetrical, we set the value of $A_{N+j,i}$ to be equal to $A_{i,N+j}$.

We also construct edges between any pair of subgraph nodes v_{N+j} and $v_{N+j'}$ $(j \leq M, j' \leq M, j \neq j')$ if g_j and $g_{j'}$ co-occur in the subgraph set of a document, i.e., $\exists\, d_i \in \mathbf{D} : f_{ij} > 0,\ f_{ij'} > 0$. The weight is defined as the Pointwise Mutual Information (PMI) between these two subgraphs, which is a popular way (Ghazvininejad et al., 2016; Yao et al., 2019) to measure co-occurrence information:

$$A_{N+j,N+j'} = \log \frac{p(j, j')}{p(j)\ p(j')} \tag{4.2}$$

$$\begin{aligned} p(j) &= \frac{|d \in \mathbf{D} : g_j \in d|}{N} \\ p(j, j') &= \frac{|d \in \mathbf{D} : g_j \in d,\ g_{j'} \in d|}{N} \end{aligned} \tag{4.3}$$

The PMI can be positive or negative. Following previous work, we clip negative PMI values at 0 since this strategy works well across many tasks (Kiela and Clark, 2014; Milajevs et al., 2016; Salle and Villavicencio, 2019).

4.2.4 GCN Encoder

We adopt a GCN (Kipf and Welling, 2017) to encode the built doc-subgraph graph. GCN is a graph neural network that directly operates on graph-structured data. By integrating the normalized adjacency matrix, the GCN learns node representations based on both the connectivity patterns and feature attributes of the graph (Li et al., 2018).

Formally, given the built graph with $(N + M)$ nodes, we represent it using an $(N + M) \times (N + M)$ adjacency matrix A. Following Kipf and Welling (2017), we first add self-connections for each node:

$$\tilde{A} = A + I_{N+M} \tag{4.4}$$

where I_{N+M} is an identity matrix. A two-layer GCN is then applied to the graph, with the convolution operation at the l-th layer defined as:

$$H^{(l)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}\mathbf{W}^{(l-1)}\right) \tag{4.5}$$

Here, $\tilde{D}$ denotes the degree matrix (i.e., $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$) and $\mathbf{W}^{(l-1)}$ is a layer-specific trainable weight matrix. σ is an activation function, such as ReLU. $H^{(l)}$ denotes the output of l-th GCN layer; $H^{(0)} = X$, which is a matrix of node features. We use representations from the pre-trained model as features of document nodes due to its excellent performance on document-level tasks (Guo and Nguyen, 2020; Yin et al., 2021; Zhou et al., 2021). For subgraph nodes, since they have no textual contents, we set their features to zero vectors, which is a common setting in heterogeneous graphs (Ji et al., 2021). Finally, we feed the outputs of the two-layer GCN into a softmax classifier:

$$P = \mathrm{softmax}(H^{(2)}) \tag{4.6}$$

and train the model by minimizing the Cross-Entropy loss over document nodes:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} Y_{i,c} \cdot \log\left(P_{i,c}\right) \tag{4.7}$$

where Y_i is the label of document node v_i with a one-hot scheme, C is the number of classes.

While evaluating, for each document in the test corpus, we add it to the doc-subgraph graph, normalize the adjacency matrix of the updated graph, and predict its label, as shown in Algorithm 3.

4.3 Experiments

4.3.1 Datasets

We evaluate the proposed method on two benchmark tasks: assessing discourse coherence (ADC) and automated essay scoring (AES). Detailed descriptions of the datasets used for each task are provided in Section 2.2.1.

Algorithm 3 Evaluation

Input: Test corpus **TC**, Doc-subgraph graph G, Trained GCN
Output: Predictions $preds$

```
preds ← []
N ← len(TC)
for i ← 1 to N do
    d_i ← TC[i]
    G* ← Add(d_i, G)          ▷ Add document
    G* ← Norm(G*)             ▷ Norm graph
    l_i ← GCN(G*)             ▷ Predict label
    Append(l_i, preds)
end for
```

Assessing Discourse Coherence. ADC refers to the task of measuring the coherence of a given text. The benchmark dataset used for this task is the Grammarly Corpus of Discourse Coherence (GCDC) dataset (Lai and Tetreault, 2018). GCDC contains texts from four domains: **Yahoo** online forum posts, emails from Hillary **Clinton**'s office, emails from **Enron**, and **Yelp** online business reviews. Each text is annotated by expert raters with a coherence score in $\{1, 2, 3\}$, indicating low, medium, and high levels of coherence, respectively.

Automated Essay Scoring. AES is the task of assigning scores to essays and has been used to evaluate coherence models (Burstein et al., 2010; Jeon and Strube, 2020b). Following previous work (Jeon and Strube, 2020b), we employ the Test of English as a Foreign Language (TOEFL) dataset (Blanchard et al., 2014) in our experiments. The corpus contains essays written in response to **eight prompts**, with each essay annotated with a score level: low, medium, or high.

4.3.2 Experimental Settings

We implement our method using the PyTorch library. The pre-trained embedding we use to calculate the similarity between nouns is GloVe (Pennington et al., 2014), and we set the similarity threshold δ to 0.65. For the subgraph set construction, we use 4-node subgraphs as basic units for the ADC task and 5-node subgraphs for the AES task, and limit the maximum sentence distance w to 8 for both tasks. A two-layer GCN is employed in our method, with ReLU as the activation function. We follow previous work (Jeon and Strube, 2020b) to use the representation from XLNet_{base} (Yang et al., 2019) as document node features, and initialize XLNet using the pre-trained checkpoint from Huggingface.[1]

[1] `https://huggingface.co/xlnet-base-cased`

For the GCDC dataset, we perform 10-fold cross-validation on the training dataset following previous work (Lai and Tetreault, 2018). The dimensionality of the two-layer GCN is set to 240 for the Clinton and Enron domains, and 360 for the Yahoo and Yelp domains. We use the Adam optimizer with an initial learning rate of 0.01 for Clinton and Enron, and 0.008 for Yahoo and Yelp. For the TOEFL corpus, we conduct 5-fold cross-validation on the dataset for each prompt, which is the standard evaluation setting for the AES task (Taghipour and Ng, 2016). A two-layer GCN with a dimension size of 240 and the Adam optimizer with an initial learning rate of 0.05 is employed for every prompt dataset. A dropout rate of 0.5 is applied to both tasks. We train the model for 160 epochs on the GCDC dataset and 400 epochs on the TOEFL dataset. All experiments are conducted on a single Tesla P40 GPU with 24 GB of memory. Training takes approximately 0.5 days for the GCDC dataset and 1.5 days for the TOEFL dataset.

Baselines. To assess the effectiveness of structural similarities between documents for coherence modeling, we conduct an empirical comparison between our proposed method and a baseline that does not use such information. We refer to this baseline as XLNet+DNN, which inputs document representations from XLNet as features, learns document embeddings with a two-layer deep neural network (DNN), and uses a softmax layer as the classifier. The only difference between the XLNet+DNN baseline and our method in terms of mathematical form is whether the regularized adjacency matrix $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is applied (Li et al., 2018). We configure this baseline to have the same number of parameters as our method for a fair comparison.

We also compare our method with the approach of Mesgar and Strube (2016), which incorporates subgraphs as extra input features. For a fair comparison, we input document representations from XLNet to this model, equip it with a two-layer DNN and a softmax layer for feature extraction and classification. Furthermore, we evaluate our model against existing state-of-the-art methods for each task to validate its effectiveness.

4.3.3 Overall Results

Assessing Discourse Coherence. Table 4.1 presents the experimental results on the GCDC dataset.[2] The first three rows in the upper block of the table report the performance of embedding-based models (Li and Jurafsky, 2017; Mesgar and Strube, 2018; Lai and Tetreault, 2018), while the following four rows (Mesgar and Strube, 2016; Moon et al., 2019; Jeon and Strube, 2020a,b) present results from state-of-the-art models that utilize XLNet. The latter

[2]In Tables 4.1 and 4.2, † indicates that the same XLNet encoder used in our method is employed.

Model	Yahoo	Clinton	Enron	Yelp	Avg
Li and Jurafsky (2017)	53.50	61.00	54.40	49.10	54.50
Lai and Tetreault (2018)	54.90	60.20	53.20	54.40	55.70
Mesgar and Strube (2018)	47.30	57.70	50.60	54.60	52.55
Mesgar and Strube (2016)†	$61.30_{0.84}$	$64.60_{0.89}$	$55.74_{0.90}$	$56.70_{0.78}$	59.59
Moon et al. (2019)†	$56.80_{0.95}$	$60.65_{0.76}$	$54.10_{0.89}$	$55.85_{0.85}$	56.85
Jeon and Strube (2020a)†	$56.75_{0.83}$	$62.15_{0.88}$	$54.60_{0.97}$	$56.45_{0.97}$	57.49
Jeon and Strube (2020b)†	57.30	61.70	54.50	56.90	57.60
XLNet+DNN	$60.70_{1.03}$	$64.00_{1.36}$	$55.15_{1.14}$	$56.45_{0.94}$	59.10
Our Method*	$\mathbf{63.65}_{0.74}$	$\mathbf{66.20}_{0.81}$	$\mathbf{57.00}_{0.81}$	$\mathbf{58.05}_{1.21}$	**61.23**

Table 4.1 Mean accuracy (standard deviation) on GCDC. * indicates that our model significantly outperforms the XLNet+DNN baseline ($p < 0.05$).

group, which leverages a pre-trained transformer as the encoder, substantially outperforms the embedding-based methods, highlighting the effectiveness of contextualized representations.

The performance of the XLNet+DNN baseline and our proposed method is reported in the last two blocks of Table 4.1. As shown, incorporating structural similarity information between documents significantly improves coherence assessment, increasing the average accuracy from 59.10% (XLNet+DNN) to 61.23% with our approach. While using subgraphs as input features (Mesgar and Strube, 2016) also contributes to performance gains, the improvement is comparatively limited. We hypothesize that simply concatenating subgraph features does not effectively capture structural similarities across documents. In contrast, our method explicitly models these similarities by connecting structurally related documents in a graph, thereby making more effective use of this information. Notably, even our simple XLNet+DNN baseline outperforms previous state-of-the-art models built on XLNet. This may be because the GCDC dataset contains mostly short and informal texts, while previous SOTA models are designed to handle long and well-formatted documents. In contrast, our method performs well on the corpus, achieving the best results.

Automated Essay Scoring. As discussed in Section 4.3.1, Automated Essay Scoring (AES) is a task aimed at evaluating the overall quality of essays and has been widely adopted as a benchmark for assessing coherence models. To better illustrate the effectiveness of our approach, we report the performance of both existing coherence models (Mesgar and Strube, 2018; Moon et al., 2019; Jeon and Strube, 2020a,b) and a representative model specifically designed for the AES task. For the latter, we report the results of Dong et al. (2017), a state-of-the-art AES method.

Model	Prompt								Avg
	1	2	3	4	5	6	7	8	
Dong et al. (2017)	69.30	66.47	65.84	66.38	68.89	64.20	67.11	65.73	66.74
Mesgar and Strube (2016)†	$75.31_{0.77}$	$74.90_{0.94}$	$73.42_{0.81}$	$74.35_{1.18}$	$76.10_{0.74}$	$75.42_{0.68}$	$72.48_{0.83}$	$72.31_{0.65}$	74.29
Moon et al. (2019)†	$73.84_{0.81}$	$72.54_{0.87}$	$72.32_{1.27}$	$73.26_{0.67}$	$75.34_{0.72}$	$74.72_{0.78}$	$71.97_{0.71}$	$72.14_{0.93}$	73.27
Jeon and Strube (2020a)†	$75.10_{0.74}$	$73.35_{0.92}$	$74.75_{0.61}$	$74.18_{1.07}$	$76.38_{0.91}$	$74.30_{1.13}$	$73.61_{0.72}$	$73.44_{1.15}$	74.39
Jeon and Strube (2020b)†	75.60	73.40	75.00	73.50	76.80	75.20	73.50	72.80	74.48
XLNet+DNN	$74.70_{0.88}$	$74.46_{0.97}$	$73.07_{0.92}$	$74.09_{1.04}$	$75.45_{0.83}$	$75.21_{0.94}$	$71.17_{0.76}$	$71.95_{0.81}$	73.84
Our Method*	$\mathbf{75.97}_{1.14}$	$\mathbf{76.25}_{1.07}$	$\mathbf{74.14}_{1.18}$	$\mathbf{75.81}_{0.71}$	$\mathbf{77.01}_{0.94}$	$\mathbf{77.08}_{1.14}$	$\mathbf{73.55}_{0.80}$	$\mathbf{72.91}_{0.66}$	**75.34**

Table 4.2 Mean accuracy (standard deviation) on TOEFL. * indicates that our model significantly outperforms the XLNet+DNN baseline ($p < 0.05$).

Table 4.2 presents the results on the TOEFL dataset. Previous coherence models and the XLNet+DNN baseline significantly outperform the AES model proposed by Dong et al. (2017). Similar to our findings on the GCDC dataset, using subgraphs as input features leads to marginal improvements. However, the XLNet+DNN baseline fails to surpass the performance of existing state-of-the-art coherence models on this dataset. The results are reasonable because those coherence models are not only based on XLNet but also consider the characteristics of long documents. Consistent with our observations on the GCDC dataset, our method, by explicitly modeling structural similarities between documents, outperforms the XLNet+DNN baseline on the TOEFL dataset and achieves state-of-the-art performance.

4.3.4 Performance Analysis

To understand how structural similarity contributes to coherence modeling, we compare our model with the XLNet+DNN baseline in terms of predicted label distribution and document length.

Predicted Label Distribution. Figure 4.4 presents the distribution of predicted essay scores produced by the XLNet+DNN baseline and our proposed model on the TOEFL P1 dataset. The predictions from XLNet+DNN exhibit a strong bias toward the medium score category, with approximately 60% of essays assigned to this group. We speculate this is caused by the uneven label distribution in the TOEFL P1 dataset, where low-, medium-, and high-scoring essays comprise 10.3%, 53.8%, and 35.9% of the data, respectively. In contrast, our model appears less influenced by this uneven distribution, making more low and high score predictions. We also collect the prediction accuracy of the two models for each essay score. The prediction accuracy of the XLNet+DNN model for low, medium, and high scores is 35.29%, 83.71%, and 76.47%, respectively, and that of our method is 50.00%, 82.02%, and

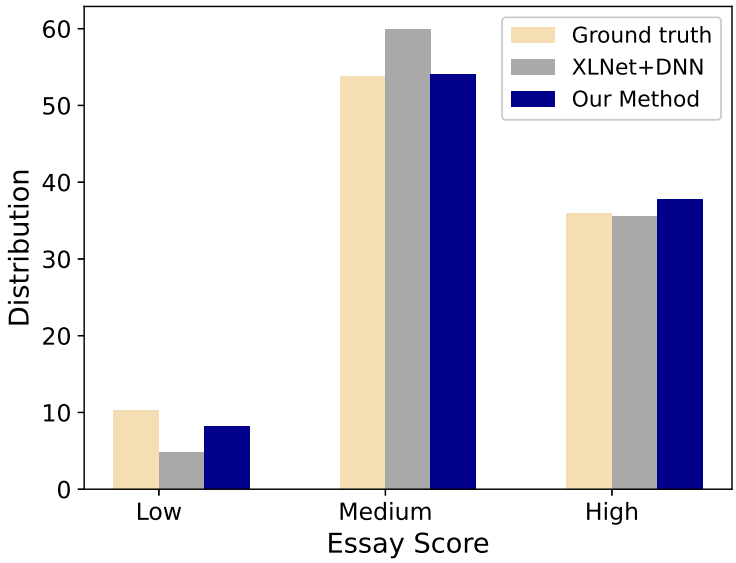


Fig. 4.4 Predicted label distribution in TOEFL P1 dataset.

84.87%, respectively. XLNet+DNN mainly predicts medium scores, so this label's recall value is high. Compared with the baseline, our method makes relatively accurate predictions for all essay scores, suggesting that capturing structural similarities between essays helps mitigate the effects of uneven label distribution and thus focuses on learning coherence patterns.

Document Length. Figure 4.5 shows the accuracy trends of the baseline and our method on the TOEFL P1 dataset as essays become longer. The curve of XLNet+DNN generally shows a downward trend, with accuracy decreasing as the essay's length increases. The results are not surprising, since long documents contain more complicated semantics and are therefore more challenging. Our model performs comparably to the baseline on short documents (length $\leq$ 200). However, as essay length increases, our method maintains relatively high accuracy and even shows a slight improvement in the medium-length range (200 < length $\leq$ 400). These results suggest that structural similarity information can improve the model's robustness when the document length increases.

4.3.5 Ablation Study

To assess the contribution of each type of edge in our method, we conduct an ablation study by selectively removing specific edges from the graph structure. Specifically, we evaluate the performance of our model after removing the edges between subgraph nodes (denoted ESS) and then the edges between the document node and subgraph nodes (denoted EDS). Notably,

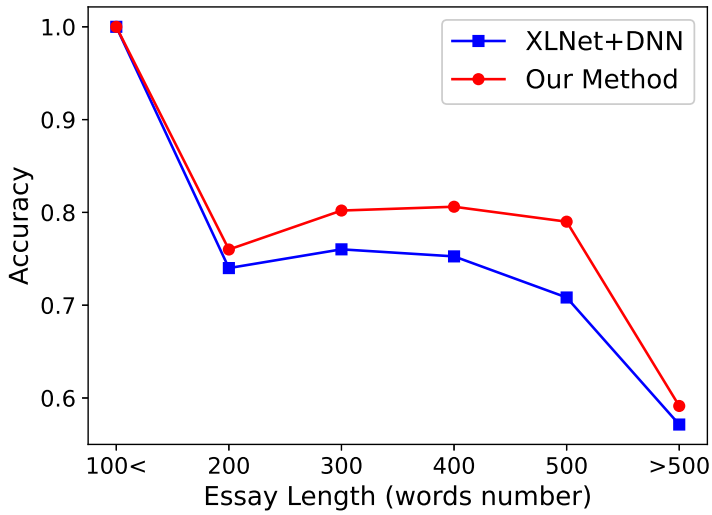


Fig. 4.5 Accuracy against essay length.

when all edges are removed, resulting in each document node being completely isolated, our method degrades to the XLNet+DNN baseline.

Model	GCDC Clinton	TOEFL P1
	Acc	Acc
Our Method	**66.20**	**75.97**
- ESS	66.00	75.42
- ESS, EDS	64.00	74.70

Table 4.3 Ablation study for different edges on the GCDC Clinton and TOEFL P1 dataset.

Table 4.3 presents the experimental results on the GCDC Clinton and TOEFL P1 datasets. It is evident that removing either type of edge negatively impacts model performance. Notably, the performance degradation is more pronounced when edges between the document node and subgraph nodes (EDS) are removed, compared to the removal of edges between subgraph nodes (ESS). This outcome aligns with the intuition that edges between documents and subgraphs are the key to connecting documents with similar structures, while edges between subgraphs are considered to further assist it (Kondor et al., 2009).

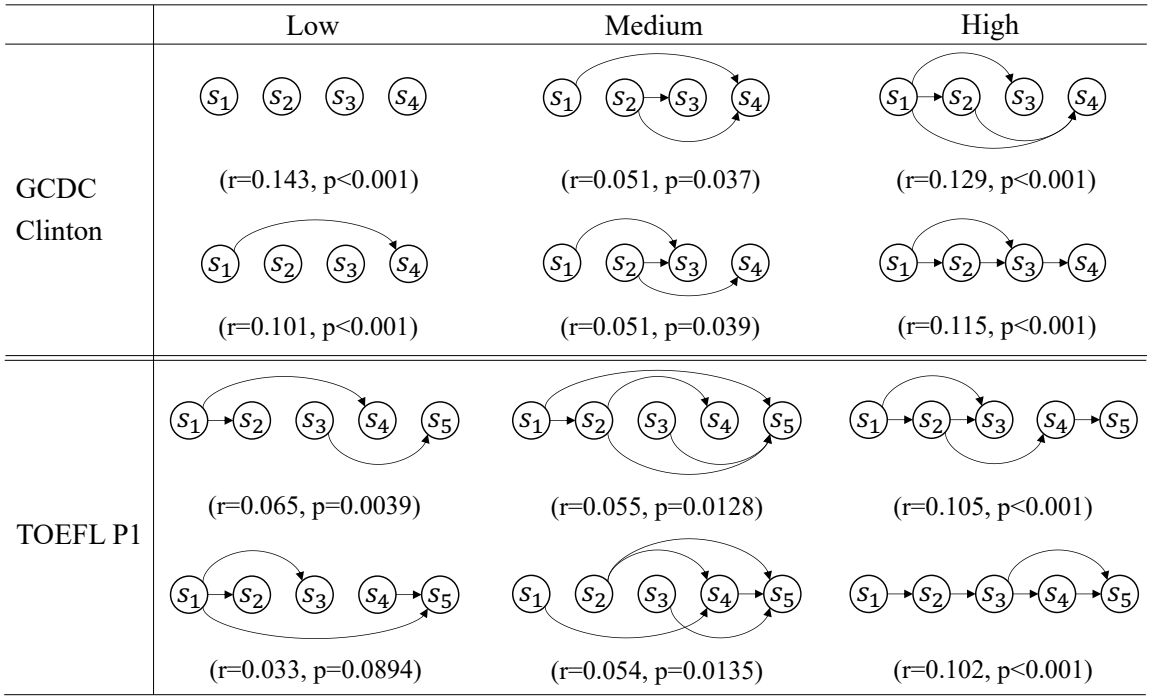


Fig. 4.6 The top two most positively correlated subgraphs for each coherence level on the GCDC Clinton and TOEFL P1. r denotes the correlation coefficient value, and p is the p_value ($p < 0.05$ means statistically significant).

4.3.6 Subgraph Analysis

In this section, we conduct a statistical analysis to identify which subgraphs,[3] representing sentence connection patterns, are most strongly associated with each level of coherence. Specifically, we compute the Pearson correlation coefficient between the frequency of each subgraph and the corresponding coherence label and test the significance of these correlations. Figure 4.6 presents the two most highly correlated subgraphs for both the GCDC Clinton and TOEFL P1 datasets.

Overall, subgraphs that exhibit positive correlations with higher coherence tend to contain more edges. This observation aligns with prior findings (Guinaudeau and Strube, 2013) that coherence correlates with the average out-degree of sentence graphs. Weakly connected subgraphs are more likely to reflect higher coherence than disconnected ones. For instance, in the GCDC Clinton dataset, the two subgraphs most strongly correlated with low coherence contain isolated nodes or disconnected components, whereas nodes in subgraphs associated with high coherence are (weakly) connected. Furthermore, subgraphs with more connections between adjacent sentences seem to be more correlated with high coherence. For example,

[3]We show readable text examples of subgraphs in Appendix A.1.

there is an almost linear subgraph (or contains linear structure) in the high category of both datasets.

We also find that the subgraph results per coherence level on the GCDC Clinton dataset differ from those on the TOEFL P1 dataset. This discrepancy may stem from two factors: first, the datasets comprise texts from different domains, each with distinct writing styles and structures; and second, the annotation processes involved different annotators who may have had varying preferences for text organization styles.

4.4 Summary

In this chapter, we explore the effectiveness of leveraging structural similarity between documents for coherence modeling. We introduce a graph-based approach that connects documents exhibiting similar structural patterns through shared subgraphs and employs a graph convolutional network (GCN) to model these connectivity relationships. Experimental results on two benchmark datasets demonstrate that our method consistently outperforms strong baselines, achieving state-of-the-art performance on both tasks. Furthermore, we present a comprehensive comparison and in-depth analysis, demonstrating that structural similarity information helps alleviate the impact of uneven label distributions in the datasets and improve the model's robustness across documents of varying lengths.

Chapter 5

Annotation-inspired Implicit Discourse Relation Classification

In linguistics, textual coherence can be achieved not only through the continuity of entities but also via discourse relations. Discourse coherence theories posit relations between text spans as a key feature of coherent text. Nevertheless, existing research on coherence modeling has largely overlooked the role of discourse relations. One contributing factor is the limited accuracy of current discourse parsers, particularly in the classification of implicit discourse relations. For example, the PDTB parser employed by Lin et al. (2011) achieves an F1-score of only 25.46 in recognizing top-level implicit discourse relations. Poor parsing results can undermine the reliability of findings, potentially leading to wrong conclusions regarding the contribution of discourse relations to coherence modeling.

In this chapter, we aim to enhance the performance of implicit discourse relation classification, thereby laying a solid foundation for the coherence analysis based on discourse relations presented in subsequent chapters. We begin by briefly introducing the challenges associated with classifying implicit discourse relations. Next, we revisit the annotation process for these relations and describe the key motivation behind our proposed method. We then provide a detailed description of our approach, which is inspired by the human annotation process. Finally, we demonstrate empirically that our method substantially outperforms previous work, achieving an accuracy of 76.23% in top-level relation classification of PDTB 3.0.

5.1 Why Is Implicit Relation Classification Challenging?

Discourse relations, such as *Cause* and *Contrast*, describe the logical relation between two text spans (Pitler et al., 2009). Discourse connectives (e.g., *but*, *as a result*) are words or phrases that signal the presence of a discourse relation (Pitler and Nenkova, 2009). They can be explicit, as in Example (5.1), or implicit, as in Example (5.2):

(5.1) [I refused to pay the cobbler the full $95]$_{\text{Arg1}}$ **because** [he did poor work.]$_{\text{Arg2}}$

(5.2) [They put the treasury secretary back on the board.]$_{\text{Arg1}}$ (**Implicit=However**) [There is doubt that the change would accomplish much.]$_{\text{Arg2}}$

When discourse connectives are explicitly present between arguments, identifying the sense of a discourse relation is relatively straightforward, as there is typically a strong correspondence between specific connectives and particular relation types. For instance, the connective *because* frequently signals a *Cause* relation. Pitler and Nenkova (2009) demonstrate that using only discourse connectives as features, a four-way classification task of explicit discourse relations in PDTB 2.0 can achieve an accuracy of 85.8%. In contrast, classifying implicit discourse relations is challenging, as there are no connective cues present in the text. In such cases, it is necessary to rely on the context or semantics of the two arguments to infer the underlying relation. For instance, in Example (5.2), cues such as "put somebody back", "doubt", and "change" suggest the presence of a *Contrast* relation between the arguments. Existing work has attempted to perform implicit discourse relation classification directly from arguments. These approaches range from designing linguistically informed features from arguments (Lin et al., 2009; Pitler et al., 2009) to modeling interaction between arguments using neural networks (Lei et al., 2017; Guo et al., 2018). Despite their impressive performance, the absence of explicit discourse connectives makes the prediction extremely hard and hinders further improvement (Lin et al., 2014; Qin et al., 2017).

Due to the significant performance gap between explicit and implicit discourse relation classification, some studies have attempted to incorporate implicit connectives into the training of implicit relation classifiers (i.e., connective-enhanced methods). For example, Qin et al. (2017) propose an adversarial model to transfer knowledge from a model trained with access to implicit connectives to one that does not have access to such information. Similarly, Kishimoto et al. (2020) introduce a multi-task learning framework that incorporates implicit connective prediction as an auxiliary training objective. However, these approaches may be suboptimal, as discourse connectives are still absent from the input text.

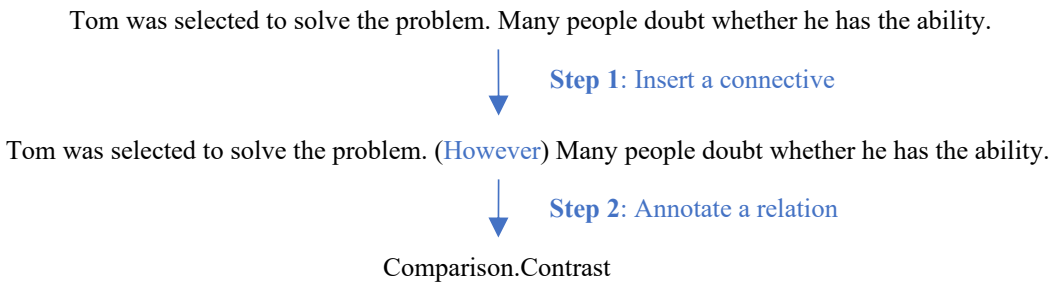


Fig. 5.1 An example illustrating the two-step annotation procedure for implicit discourse relations in the Penn Discourse Treebank (PDTB) 2.0.

5.2 The Annotation Process of Implicit Relations

According to the annotation manual of the Penn Discourse Treebank 2.0 (Prasad et al., 2006), annotators follow a two-step procedure to label implicit discourse relations. Given a pair of arguments, they first insert an appropriate discourse connective between them. Based on the inserted connective and the content of the arguments, they then annotate a discourse relation. Prasad et al. (2006) note that the two-step method facilitates the annotation of implicit discourse relations and improves inter-annotator agreement. Figure 5.1 illustrates an example of this annotation process.

This annotation strategy raises a natural question: can we design a model that mimics this process in order to improve the performance of implicit discourse relation classification?

5.3 An Annotation-inspired Model

Inspired by the PDTB annotation process, we explicitly generate discourse connectives for implicit relation classification. Following previous work (Lin et al., 2009), we use the gold standard arguments and focus on relation prediction. Figure 5.2 shows the overall architecture of our proposed model. It consists of two components: (1) generating a discourse connective between arguments; (2) predicting a discourse relation based on arguments and the generated connective. In this section, we provide a detailed description of each component, discuss the challenges encountered during training, and show our solutions.

Formally, let $X_1 = \{x_1, ..., x_n\}$ and $X_2 = \{x_{n+1}, ..., x_{n+m}\}$ be the two input arguments (Arg1 and Arg2) of implicit relation classification, where x_i denotes the i-th word in Arg1 and x_{n+j} denotes the j-th word in Arg2. We denote the relation between those two arguments as y. Similar to the setup in existing connective-enhanced methods, each training sample

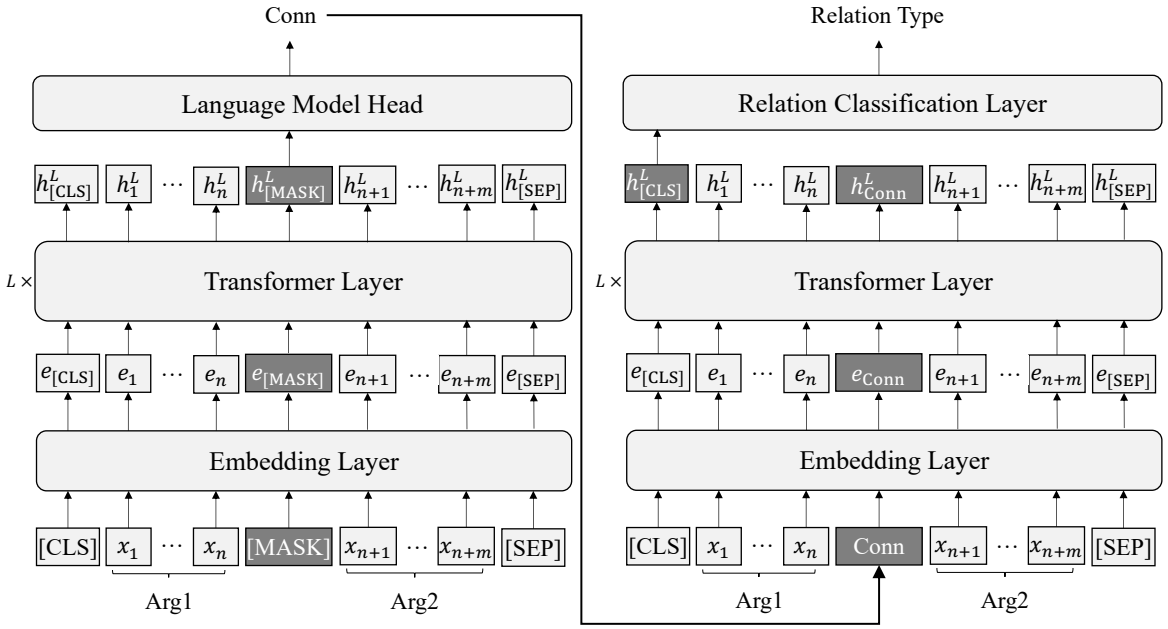


Fig. 5.2 An overview of the proposed approach. The left part is the connective generation module, which generates a connective at the masked position between arguments (Arg1, Arg2). The right part is the relation classification module, which predicts the relation based on both arguments and the generated connective. The two modules share the embedding layer and transformer blocks, and the entire model is trained in an end-to-end manner.

(X_1, X_2, c, y) also includes an annotated implicit connective c that best expresses the relation. During evaluation, only arguments (X_1, X_2) are available to the model.

5.3.1 Connective Generation

Connective generation aims to generate a discourse connective between two arguments (shown in the left part of Figure 5.2). We achieve this by using bidirectional masked language models (Devlin et al., 2019), such as RoBERTa. Specifically, we insert a $[\mathrm{MASK}]$ token between two arguments and generate a connective on the masked position.

Given a pair of arguments Arg1 and Arg2, we first concatenate a $[\mathrm{CLS}]$ token, argument Arg1, a $[\mathrm{MASK}]$ token, argument Arg2, and a $[\mathrm{SEP}]$ token into $\widetilde{X} = \{[\mathrm{CLS}]\ X_1\ [\mathrm{MASK}]\ X_2\ [\mathrm{SEP}]\}$. For each token $\tilde{x}_i$ in $\widetilde{X}$, we convert it into the vector space by adding token, segment, and position embeddings, thus yielding input embeddings $E \in \mathbb{R}^{(n+m+3)\times d}$, where d is the hidden size. Then, we input E into L stacked Transformer blocks, and each Transformer

layer acts as follows:

$$\begin{aligned} G &= \mathrm{LN}(H^{l-1} + \mathrm{MHAttn}(H^{l-1})) \\ H^{l} &= \mathrm{LN}(G + \mathrm{FFN}(G)) \end{aligned} \tag{5.1}$$

where H^l denotes the output of the l-th layer and $H^0 = E$; LN is layer normalization; MHAttn is the multi-head attention mechanism; FFN is a two-layer feed-forward network with ReLU as hidden activation function. To generate a connective on the masked position, we feed the hidden state of the $[\mathrm{MASK}]$ token after L Transformer layers into a language model head (LMHead):

$$\mathbf{p}^c = \mathrm{LMHead}(h^L_{[\mathrm{MASK}]}) \tag{5.2}$$

where $\mathbf{p}^c$ denotes the probabilities over the whole connective vocabulary. However, a normal LMHead can only generate one word without the capacity to generate multi-word connectives, such as "for instance". To overcome this shortcoming, we create several special tokens in LMHead's vocabulary to represent those multi-word connectives, and initialize their embedding with the average embedding of the contained single words. Taking "for instance" as an example, we create a token $[\mathrm{for\_instance}]$ and set its embedding as $\mathrm{Average}(\mathrm{embed}(\text{"for"}), \mathrm{embed}(\text{"instance"}))$.

We choose cross-entropy as the loss function for the connective generation module:

$$\mathcal{L}_{Conn} = -\sum_{i=0}^{N}\sum_{j=0}^{CN} C_{ij}\log(P^c_{ij}) \tag{5.3}$$

where C_i is the annotated implicit connective of the i-th sample represented as a one-hot scheme, CN is total number of connective classes.

5.3.2 Relation Classification

The goal of relation classification is to predict the implicit relation between arguments. Typically, it is solved using only arguments as input (Zhang et al., 2015; Kishimoto et al., 2018). In this work, we propose to predict implicit relations based on both input arguments and the generated connectives (shown in the right part of Figure 5.2).

First, we need to obtain a connective from the connective generation module. A straightforward way is to apply the $\arg\max$ operation on the probabilities output by LMHead, i.e. $\mathrm{Conn} = \arg\max(\mathbf{p}^c)$. However, it is a non-differentiable process, which means the training signal of relation classification cannot be propagated back to adjust the parameters of the connective generation module. Hence, we adopt the Gumbel-Softmax technique (Jang et al., 2017) for the task. The Gumbel-Softmax technique has been shown to be an effective

approximation to the discrete variable (Shi et al., 2021). Therefore, we use

$$\begin{aligned} g &= -\log(-\log(\xi)),\ \xi \sim \mathrm{U}(0,1) \\ \mathbf{c}_i &= \frac{\exp((\log(p_i^c) + g_i)/\tau)}{\sum_j \exp((\log(p_j^c) + g_j)/\tau)} \end{aligned} \tag{5.4}$$

as the approximation of the one-hot vector of the generated connective on the masked position (denoted as Conn in Figure 5.2), where g is the Gumbel distribution, U is the uniform distribution, p_i^c is the probability of i-th connective output by the LMHead, $\tau \in (0, \infty)$ is a temperature parameter.

Once the generated connective, denoted as "Conn", is obtained, we concatenate it with arguments and construct a new input as $\bar{X} = \{[\mathrm{CLS}]\ X_1\ \mathrm{Conn}\ \mathrm{X}_2\ [\mathrm{SEP}]\}$. This new form of input is precisely the same as the input in explicit discourse relation classification. We argue that the key to fully using connectives is to insert them into the input texts instead of treating them merely as a training objective. Like the connective generation module, we feed $\bar{X}$ into an Embedding Layer and L stacked Transformer blocks. Note that we share the Embedding Layer and Transformers between the connective generation and relation classification modules. Doing so can not only reduce the total memory for training the model but also prompt the interaction between the two tasks. Finally, we feed the output of the L-th Transformer at the $[\mathrm{CLS}]$ position to a relation classification layer:

$$\mathbf{p}^r = \mathrm{softmax}(\mathbf{W}_r h^L_{[\mathrm{CLS}]} + \mathbf{b}_r) \tag{5.5}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are learnable parameters. Similarly, we use cross-entropy for training, and the loss is formulated as:

$$\mathcal{L}_{Rel} = -\sum_{i=0}^{N}\sum_{j=0}^{RN} Y_{ij}\log(P^r_{ij}) \tag{5.6}$$

where Y_i is the ground truth relation of the i-th sample with a one-hot scheme, RN is the total number of relations.

5.3.3 Training and Evaluation

To jointly train those two modules, we use a multi-task loss:

$$\mathcal{L} = \mathcal{L}_{Conn} + \mathcal{L}_{Rel} \tag{5.7}$$

A potential issue in this joint training is that poorly generated connectives in the early stages may mislead the relation classifier. One possible solution is always providing manually

Algorithm 4 Scheduled Sampling in Training

Input: relation classifier $\mathbf{RelCls}$, arguments $\mathrm{X}_1, \mathrm{X}_2$, annotated connective $\mathrm{true\_conn}$, generated connective $\mathrm{gene\_conn}$, training step t, hyperparameter in decay k
Output: logits

$\mathrm{p} = \mathrm{random}()$ ▷ $[0.0, 1.0)$
$\epsilon_\mathrm{t} = \frac{\mathrm{k}}{\mathrm{k}+\exp(\mathrm{t}/\mathrm{k})}$
if $\mathrm{p} < \epsilon_\mathrm{t}$ **then**
    $\mathrm{logits} = \mathbf{RelCls}(\mathrm{X}_1, \mathrm{X}_2, \mathrm{true\_conn})$
else
    $\mathrm{logits} = \mathbf{RelCls}(\mathrm{X}_1, \mathrm{X}_2, \mathrm{gene\_conn})$
end if

annotated implicit connectives during training to the relation classifier, similar to Teacher Forcing (Ranzato et al., 2016). However, this might lead to a severe discrepancy between training and inference since manually annotated connectives are not available during inference. We address those issues by introducing Scheduled Sampling (Bengio et al., 2015) into our method. Scheduled Sampling is designed to sample tokens between gold references and model predictions with a scheduled probability in seq2seq models. We incorporate Scheduled Sampling in our training by sampling between the manually annotated and the generated connectives. Specifically, we use the inverse sigmoid decay (Bengio et al., 2015), in which the probability of sampling manually annotated connectives at the t-th training step is calculated as follows:

$$\epsilon_t = \frac{k}{k + \exp(t/k)} \tag{5.8}$$

where $k \geq 1$ is a hyperparameter to control the convergence speed. In the beginning, training is similar to Teacher Forcing due to $\epsilon_t \approx 1$. As the training step t increases, the relation classifier gradually uses more generated connectives, and eventually uses only generated ones (identical to the evaluation setting) when $\epsilon_t \approx 0$. We show the sampling process during training in Algorithm 4.

5.4 Experiments

We carry out a set of experiments to investigate the effectiveness of our method across different corpora and dataset splits. In addition, we perform in-depth analyses to demonstrate that our model learns a better balance between using connectives and arguments than baselines.

Comparison.Concession	Comparison.Contrast
Contingency.Cause	Expansion.Conjunction
Expansion.Equivalence	Expansion.Instantiation
Expansion.Level-of-detail	Temporal.Asynchronous

Table 5.1 Second-level (L2) relations of PCC used in our experiments.

5.4.1 Experimental Settings

Datasets. We evaluate our model on two English corpora, PDTB 2.0 (Prasad et al., 2008) and PDTB 3.0 (Webber et al., 2019b), as well as a German corpus, the Potsdam Commentary Corpus (Bourgonje and Stede, 2020). In the PDTB corpora, discourse relations are annotated using a three-level sense hierarchy. Following prior work (Ji and Eisenstein, 2015; Kim et al., 2020), we perform both top-level 4-way and second-level 11-way classification for PDTB 2.0, and top-level 4-way and second-level 14-way classification for PDTB 3.0. For dataset splitting, we adopt two widely used settings for both PDTB 2.0 and PDTB 3.0. The first, introduced by Ji and Eisenstein (2015), uses sections 2–20 for training, sections 0–1 for development, and sections 21–22 for testing. The second, known as section-level cross-validation (Kim et al., 2020), divides 25 sections into 12 folds, with each fold comprising 21 training sections, 2 validation sections, and 2 test sections. Although PDTB contains over one hundred distinct connectives (e.g., 102 in PDTB 2.0), many of them appear infrequently (e.g., the connective *next* occurs only 7 times in PDTB 2.0). To reduce the complexity of connective generation and ensure sufficient training data for each connective, we limit our experiments to those that occur at least 100 times in the dataset.

The Potsdam Commentary Corpus (PCC) is a German corpus constructed following the annotation guideline of PDTB (Bourgonje and Stede, 2020). In this dataset, relations are also organized in a three-level hierarchy structure. However, this corpus is relatively small, containing only 905 instances of implicit discourse relations, and exhibits a highly imbalanced distribution of relation types, particularly at the top level. For example, the "Expansion" (540) and "Contingency" (246) account for more than 86% of the data among all top-level relations. Bourgonje (2021) concludes that two of four relations are never predicted in his classifier due to the highly uneven distribution of the top-level relation data. Therefore, we only use the second-level relations in our experiments. Furthermore, we use a similar setup to PDTBs for PCC, considering only relations whose frequency is not too low (over 10 in our setting). The final PCC used in our experiments contains 891 isolated data points

covering 8 relations (shown in Table 5.1). As for connectives, we consider only those with a frequency of at least 5, due to the limited size of this corpus.

Implementation Details. We implement our model using the PyTorch library. The bidirectional masked language model employed in this work is $\mathrm{RoBERTa}_{\mathrm{base}}$, initialized with the pre-trained checkpoint provided by Huggingface. For hyperparameter settings, we primarily follow the configuration used in the original RoBERTa model (Liu et al., 2019). Specifically, we use the AdamW optimizer with an initial learning rate of 1e-5, a batch size of 16, and train for a maximum of 10 epochs. Given the variability in training outcomes on the PDTB datasets, we report the average performance over five random runs for the "Ji" data splits, as well as for section-level cross-validation (Xval), following the protocol of Kim et al. (2020). For the PCC corpus, due to its smaller size, we perform 5-fold cross-validation. Model performance is evaluated using standard metrics: accuracy (Acc, %) and macro-averaged F1 score (F1, %).

Baselines. To demonstrate the effectiveness of our model, we compare it against state-of-the-art connective-enhanced methods and several variants of our model:

- **RoBERTa**. This baseline fine-tunes $\mathrm{RoBERTa}_{\mathrm{base}}$ for implicit discourse relation classification, using only the argument pair (Arg1, Arg2) as input. No discourse connective information is used during training.
- **RoBERTaConn**. A variant of the RoBERTa baseline that incorporates gold (annotated) connectives during training. Specifically, the input to the model is (Arg1, true_conn, Arg2). During inference, however, only the arguments (Arg1, Arg2) are provided.
- **Adversarial**. An adversarial-based connective-enhanced method (Qin et al., 2017), in which an implicit relation network is driven to learn from another neural network with access to connectives. We replace its encoder with $\mathrm{RoBERTa}_{\mathrm{base}}$ for a fair comparison.
- **Multi-Task**. A multi-task framework for implicit relation classification (Kishimoto et al., 2020), in which connective prediction is introduced as another training task. We equip it with the same $\mathrm{RoBERTa}_{\mathrm{base}}$ as our method.
- **Pipeline**. A pipeline variant of our method, in which we first train a connective generation model, then train a relation classifier with arguments and the generated connectives. Note that these two modules are trained separately.

Furthermore, we compare our model against previously reported state-of-the-art results on each corpus to provide a comprehensive evaluation. More detailed descriptions of the datasets and baselines are provided in Appendix B.1.

	Level-1 4-way				Level-2 11-way			
	Ji		Xval		Ji		Xval	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Liu et al. (2020b)	$69.06_{0.43}$	$63.39_{0.56}$	-	-	$58.13_{0.67}$	-	-	-
Kim et al. (2020)	66.30	56.00	-	-	$54.73_{0.79}$	-	$52.98_{0.29}$	-
Wu et al. (2022)	71.18	63.73	-	-	60.33	40.49	-	-
Zhou et al. (2022)	70.84	64.95	-	-	60.54	41.55	-	-
Long and Webber (2022)	72.18	**69.60**	-	-	61.69	**49.66**	-	-
RoBERTa	$68.61_{0.73}$	$60.89_{0.19}$	$68.66_{1.29}$	$60.49_{1.86}$	$58.84_{0.48}$	$39.31_{0.83}$	$55.40_{1.65}$	$36.51_{2.75}$
RoBERTaConn	$55.34_{0.39}$	$37.47_{2.27}$	$54.28_{2.12}$	$34.71_{2.75}$	$31.97_{2.75}$	$17.10_{2.81}$	$32.12_{2.63}$	$17.91_{2.12}$
Adversarial	$69.43_{0.70}$	$62.44_{0.61}$	$69.13_{1.14}$	$60.63_{1.47}$	$57.63_{1.10}$	$38.81_{2.25}$	$54.43_{1.79}$	$36.79_{2.24}$
Multi-Task	$70.82_{0.72}$	$63.79_{0.82}$	$70.02_{1.40}$	$62.19_{1.84}$	$60.21_{0.94}$	$39.75_{0.70}$	$56.85_{1.13}$	$36.83_{2.42}$
Pipeline	$71.01_{0.89}$	$64.65_{1.03}$	$69.12_{1.03}$	$61.65_{0.89}$	$59.42_{0.54}$	$40.84_{0.39}$	$55.24_{1.72}$	$37.03_{2.83}$
Our Model	$\mathbf{74.59}_{0.44}$	$68.64_{0.67}$	$\mathbf{71.33}_{1.25}$	$\mathbf{63.84}_{1.96}$	$\mathbf{62.75}_{0.59}$	$42.36_{0.38}$	$\mathbf{57.98}_{1.22}$	$\mathbf{39.05}_{3.53}$

Table 5.2 Results on PDTB **2.0**. Subscripts are the standard deviation of the mean performance.

5.4.2 Overall Results

PDTB 2.0. Table 5.2 shows the experimental results on PDTB 2.0. RoBERTaConn performs much worse than the RoBERTa baseline on this corpus, indicating that simply feeding annotated connectives to the model causes a severe discrepancy between training and evaluation. This finding aligns with the observations of Sporleder and Lascarides (2008b), who showed that models trained on explicitly marked relations often generalize poorly to implicit relation identification. Models enhanced with discourse connective information, namely Adversarial, Multi-Task, Pipeline, and Our Model, consistently outperform the RoBERTa baseline. This demonstrates the effectiveness of leveraging annotated connective information during training for improving implicit discourse relation classification. However, the performance improvements of the Adversarial and Multi-Task models over the RoBERTa baseline are relatively limited and unstable. We attribute this to the fact that these methods incorporate connectives as auxiliary training objectives rather than as explicit input features, thereby limiting their contributions to implicit relation classification. The Pipeline variant also yields limited gains over the RoBERTa baseline. We speculate that this is due to its sequential, non-joint training procedure, i.e., connective generation followed by relation classification, which may lead to error propagation from the first stage to the second, as previously discussed in Qin et al. (2017). Compared to the above connective-enhanced models, our method shows a greater improvement over the RoBERTa baseline, which suggests that our approach is more efficient in utilizing connectives. To further show the efficiency of our approach, we compare

	Level-1 4-way				Level-2 14-way			
	Ji		Xval		Ji		Xval	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Kim et al. (2020)	71.30	64.80	-	-	-	-	$60.78_{0.24}$	-
Xiang et al. (2022)	74.36	69.91	-	-	-	-	-	-
Long and Webber (2022)	75.31	70.05	-	-	64.68	**57.62**	-	-
RoBERTa	$73.51_{0.69}$	$67.98_{0.97}$	$73.42_{0.90}$	$67.54_{1.40}$	$63.32_{0.40}$	$52.49_{1.26}$	$62.65_{1.32}$	$53.19_{1.20}$
RoBERTaConn	$51.74_{0.76}$	$41.45_{0.69}$	$53.90_{1.71}$	$39.39_{2.74}$	$33.67_{1.78}$	$25.40_{2.11}$	$36.68_{2.39}$	$28.18_{4.11}$
Adversarial	$73.83_{0.28}$	$68.60_{0.75}$	$73.30_{1.32}$	$67.23_{1.85}$	$63.00_{0.48}$	$54.28_{1.76}$	$62.12_{1.46}$	$53.85_{1.46}$
Multi-Task	$74.97_{0.70}$	$69.67_{0.76}$	$73.83_{0.94}$	$68.04_{1.30}$	$64.52_{0.31}$	$53.12_{0.63}$	$62.81_{1.36}$	$53.07_{1.40}$
Pipeline	$74.54_{0.22}$	$69.19_{0.60}$	$73.70_{0.89}$	$68.31_{1.78}$	$63.98_{0.63}$	$52.95_{0.48}$	$63.07_{1.70}$	$53.43_{1.63}$
Our Model	$\mathbf{76.23}_{0.19}$	$\mathbf{71.15}_{0.47}$	$\mathbf{75.41}_{0.89}$	$\mathbf{70.06}_{1.72}$	$\mathbf{65.51}_{0.41}$	$54.92_{0.81}$	$\mathbf{64.59}_{1.21}$	$\mathbf{55.26}_{1.32}$

Table 5.3 Results on PDTB **3.0**.

	Level-2 8-way	
	Xval	
Models	Acc	F1
RoBERTa	$35.80_{1.13}$	$15.08_{0.97}$
RoBERTaConn	$30.30_{2.86}$	$12.62_{2.06}$
Adversarial	$35.02_{3.18}$	$18.48_{1.51}$
Multi-Task	$40.48_{1.47}$	$21.22_{2.01}$
Pipeline	$42.97_{3.48}$	$22.66_{1.20}$
Our Model	$\mathbf{44.54}_{3.06}$	$\mathbf{26.93}_{2.06}$

Table 5.4 Results on **PCC**.

it against previous state-of-the-art models on PDTB 2.0 (Liu et al., 2020b; Kim et al., 2020; Wu et al., 2022; Zhou et al., 2022; Long and Webber, 2022). These results are summarized in the first block of Table 5.2. Our model outperforms most existing methods, particularly in terms of accuracy, and achieves the best overall performance on this dataset. The only exception is the F1 score, where our model lags behind that of Long and Webber (2022), especially for second-level (Level-2) classification. This discrepancy can be attributed to our model's inability to predict certain fine-grained discourse relations (see Section 5.4.4), such as Comparison.Concession, which negatively impacts the macro-averaged F1 score.

PDTB 3.0 / PCC. Results on PDTB 3.0 and PCC are shown in Tables 5.3 and 5.4. Similar to the results on the PDTB 2.0, simply feeding connectives for training (RoBERTaConn) hurts performance, especially on the Level-2 classification of PDTB 3.0. Although the Adversarial and Multi-Task models outperform the RoBERTa baseline, the gains are relatively

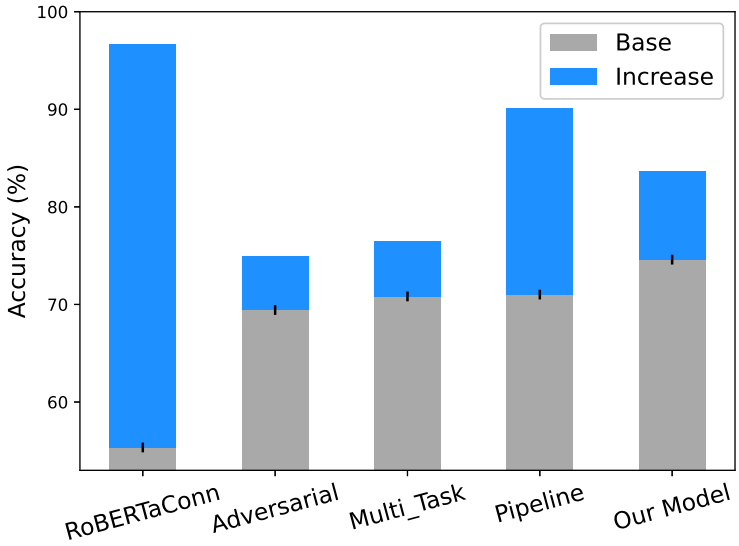


Fig. 5.3 Level-1 classification results on PDTB 2.0 (Ji split) when annotated connectives are fed to connective-enhanced models. "Increase" denotes performance gain compared to the model with default settings ("Base").

modest. Interestingly, despite being affected by cascading errors, the Pipeline variant achieves comparable or even superior results to Adversarial and Multi-Task on both datasets. This suggests the advantage of using connectives as explicit input features, rather than merely as auxiliary training targets, particularly in the case of the PCC corpus. Consistent with the results on PDTB 2.0, our method outperforms Adversarial, Multi-task, and Pipeline on both datasets, demonstrating the superiority of inputting connectives to the relation classifier in an end-to-end manner. It also shows that the method generalizes well across different languages. We further compare our method with three existing SOTA models on PDTB 3.0, Kim et al. (2020), Xiang et al. (2022), and Long and Webber (2022). Results in Table 5.3 show that our approach performs better than these three models.

5.4.3 Performance Analysis

To better understand the effectiveness of our model, we conduct a series of analyses aimed at addressing the following two questions: (1) Does it really benefit from discourse connectives? (2) Can it still make correct predictions when connectives are missing? Additionally, we examine the performance of different models on relation classification when connectives are correctly and incorrectly generated (or predicted).

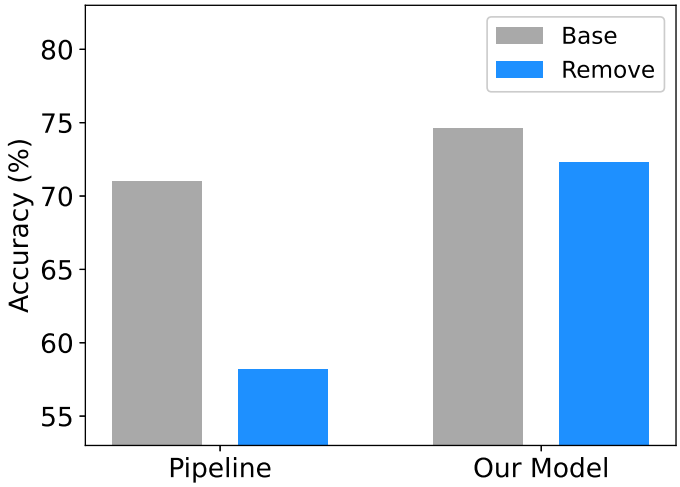


Fig. 5.4 Level-1 classification results on PDTB 2.0 (Ji split). "Remove" denotes the generated connectives are removed from the original model ("Base").

We perform the first analysis by replacing the generated connectives in our model with manually annotated ones,[1] and compare the model's performance before and after this setup. Intuitively, if our model benefits from discourse connectives, both accuracy and macro-averaged F1 score should improve under this setup. For comparison, we apply the same procedure to other connective-enhanced models. We conduct experiments on the Level-1 classification task of PDTB 2.0 using the Ji split, with accuracy results presented in Figure 5.3. As expected, our model shows a substantial performance improvement when provided with gold connectives, confirming that it effectively learns to utilize them for implicit relation classification. While other connective-enhanced models also benefit from gold connectives, the degree of improvement varies. Notably, models that incorporate connectives as part of the input during training (RoBERTaConn, Pipeline, and Our Model) exhibit greater performance gains and higher upper bounds than those that treat connectives solely as auxiliary training objectives (Adversarial and Multi-Task). These findings support our hypothesis that directly incorporating connectives into the model's input is a more effective strategy for enabling the model to utilize them. However, this approach introduces a potential drawback: models may become overly dependent on connectives. For example, RoBERTaConn achieves 96.69% accuracy when gold connectives are available, but its performance drops sharply to 55.34% in their absence, highlighting its over-reliance on connective information.

To examine whether our model suffers from over-reliance on discourse connectives, we perform the second analysis by removing the generated connectives in our model and

[1] In both PDTB 2.0 and 3.0, each instance includes an annotated implicit connective, enabling this analysis.

Models	Correct Group	Incorrect Group
$\text{Base}_{\text{Multi-Task}}$	83.67	59.82
Multi-Task	90.60(+6.93)	59.88(+0.06)
$\text{Base}_{\text{Pipeline}}$	78.87	61.46
Pipeline	89.29(+10.4)	59.81(-1.64)
$\text{Base}_{\text{Our Model}}$	80.28	60.56
Our Model	94.04(+13.8)	62.22(+1.66)

Table 5.5 Level-1 classification results on PDTB 2.0 (Ji split) when connectives are correctly and incorrectly generated (or predicted). "+" and "-" denote the increase and decrease compared to the RoBERTa baseline (Base).

observing changes in its performance. For comparison, we apply the same setting to the Pipeline model. Figure 5.4 presents the Level-1 classification results on PDTB 2.0 (Ji split). Although both models experience a drop in performance, they still outperform RoBERTaConn. This can be attributed to the fact that both models were trained using generated (rather than manually annotated) connectives, which mitigates their dependence on connective information. Notably, our model shows a relatively small performance decrease (from 74.59% to 72.27%), while Pipeline exhibits a more substantial decline (from 71.01% to 58.15%). We hypothesize that this difference arises from the end-to-end nature of our training framework, which enables the model to learn a better balance between argument content and connective cues for relation classification. In contrast, the Pipeline model, with its separately trained connective generation and relation classification components, fails to achieve this balance effectively.

Finally, Table 5.5 presents the performance of the relation classifiers in Multi-Task, Pipeline, and Our Model[2] on PDTB 2.0, evaluated under two conditions: when the connectives are correctly and incorrectly generated or predicted. It is important to note that the results in the "correct" and "incorrect" groups are not directly comparable across models, as each model produces different connective predictions. To address this, we report the performance gain of each model relative to the RoBERTa baseline and compare them from this perspective. When connectives are correctly generated, both Pipeline and our method show an improvement of over 10% in accuracy compared to the RoBERTa baseline, whereas Multi-Task achieves a smaller gain of 6.9%. This suggests that Pipeline and our method make more effective use of connective information than Multi-Task. Conversely, when connectives are incorrectly generated, the Pipeline model performs 1.64% worse than the baseline. By

[2]This analysis excludes models such as Adversarial, which do not generate or predict connectives.

Labels	RoBERTa	Adversarial	Multi-Task	Pipeline	Our Model
Temporal.Asynchronous	54.62	55.01	58.37	55.69	**59.48**
Temporal.Synchrony	00.00	**06.03**	00.00	04.00	00.00
Contingency.Cause	60.03	59.00	64.24	65.40	**66.35**
Contingency.Pragmatic cause	00.00	**05.00**	00.00	00.00	00.00
Comparison.Contrast	60.44	58.20	61.73	60.78	**65.75**
Comparison.Concession	00.00	01.14	00.00	**01.82**	00.00
Expansion.Conjunction	56.03	53.26	**58.94**	54.79	57.04
Expansion.Instantiation	74.07	72.85	**74.12**	70.76	73.87
Expansion.Restatement	57.87	56.94	59.68	57.75	**60.94**
Expansion.Alternative	49.06	44.76	**54.82**	43.96	51.13
Expansion.List	18.07	11.68	11.43	**29.96**	25.47

Table 5.6 F1 results for each second-level relation of PDTB 2.0.

contrast, both Multi-Task and our method maintain performance levels comparable to the baseline, showing good robustness when exposed to incorrect connectives. Although our method consistently outperforms the baseline in both scenarios, its performance drops considerably in the incorrect connective group compared to the correct one. This indicates that its major performance bottleneck originates from the incorrectly generated connectives. A possible solution to this bottleneck is to first pre-train our model on a large explicit connectives corpus, like Sileo et al. (2019). By doing so, the connective generation module may generate more correct connectives, thus improving classification performance, which we leave for future work.

5.4.4 Relation Analysis

We examine which discourse relations benefit most from the joint training of connective generation and relation classification, and compare the results with those of other baselines. Table 5.6 presents the F1-scores for each second-level sense in PDTB 2.0 (Ji split) across different models. Generally, incorporating connectives improves the prediction performance for most relation types, especially in the Multi-Task, Pipeline, and Our Model. For instance, these three models outperform the RoBERTa baseline by more than 4% in F1-score on the *Contingency.Cause* relation.

However, for certain relations such as *Expansion.Instantiation*, the connective-enhanced models exhibit mixed results, with some showing improvement while others experience declines. Notably, all models fail to accurately predict relations such as *Temporal.Synchrony*,

	PDTB 2.0		PDTB 3.0	
Models	Acc	F1	Acc	F1
Our Model	**74.59**	**68.64**	**76.23**	**71.15**
- SS	73.42	66.68	75.87	70.68
- SS, $\mathcal{L}_{Conn}$	70.63	63.43	74.58	69.17
RoBERTa	68.61	60.89	73.51	67.98

Table 5.7 Ablation study for **S**cheduled **S**ampling and connective generation loss $\mathcal{L}_{Conn}$.

Contingency.Pragmatic cause, and *Comparison.Concession*, despite being trained with manually annotated connectives. We hypothesize that this limitation stems from the small number of training instances for these relations, causing models to predict more frequent labels. A feasible solution to this issue is Contrastive Learning (Chen et al., 2020), which has been shown to improve the predictive performance of these three relations (Long and Webber, 2022). We leave the integration of Contrastive Learning with our method to future work.

5.4.5 Ablation Study

We conduct ablation studies to assess the effectiveness of two key components in our framework: Scheduled Sampling (SS) and the connective generation loss, $\mathcal{L}_{Conn}$. To this end, we test the performance of our method by first removing the Scheduled Sampling and then omitting the connective generation loss $\mathcal{L}_{Conn}$. It is important to note that removing $\mathcal{L}_{Conn}$ means that our whole model is trained with only gradients from $\mathcal{L}_{Rel}$.

Table 5.7 presents the Level-1 classification results on PDTB 2.0 and PDTB 3.0 (Ji split). The results show that removing either Scheduled Sampling or the connective generation loss $\mathcal{L}_{Conn}$ leads to a noticeable drop in performance, highlighting the importance of both components for achieving strong results. Interestingly, even when the model is trained solely with the relation classification loss $\mathcal{L}_{Rel}$, it still significantly outperforms the RoBERTa baseline. This suggests that the performance improvements of our full model stem not only from supervision provided by manually annotated connectives but also from the well-designed structure inspired by PDTB's annotation (i.e., the connective generation module and relation prediction module).

5.5 Summary

In this chapter, we propose a novel connective-enhanced method for implicit discourse relation classification, inspired by the annotation framework of the Penn Discourse Treebank (PDTB). We introduce several key techniques to enable effective end-to-end training of our model. Experimental results on three benchmark datasets demonstrate that our approach consistently outperforms a range of baseline models. Further analyses of model behavior reveal that our approach can learn a good balance between using arguments and connectives for implicit discourse relation prediction.

Chapter 6

Explicit to Implicit Discourse Relation Classification

In the previous chapter, we mentioned that discourse relations can either be signaled explicitly with connectives, as in Example (6.1), or expressed implicitly, as in Example (6.2):

(6.1) [The city had expected to pay about 11 million yen]$_{\text{Arg1}}$ **but** [Fujitsu essentially offered to do it for free.]$_{\text{Arg2}}$ — Contingency.Cause

(6.2) [He has not changed, but those around him have.]$_{\text{Arg1}}$ [Many of his view on the protection of wilderness areas are now embraced by mainstream.]$_{\text{Arg2}}$ — Contingency.Cause

Implicit discourse relations present a significant challenge not only for classification but also for annotation, as annotators must infer the relation based solely on the content of the arguments. In contrast, explicit discourse relations are relatively easier to annotate due to the strong association between discourse connectives and relation types. This distinction has prompted many early studies (Marcu and Echihabi, 2002; Lapata and Lascarides, 2004; Sporleder and Lascarides, 2005; Saito et al., 2006) to use explicit examples to classify implicit relations (dubbed **explicit to implicit relation recognition**). The main idea is to construct an *implicit-like* corpus by removing connectives from explicit instances and use it to train a classifier for implicit relation recognition. While this method attains good results on test sets constructed in the same manner, it is reported by Sporleder and Lascarides (2008b) to perform poorly in real implicit scenarios. They claim this is caused by the linguistic dissimilarities between explicit and implicit examples, but provide no corpus-level empirical evidence. More recent works (Huang and Li, 2019; Kurfalı and Östling, 2021) focus on enhancing transfer performance from explicit to implicit discourse relations. However, little attention has been paid to the underlying causes of these poor results.

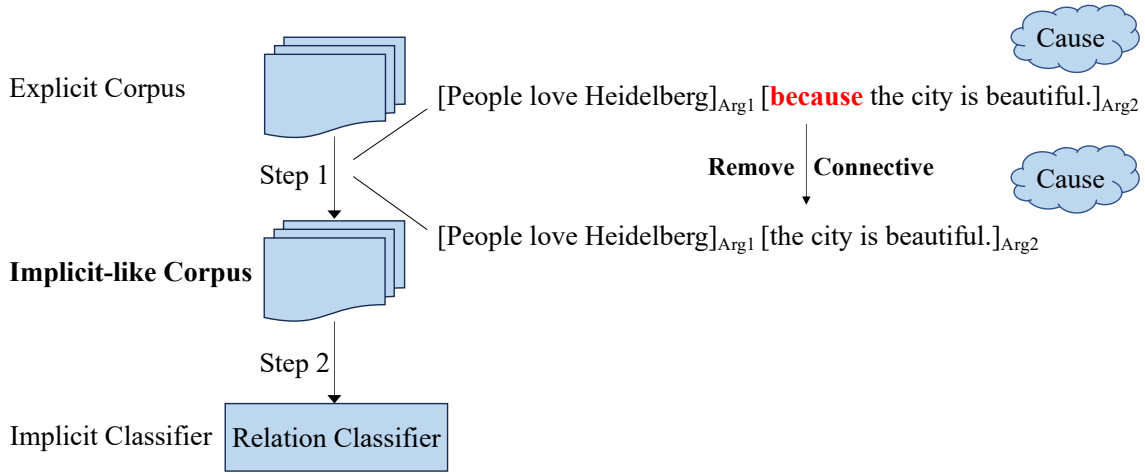


Fig. 6.1 An illustration of the process for training an implicit discourse relation classifier using explicit relation examples.

In this chapter, we show that one cause of the poor transfer performance in *explicit to implicit discourse relation classification* is the presence of a label shift in the construction of the *implicit-like* corpus. We begin by formally defining the task, introducing the benchmark dataset, and highlighting the degraded performance of implicit discourse relation classification under this transfer setting compared to the standard setting, where models are both trained and evaluated on real implicit examples. Next, we define the concept of label shift and provide both manual analysis and empirical evidence to demonstrate its presence. We then analyze why label shift happens in the *implicit-like* corpus by considering factors such as the syntactic role played by connectives, the ambiguity of connectives, and more. Finally, we propose and evaluate two strategies, one data-centric and the other model-centric, to mitigate the impact of label shift, and present experimental results that confirm the effectiveness of our proposed solutions.

6.1 Background

6.1.1 Task

The task of *explicit to implicit relation classification* aims to build an implicit classifier based on explicit examples. The traditional setup for this task is to construct an *implicit-like* corpus by excluding connectives from explicit examples, and then train a classifier on this corpus with the original explicit relations as ground-truth labels, as illustrated in Figure 6.1.

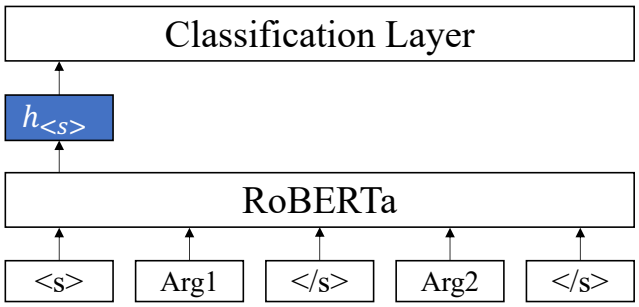


Fig. 6.2 The RoBERTa classifier used in our analyses.

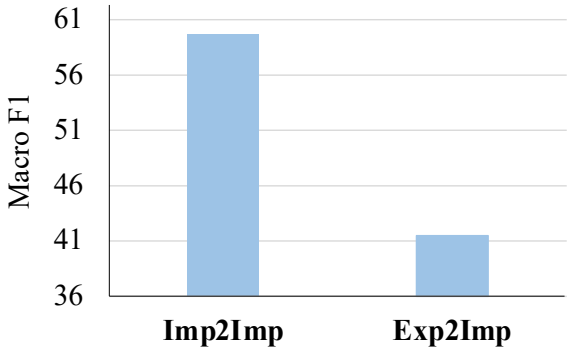


Fig. 6.3 Performance comparison of implicit discourse relation classification between the classifier trained on explicit examples (Exp2Imp) and one trained on real implicit examples (Imp2Imp), using the PDTB 2.0 dataset.

6.1.2 Datasets

The most commonly used datasets for *explicit to implicit discourse relation classification* are the Penn Discourse Treebank (PDTB) versions 2.0 and 3.0 (Ji et al., 2015; Huang and Li, 2019; Kurfalı and Östling, 2021). These corpora (Prasad et al., 2008; Webber et al., 2019b) are annotated using a lexicalized framework that categorizes discourse relations into several types, including the two central to this study: explicit and implicit relations. This clear grouping makes the PDTBs particularly well-suited for *explicit to implicit relation classification*, as it eliminates the need to manually distinguish between explicit and implicit instances (Huang and Li, 2019; Kurfalı and Östling, 2021). Furthermore, both versions provide manually annotated connectives for implicit examples, which is especially valuable for our comparative analysis between explicit and implicit discourse relations.

6.1.3 The Performance Gap

Although *explicit to implicit relation classification* offers a cost-effective and seemingly ideal approach for training implicit discourse relation classifiers, its practical performance is poor (Ji et al., 2015). To examine this issue, we train two classifiers with identical architectures, each consisting of a RoBERTa encoder followed by a linear classification layer (see Figure 6.2). One classifier is trained on the *implicit-like* corpus constructed from explicit instances in PDTB 2.0, while the other is trained directly on real implicit instances in PDTB 2.0. We train the two models following most of the default settings in RoBERTa. The optimizer used in the experiments is AdamW, with an initial learning rate of 1e-5, a batch size of 16, and a maximum of 10 training epochs. The maximum input sequence length is set to 256 tokens.

Figure 6.3 presents their performance on the PDTB 2.0 test set of implicit discourse relations. As shown, there is a substantial performance gap between the two models: the classifier trained on real implicit data outperforms the one trained on explicit-derived data by approximately 20 macro-F1 points.

In this chapter, we aim to answer the question: why does a classifier trained on explicit examples (with connectives removed) perform poorly in a real implicit scenario? We identify that one key cause of this failure is the occurrence of label shift induced by the removal of connectives from explicit examples. This label shift can lead the classifier to learn inconsistent and unreliable patterns, resulting in poor performance when classifying real implicit discourse relations.

6.2 Label Shift in Discourse Relations

6.2.1 What Is Label Shift?

We consider label shift as the difference in relations observed between the same example with and without a connective:

$$\mathrm{Rel}(\mathrm{Arg1}, \mathrm{Conn}, \mathrm{Arg2}) \neq \mathrm{Rel}(\mathrm{Arg1}, \mathrm{Arg2}) \tag{6.1}$$

where $\mathrm{Arg1}$ and $\mathrm{Arg2}$ are the arguments of the example, and Conn denotes the connective. Figure 6.4 shows examples of suffering and not suffering from label shift. Example (6.3) is originally annotated as an *Expansion.Conjunction* relation due to the presence of the connective *and*. However, once the connective *and* is removed, the example tends to convey a *Comparison.Contrast* relation, as suggested by the contrasting lexical cues (e.g., "would

ID	Label Shift	Text	Relation
(6.3)	Yes	[We backed this bill because we thought it would help Skinner]$_{\text{Arg1}}$ [now we're out there dangling in the wind.]$_{\text{Arg2}}$	Comparison.Contrast
		[We backed this bill because we thought it would help Skinner]$_{\text{Arg1}}$ **and** [now we're out there dangling in the wind.]$_{\text{Arg2}}$	Expansion.Conjunction
(6.4)	No	[The procedure causes great uncertainty]$_{\text{Arg1}}$ [an investor can't be sure of his or her individual liability.]$_{\text{Arg2}}$	Contingency.Cause
		[The procedure causes great uncertainty]$_{\text{Arg1}}$ **because** [an investor can't be sure of his or her individual liability.]$_{\text{Arg2}}$	Contingency.Cause

Fig. 6.4 Examples of suffering and not suffering from label shift.

help" vs. "dangling in the wind"). In contrast, Example (6.4) maintains the same relation, *Contingency.Cause*, even after the connective *because* is removed. This is because the first argument describes a result ("uncertain"), while the second provides a reason ("unsure of liability"), thereby preserving the causal semantics.

6.2.2 Do Explicit Examples Suffer from Label Shift?

We manually analyze 100 explicit instances in PDTB 2.0 to ascertain the existence of label shift. Specifically, we randomly sample 100 explicit examples and remove the connectives from each instance. Two student annotators are then trained[1] to label discourse relations according to the PDTB framework, using raw text without connectives. Upon completion of the training, the annotators independently annotate the 100 connective-removed examples. The inter-annotator agreement, measured using Cohen's Kappa,[2] is 0.7346. We find that 37 of these 100 examples are annotated with relations different from the original annotation, suggesting the presence of label shift. We categorize the observed label shift phenomena into three distinct cases:

(i) **Removing connectives leads to different relations.** For example, in Example (6.5), the connective *then* signals a *Temporal* relation, while the arguments express a *Contingency* relation because the first argument describes a result ("stock plummet") and the second points out the reason, a suspension of dividend pay.

[1] See Appendix C.1 for more details about the annotation.
[2] We use the cohen_kappa_score function from the scikit-learn library.

(6.5) [Crossland Savings Bank's stock plummeted.]$_{\text{Arg1}}$ **Then** [management recommended a suspension of dividend payments on both its common and preferred stock.]$_{\text{Arg2}}$

—— Temporal.Asynchronous

(6.6) [Mr. Stein and other officers decided to sell that business]$_{\text{Arg1}}$ **after** [Japanese competitors grabbed a dominant share of the market.]$_{\text{Arg2}}$

—— Temporal.Asynchronous

(6.7) [There's nothing in the least contradictory in all this]$_{\text{Arg1}}$ **and** [it would be nice to think that Washington could tolerate a reasonably sophisticated, complex view.]$_{\text{Arg2}}$

—— Expansion.Conjunction

Fig. 6.5 Different cases suffering from label shift.

(ii) **Deleting connectives causes ambiguity in relations.** This occurs when the arguments contain clues to multiple relations without clearly favoring one. In Example (6.6) in Figure 6.5, the arguments can express either *Contingency* or *Temporal* relations, since inserting *because* or *after* between them is acceptable.

(iii) **No relation is observed after eliminating the connective.** This happens when there are no clues indicating discourse relations, or when the arguments are too short to provide sufficient context. For example, in Example (6.7) in Figure 6.5, there is low lexical cohesion between the two arguments, requiring extensive world knowledge to understand that "Washington" refers to the U.S. government and that "politics" can be "complex" or "contradictory," making it hard to infer any relation.

6.2.3 Does Label Shift Exist at the Corpus Level?

We devise an empirical approach to show that label shift exists at the corpus level. The key idea comes from our definition of label shift, where an example is considered to suffer from label shift if its expressed relations are different when containing or not containing a connective. We mimic this judgment process but replace relations inferred by humans with those predicted by relation classifiers.

Given a corpus containing connectives, either an explicit corpus or an implicit corpus with implicit connectives, we first train a discourse relation classifier using arguments–label pairs from the corpus.[3] We then evaluate the classifier on the same corpus under two conditions:

[3]We did not use examples with connectives to train classifiers because models trained in this way rely heavily on connectives for prediction (Pitler and Nenkova, 2009). By contrast, classifiers trained on arguments without connectives make predictions grounded in the semantics of examples.

Algorithm 5 Measuring Label Shift

Input: Relation Classifier $\mathbf{M}$, Corpus with Connectives $\{(\text{Arg1}_\text{i}, \text{Conn}_\text{i}, \text{Arg2}_\text{i}, \text{Rel}_\text{i})\}|_{\text{i=1}}^{\text{N}}$
Output: $\text{diff\_num}, \text{scores}$

```
Train(M, {(Arg1_i, Arg2_i, Rel_i)}|_{i=1}^{N})
diff_num = 0
scores = []
for i = 1, ..., N do
    # without and with connectives
    p_1 = M.pred(Arg1_i, Arg2_i)
    p_2 = M.pred(Arg1_i, Conn_i, Arg2_i)
    v1 = M.get_rep(Arg1_i, Arg2_i)
    v2 = M.get_rep(Arg1_i, Conn_i, Arg2_i)
    if p_1 ≠ p_2 then
        diff_num = diff_num + 1
    end if
    value = cosine_similarity(v1, v2)
    Append(scores, value)
end for
```

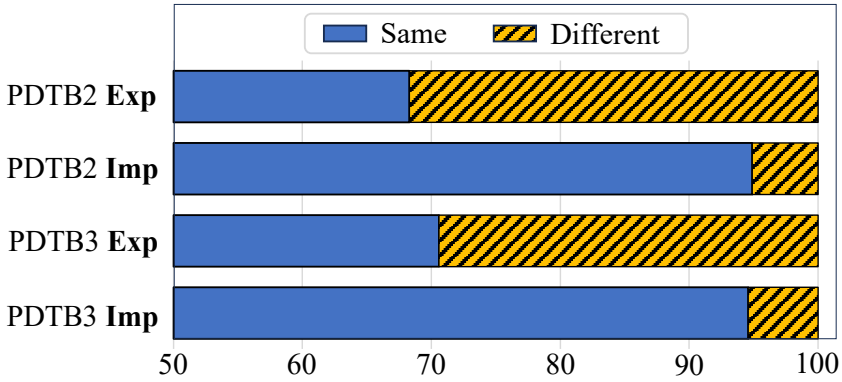
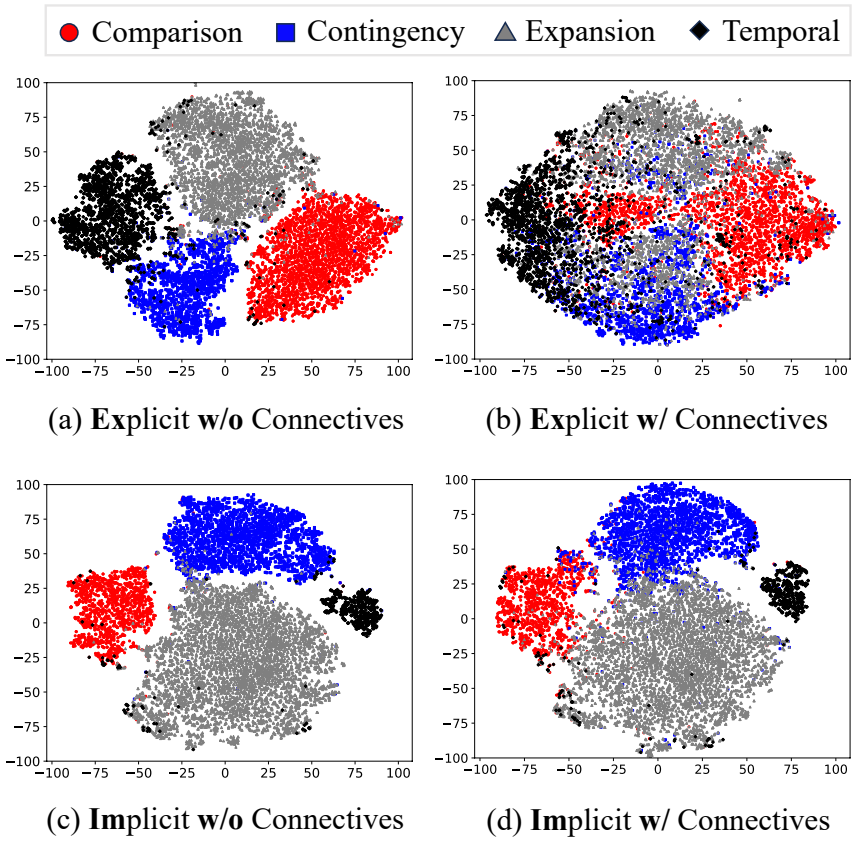


Fig. 6.6 Percentage of examples in **Exp**licit and **Imp**licit corpora that receive the same and different predictions when the input contains and not contains a connective.

with and without connectives (i.e., explicit examples vs. explicit examples with connectives removed, or implicit examples with implicit connectives vs. implicit examples). If there is a substantial difference in the classifier's predictions between the two conditions, quantified by diff_num defined in Algorithm 5, this indicates that connectives can substantially affect the semantics of examples throughout the corpus. That is, label shift exists across the entire dataset.

We perform analyses on both the explicit and implicit portions of the PDTB 2.0 and PDTB 3.0 corpora, providing a comparison between these two types of examples. Figure 6.6 shows the assessment results on PDTB 2.0 and 3.0 (on top-level relations). In explicit



(a) **Ex**plicit **w/o** Connectives (b) **Ex**plicit **w/** Connectives

(c) **Im**plicit **w/o** Connectives (d) **Im**plicit **w/** Connectives

Fig. 6.7 Visualization of example representations in PDTB 2.0 with and without connectives.

corpora, connectives are more likely to influence the predictions of relation classifiers, with approximately 30% of the examples being predicted as different relations when containing and not containing a connective. By contrast, only about 5% of instances in the implicit corpora are predicted in different relations under the same settings.

We further visualize the representations of examples with and without a connective (see **v1** and **v2** in Algorithm 5). Figure 6.7 shows the visualized results on the training set of PDTB 2.0 (top-level relation) using t-SNE (van der Maaten and Hinton, 2008). Without connectives (see Fig. 6.7a), explicit examples are well separated since the classifier is trained on arguments-label pairs. When inserting explicit connectives into inputs (see Fig 6.7b), the representations undergo significant changes, intertwining examples of different relations. Compared to the explicit cases, the representations of implicit instances generally remain unchanged after incorporating connectives (see Fig. 6.7c and 6.7d), suggesting that relations expressed by implicit arguments are barely affected by connectives.

The above results indicate that, after removing connectives, many examples in the explicit corpus express relations that differ from the original annotation. Consequently, classifiers

trained on explicit examples (with connectives removed) learn a chaotic pattern for relation prediction, resulting in poor performance in real implicit scenarios.

6.2.4 Can Label Shift Be Measured?

Different explicit instances exhibit varying degrees of label shift. For example, case (i) in Section 6.2.2 is more severe than case (ii), as deleting the connective causes the former to convey a completely different relation (*Temporal* $\rightarrow$ *Contingency*) while rendering the latter ambiguous (but the original relation holds). We design a **label shift metric** to quantify the degree of label shift occurring in each instance of an explicit corpus. We show in Sections 6.2.5 and 6.3.1 that this metric can be used to analyze factors causing label shift and to filter out noisy examples that suffer from label shift, respectively.

Given an explicit corpus with annotated relations $\{(\mathrm{Arg1}_i, \mathrm{Conn}_i, \mathrm{Arg2}_i, \mathrm{Rel}_i)\}|_{i=1}^{N}$, we first train a classifier using arguments–relation pairs. For each instance in the corpus, we then extract two types of contextualized representations, with and without the connective, using the encoder of the trained classifier. We compute the cosine similarity between these two representations (corresponding to the variable value in Algorithm 5). A cosine similarity close to 1 indicates that the semantic representation of the instance remains largely unchanged with or without the connective, suggesting that the connective is likely removable. Conversely, a low similarity implies that the connective contributes significantly to the meaning of the instance, and its removal may cause a shift in the expressed relation. We apply this label shift metric to the explicit portions of PDTB 2.0 and PDTB 3.0. Our results show that approximately 33% of explicit examples in PDTB 2.0 and 29.6% in PDTB 3.0 exhibit a cosine similarity below 0.5, indicating that a substantial proportion of connectives in these corpora are not removable.

6.2.5 Why Does Label Shift Happen?

While we have demonstrated that label shift occurs during the construction of the *implicit-like* corpus, we know little about why removing a connective has such a significant impact. We investigate four factors that can contribute to label shift:

(i) Is the removed connective a conjunction or an adverb (Prasad et al., 2006)? Conjunctions join clauses of equal grammatical rank in a sentence or join a subordinate clause to a main clause (Blühdorn, 2017). Removing conjunctions disrupts the syntactic structure of the text and may make the expressed relations unclear.

	PDTB 2.0		PDTB 3.0	
	coefficient	p-value	coefficient	p-value
Conjunction vs. Adverb	-0.3946	<0.001	-0.3226	<0.001
Ambiguity	-0.0981	<0.001	-0.0412	<0.001
Intra- vs. Inter-Sentential	-0.1947	<0.001	-0.1898	<0.001
Input length	0.1416	<0.001	0.1944	<0.001

Table 6.1 Pearson correlation between each individual factor and the label shift metric.

(ii) Is the removed connective ambiguous (Webber et al., 2019a)? Some connectives, such as *since*, are ambiguous and signal multiple relations, which may cause the annotated relations of explicit examples to differ from the relations inferred from their arguments.

(iii) Is the status of the arguments intra- or inter-sentential (Prasad et al., 2018)? The information carried by intra-sentential arguments is incomplete (only parts of a sentence) and may not indicate a clear relation without the help of connectives.

(iv) What is the length of the input arguments? Sufficient information is key to inferring relations from text. If the arguments are very short, it will be hard to infer a relation in the absence of connectives.

We extract these four features for each example in the explicit corpus, where the first three are represented as Boolean values (i.e., 0 or 1) and the last one is represented as a floating-point value normalized between 0 and 1.

We calculate the Pearson correlation between each factor and the label shift metric calculated in Section 6.2.4, and show the results on PDTB 2.0 and 3.0 (top-level relations) in Table 6.1. All factors are significantly correlated with the label shift metric (p-value < 0.001), but with different correlation coefficients. The syntactic role played by connectives has the largest absolute correlation value, indicating that whether the removed connective is a conjunction or an adverb has the most impact on the occurrence of label shift. It is followed by the status and length of arguments. Surprisingly, the ambiguity of connectives has the lowest correlation coefficient and shows a clear gap with the other factors. This suggests that the ambiguity of connectives is not the primary cause of label shift in PDTB 2.0 and PDTB 3.0.

The results above show only the correlation of standalone factors with label shift, without considering all factors simultaneously. Inspired by Liu et al. (2023b), we train an XGBoost model (Chen and Guestrin, 2016) to determine the importance of each factor when using the four features to predict the calculated label shift metric. XGBoost is a gradient boosting

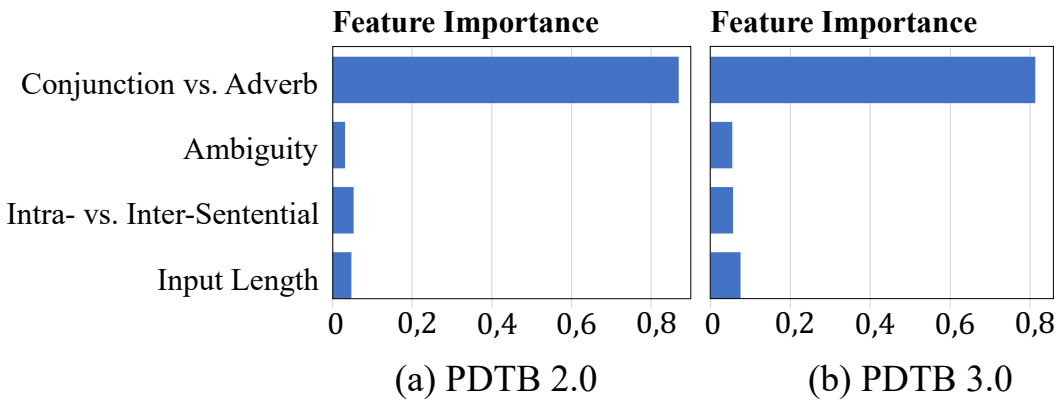


Fig. 6.8 Feature Importance of the XGBoost Model in predicting the label shift metric on PDTB 2.0 and 3.0.

framework, where the importance of a feature can be measured by the performance gain it provides (Shang et al., 2019). The framework also harnesses arbitrary interactions between features and is highly regularized to prevent overfitting, making it suitable to analyze a set of features.

We conduct experiments on PDTB 2.0 and 3.0, and show the results in Figure 6.8. Consistent with the Pearson correlation analysis, the syntactic role played by connectives is found to be overwhelmingly important in predicting the label shift metric, with an importance score exceeding 0.8. In contrast, the status and length of arguments are less important when all factors are considered together. This may be because the three factors, the syntactic role played by the connective, the state of the arguments, and the length of the arguments, are not independent of each other,[4] so the latter two factors provide limited additional information beyond the first feature in predicting label shift. The last feature, the ambiguity of the connective, remains useful but is less important than the other three factors.

6.3 Strategies to Alleviate Label Shift

In this section, we introduce two strategies to alleviate the impact of label shift in the task of *explicit to implicit relation recognition*.

[4]For example, 62.87% of explicit examples (in PDTB 2.0) whose connectives are conjunctions, contain intra-sentential arguments. And inter-sentential arguments are usually longer and contain more words than their intra-sentential counterpart.

Algorithm 6 Filtering Noisy Examples

Input: Examples with scores $\{(\mathrm{E_i}, \mathrm{Rel_i}, \mathrm{s_i})\}|_{\mathrm{i=1}}^{\mathrm{N}}$
Output: Filtered corpus C

```
groups = {}
threshold = {}
C = []
for i = 1, ..., N do
    if Rel_i in groups then
        Append(groups[Rel_i], s_i)
    else
        groups[Rel_i] = [s_i]
    end if
end for

for Rel in groups do
    threshold[Rel] = Avg(groups[Rel])
end for

for i = 1, ..., N do
    if s_i ≥ threshold[Rel_i] then
        Append(C, E_i)
    end if
end for
```

6.3.1 Filter Out Noisy Examples

Our first strategy is straightforward: filtering out examples that may have suffered from label shift. For each instance in the explicit corpus, we calculate the cosine value of each example following the approach in Section 6.2.4, and remove those with low values. Instead of applying a fixed filtering threshold to all relation types, we compute a different threshold for each relation type. This is motivated by the observation that data with different relations suffer from varying degrees of label shift. To implement this, we group examples according to their discourse relation, compute the average cosine value within each group, and discard any instance whose cosine value falls below the corresponding group average (see Algorithm 6).

6.3.2 Joint Learning with Connectives

We further investigate a joint learning framework to alleviate label shift in cases where the filtering result is imperfect. The main idea is that label shift is caused by removing connectives; therefore, if we attempt to recover the discarded connective during training, examples may be more consistent with the original relation labels.

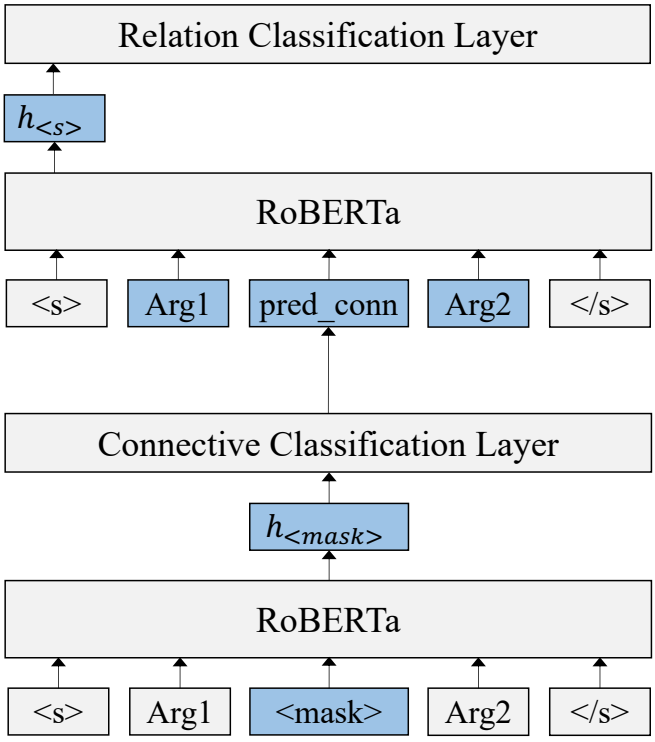


Fig. 6.9 The architecture of the joint learning model.

Given an explicit instance $(\mathrm{Arg1}, \mathrm{Conn}, \mathrm{Arg2}, \mathrm{Rel})$, we replace the connective with a <mask> token inserted between the two arguments. A connective classifier is then trained to predict a suitable connective $\mathrm{pred\_conn}$ for the masked position. Simultaneously, we train a relation classifier to predict a relation based on both input arguments and the predicted connective, i.e., $(\mathrm{Arg1}, \mathrm{pred\_conn}, \mathrm{Arg2})$. By introducing the predicted connective, we hypothesize that the input becomes more semantically aligned with the original example, thereby mitigating the effects of label shift. The architecture of the proposed joint learning model is illustrated in Figure 6.9.

6.4 Experiments

We conduct experiments to demonstrate that our method not only improves the performance of *explicit to implicit relation recognition* on both PDTB 2.0 and PDTB 3.0, but also generalizes well to a corpus annotated with RST relations.

6.4.1 Baselines and Upper Bounds

We evaluate our proposed method on both PDTB 2.0 (Prasad et al., 2008) and PDTB 3.0 (Webber et al., 2019b). For each experiment, we report the average performance over five runs with different random seeds. Our method is compared against existing state-of-the-art approaches for *explicit to implicit relation recognition*. In addition, we include several strong baselines and upper bounds to contextualize the results:

- **Common**: A naive baseline that always predicts the most frequent label in the training set.
- **E2I-Entire**: A standard explicit to implicit setting where RoBERTa is fine-tuned on the entire set of explicit training examples and evaluated on implicit examples.
- **E2I-Reduced**: A variant of E2I-Entire, where the explicit training set is downsampled to match the size of our filtered corpus.
- **I2I-Entire**: An upper-bound setting in which RoBERTa is fine-tuned directly on the entire set of implicit training examples.
- **I2I-Reduced**: A size-controlled version of I2I-Entire, using the same number of training examples as in our filtered corpus.

We follow previous work (Zhou et al., 2022; Long and Webber, 2022) to use $\mathrm{RoBERTa}_{\mathrm{base}}$ as the encoder to train E2I-Entire, E2I-Reduced, I2I-Entire, and I2I-Reduced. The optimizer used in the experiments is AdamW, with an initial learning rate of 1e-5, a batch size of 16, and a maximum of 10 training epochs. The maximum input sequence length is set to 256 tokens.

For our approach, we use the average cosine similarity score within each relation group as the threshold for data filtering. This works well for PDTB 2.0 and PDTB 3.0, but we made a slight modification to the settings for the GUM corpus. Specifically, we filter out an instance (in the GUM corpus) only if its cosine similarity score is lower than the average value of the group it belongs to, and its cosine similarity score is less than 0.6. We do so because the size of the GUM corpus is small (see Table 2.11). If we filter out too many instances, there will not be enough data to train classifiers to converge. For joint learning, we adopt settings nearly identical to those used for the baselines, including the use of $\mathrm{RoBERTa}_{\mathrm{base}}$, the AdamW optimizer, a batch size of 16, a learning rate of 1e-5, a maximum of 10 training epochs, and a maximum input length of 256 tokens.

	Top-level		Second-level	
Models	Acc	F1	Acc	F1
I2I-Entire	$67.97_{0.64}$	$59.74_{0.94}$	$58.11_{0.63}$	$37.74_{0.31}$
I2I-Reduced	$63.77_{0.53}$	$54.66_{1.31}$	$54.07_{0.83}$	$35.49_{0.49}$
Ji et al. (2015)	-	38.62	-	-
Huang and Li (2019)	-	40.90	-	-
Kurfalı and Östling (2021)	-	33.55	25.32	13.01
Common	53.73	17.48	25.22	03.66
E2I-Entire	$56.14_{0.65}$	$41.49_{0.59}$	$34.57_{0.38}$	$22.03_{0.58}$
E2I-Reduced	$55.58_{0.59}$	$39.13_{1.05}$	$31.65_{0.99}$	$18.03_{1.09}$
Our Method	$60.50_{0.34}$	$51.25_{0.70}$	$39.33_{0.28}$	$27.13_{0.50}$
w/o filtering	$58.70_{0.24}$	$45.39_{0.63}$	$36.28_{0.27}$	$23.55_{0.53}$
w/o joint learning	$57.74_{0.45}$	$44.42_{0.83}$	$35.23_{0.34}$	$22.50_{0.48}$

Table 6.2 Results on **PDTB 2.0** (with standard deviation). E2I-Entire is the typical setting for explicit to implicit discourse relation recognition, serving as the baseline, and I2I-Entire is the upper bound for implicit relation classification. Our two strategies can effectively close the gap between the baseline and the upper bound.

6.4.2 Overall Results

The evaluation results on PDTB 2.0 and PDTB 3.0 are presented in Tables 6.2 and 6.3. Classifiers trained on explicit corpora (E2I) perform significantly worse on implicit relation recognition compared to those trained directly on implicit datasets (I2I). For instance, on top-level relations, the E2I-Entire model lags behind the I2I-Entire model by 18.25% and 21.95% in F1 score on PDTB 2.0 and PDTB 3.0, respectively. These findings are consistent with prior research indicating that models trained on explicit examples tend to perform poorly when applied to real implicit relations (Lin et al., 2009). Our proposed method substantially improves *explicit to implicit relation recognition*, narrowing the F1 gap between E2I-Entire and I2I-Entire from 18.25% to 8.49% on PDTB 2.0, and from 21.95% to 16.19% on PDTB 3.0. These results highlight the effectiveness of our approach for the task, which, in turn, demonstrates that label shift is one cause for poor transfer performance from explicit to implicit relations.

Despite these improvements, our method does not fully reach the upper bound established by I2I-Entire. We attribute this to several remaining challenges: (1) explicit and implicit examples differ significantly in syntactic structure (Lin et al., 2009), and (2) the label distributions across explicit and implicit corpora are very different (see Figure 6.7). These

	Top-level		Second-level	
Models	Acc	F1	Acc	F1
I2I-Entire	$72.40_{0.21}$	$67.20_{0.34}$	$62.62_{0.87}$	$53.11_{0.58}$
I2I-Reduced	$69.86_{0.91}$	$64.12_{1.29}$	$59.43_{0.40}$	$46.65_{0.83}$
Common	15.19	27.69	03.10	
E2I-Entire	$51.49_{0.39}$	$45.25_{0.50}$	$39.09_{0.87}$	$33.56_{0.72}$
E2I-Reduced	$48.57_{0.30}$	$40.09_{0.97}$	$36.54_{0.55}$	$28.32_{1.03}$
Our Method	$57.54_{0.16}$	$51.01_{0.45}$	$41.50_{0.30}$	$37.08_{0.13}$
w/o filtering	$52.24_{0.32}$	$46.03_{0.67}$	$40.45_{0.33}$	$34.15_{0.48}$
w/o joint learning	$52.31_{0.38}$	$44.46_{0.56}$	$40.11_{0.28}$	$33.93_{0.30}$

Table 6.3 Results on **PDTB 3.0** (with standard deviation).

differences may give rise to additional types of shifts beyond label shift, which we leave for future investigation.

We further analyze the contribution of each component in our proposed method through an ablation study. Specifically, we evaluate the impact of removing the filtering strategy while retaining the joint learning component. As shown in the "w/o filtering" rows in Tables 6.2 and 6.3, excluding the filtering strategy leads to a decline in performance, with F1 scores for top-level relation recognition decreasing by 5.86% on PDTB 2.0 and 4.98% on PDTB 3.0. Conversely, when we remove the joint learning component and retain only the filtering strategy, the resulting model, structurally similar to the baseline but trained on the filtered corpus, also exhibits degraded performance (see "w/o joint learning" in Tables 6.2 and 6.3), comparable to the effect of removing the filtering strategy. These results highlight the importance of both components for achieving strong performance. Furthermore, we observe that applying each strategy individually yields only marginal improvements, and does not reach the level of effectiveness achieved when both are used in combination. This suggests that (1) neither strategy alone is sufficient to fully address the impact of label shift; and (2) the two strategies are complementary, with their integration leading to a more robust solution.

We also examine whether our filtering strategy can really improve data quality. To this end, we compare the performance of models trained on the same number of training examples drawn from three different sources: our filtered corpus (i.e., "w/o joint learning"), a randomly sampled subset of the original explicit corpus (i.e., "E2I-Reduced"), and a similarly sized subset of the implicit corpus (i.e., "I2I-Reduced"). The results, presented in Tables 6.2 and 6.3, show that models trained on our filtered corpus outperform those trained on E2I-Reduced and achieve performance closer to I2I-Reduced. These findings suggest that our

Models	Acc	F1
I2I-Entire	$64.70_{0.38}$	$53.54_{0.68}$
I2I-Reduced	$55.37_{0.46}$	$48.36_{0.84}$
Common	26.62	06.01
E2I-Entire	$43.62_{1.08}$	$37.83_{1.54}$
E2I-Reduced	$41.21_{1.23}$	$35.24_{1.74}$
Our Method	$48.24_{0.89}$	$43.86_{1.02}$
w/o filtering	$45.66_{0.67}$	$40.21_{0.92}$
w/o joint learning	$45.29_{1.02}$	$40.14_{1.35}$

Table 6.4 Results on the RST GUM corpus.

filtering strategy enhances the quality of the training data, outperforming random sampling from the original explicit corpus.

6.4.3 Results on the GUM Dataset

Our approach is developed based on analyses of the PDTB corpora. To assess its generalizability, we evaluate it on the GUM dataset (Zeldes, 2017b), which is annotated with RST relations. Among the various versions of GUM, we use the most recent release[5] from the DISRPT project, which provides PDTB-style annotations and explicitly labels each discourse instance as either explicit or implicit.

The evaluation results are shown in Table 6.4. The classifier trained on explicit examples (E2I-Entire) performs poorly on implicit relation recognition, lagging behind the classifier trained on implicit examples (I2I-Entire) by more than 15 points in F1 score. Each of our proposed strategies, filtering and joint learning, contributes modest improvements when applied individually. When combined, they achieve the best performance, resulting in a 6-point F1 improvement over the E2I-Entire baseline. These findings indicate that our approach generalizes effectively to other discourse datasets.

6.5 Summary

In this chapter, we show that one cause of the poor transfer performance from explicit to implicit relations is the occurrence of label shift when deleting connectives from explicit examples. We present both manual and empirical evidence to demonstrate the existence of

[5] `https://github.com/disrpt/latest`

such a shift in the explicit corpus. We design a cosine similarity-based metric to measure label shift in the corpus, filter out noisy data, and investigate a joint learning framework to mitigate label shift. Experiments on PDTB 2.0 and PDTB 3.0 demonstrate that training classifiers on the filtered corpus with our joint learning strategy can significantly enhance the performance of *explicit to implicit relation recognition*. Furthermore, we show that our approach also works well on the GUM dataset, suggesting its generalizability.

Chapter 7

Discourse Relation-Enhanced Coherence Modeling

In linguistics, discourse relations between text spans play a crucial role in maintaining textual coherence (Rohde et al., 2018; Jurafsky and Martin, 2025). Consider the example in Figure 8.1, which contains four sentences. This text is regarded as highly coherent due to its well-structured organization through specific discourse relations. In particular, a *Contrast* relation connects the first two sentences, an *Instantiation* relation elaborates on the strike mentioned earlier, and a *Cause* relation introduces the final sentence. Despite their potential

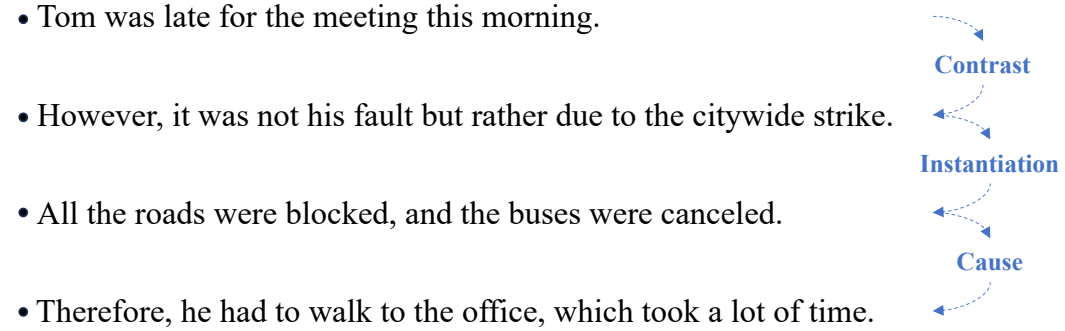


Fig. 7.1 A coherent text with discourse relations.

usefulness, existing works on coherence modeling primarily focus on integrating entity-based features (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013; Tien Nguyen and Joty, 2017; Jeon and Strube, 2022) or applying powerful pre-trained models (Shen et al., 2021; Laban et al., 2021; Abhishek et al., 2022; Liu et al., 2023a), and little attention has been paid to whether discourse relations can contribute to coherence assessment.

One key reason for this gap, as discussed in Chapter 5, is the limited accuracy of previous discourse parsers, particularly for implicit relations. With our improved parser, we investigate how discourse relations contribute to neural coherence modeling. In this chapter, we first present empirical evidence that text coherence is correlated with the sequence of discourse relations inferred from documents. Building on this finding, we introduce a novel fusion model that integrates both text-based and relation-based features to assess coherence. Finally, extensive experiments against strong baselines validate the effectiveness of our proposed approach.

7.1 Discourse Relation and Coherence

This section begins with a concise introduction to discourse relations and their extraction from documents. We then present empirical evidence demonstrating a significant correlation between discourse relation features and varying levels of text coherence. Finally, we demonstrate that a BiLSTM classifier utilizing relation sequences as input achieves performance comparable to a model relying solely on textual input.

7.1.1 Discourse Relations

Discourse relations are a means of logically connecting two segments of discourse. Over the past few decades, various frameworks have been introduced to annotate discourse relations. The most widely used among these are Rhetorical Structure Theory (RST, Mann and Thompson, 1988) and the Penn Discourse Treebank (PDTB, Prasad et al., 2008). In the RST framework, a text is represented as a hierarchical discourse tree, where relations are used to link different text spans. By contrast, PDTB does not postulate any structural constraints on discourse relations and focuses on labeling local discourse relations between sentences and clauses. In this work, we follow previous work (Lin et al., 2011) in adopting PDTB relations and leave RST relations for future work.

We use `discopy` (Knaebel, 2021) as the discourse parser to extract relations from documents, with some adjustments. First, we use the relations in PDTB 3.0 (Webber et al., 2019b) instead of PDTB 2.0 (Prasad et al., 2008), as the newer version offers an expanded set of relations and represents an improved annotation framework. We consider explicit and implicit relations between adjacent sentences of a text. For explicit relations, we consider 15 discourse relations that have sufficient training instances (Liu et al., 2024). For implicit relations, we include the 14 most frequent relations along with a "NoRel" category to account for cases where no discourse relation exists, a common occurrence

Explicit	Distribution	Implicit	Distribution
Asynchronous	8.69%	Asynchronous	4.64%
Cause	7.87%	Cause	24.23%
Concession	19.94%	Cause+Belief	0.82%
Condition	5.99%	Concession	6.72%
Conjunction	36.55%	Condition	0.85%
Contrast	4.58%	Conjunction	20.84%
Disjunction	1.23%	Contrast	3.86%
Instantiation	1.30%	Equivalence	1.21%
Level-of-detail	1.01%	Instantiation	6.84%
Manner	1.23%	Level-of-detail	14.60%
Negative-condition	0.54%	Manner	0.74%
Purpose	1.63%	Purpose	3.31%
Similarity	0.42%	Substitution	1.34%
Substitution	0.96%	Synchronous	2.35%
Synchronous	8.07%	NoRel	8.18%

Table 7.1 Explicit and Implicit relations used in this study and their distribution in the training set of PDTB 3.0.

in low-coherence texts. The complete set of relations used in our study, along with their distributions in PDTB 3.0, is presented in Table 7.1. Second, we adopt the connective-enhanced approach from Liu and Strube (2023) for implicit relation recognition, as it achieves state-of-the-art performance. The parser is trained on the PDTB 3.0 corpus using the data split established by Ji and Eisenstein (2015), with evaluation conducted at the second-level relations. Our implementation achieves an accuracy of 89.61% for explicit relations and 67.80% for implicit relations, demonstrating robust performance that provides a reliable foundation for subsequent analysis.

7.1.2 Correlation Analysis

In coherence theories, discourse relations between text spans play a key role in achieving text coherence (Jurafsky and Martin, 2025). Furthermore, Lin et al. (2011) observed that coherent text exhibits preferences for specific discourse relation ordering. This is somehow verified by Biran and McKeown (2015), which shows that relation N-gram planning (transitions between discourse relations) helps generate coherent text. Inspired by these works, we aim

	GCDC Enron		
		coef	p-value
2-gram	Synchronous → Conjunction	0.3924	<0.01
	Asynchronous → Asynchronous	0.3675	<0.01
	Level-of-detail → Asynchronous	0.3040	0.042
	Cause → NoRel	-0.2300	0.015
3-gram	Cause → NoRel → Conjunction	-0.4835	<0.01
	NoRel → Conjunction → Cause	-0.4359	<0.01
	Cause → Level-of-detail → Conjunction	0.4160	0.012
	Conjunction → Cause → Asynchronous	0.3133	0.056

Table 7.2 Correlation between discourse relation N-gram patterns and coherence levels. Only the top four patterns with the highest absolute correlation coefficients are shown.

to provide evidence demonstrating the correlation between relation N-gram patterns and text coherence.

Dataset. We conduct analyses on two widely used corpora in coherence modeling: the Grammarly Corpus of Discourse Coherence (GCDC) (Lai and Tetreault, 2018) and the TOEFL dataset (Blanchard et al., 2014). GCDC is a corpus constructed for assessing discourse coherence (ADC), containing texts from four domains: **Yahoo**, **Enron**, **Clinton**, and **Yelp**. The TOEFL dataset was originally used for automated essay scoring (AES) but has been used to evaluate coherence models (Burstein et al., 2010; Jeon and Strube, 2020b). See Section 2.2.1 for detailed descriptions these two corpora.

For each document d in the two corpora, we use Stanza (Qi et al., 2020) to segment it into sentences $\{s_1, s_2, ..., s_L\}$ and employ the enhanced `discopy` parser to recognize the relations between adjacent sentences, obtaining a relation sequence $\{r_1, r_2, ..., r_{L-1}\}$, where r_i denotes the parsed relation between s_i and s_{i+1}. From these relation sequences, we extract all relation n-gram transition patterns. Finally, we compute Spearman's rank correlation coefficient[1] between the frequency of each n-gram pattern and the document's ground-truth coherence rating.

Results. Tables 7.2 and 7.3 show the results on the GCDC Enron and TOEFL P1 datasets. In general, relation N-gram features are empirically correlated with coherence levels. For example, in GCDC Enron, relation 3-grams containing *NoRel*, e.g., $\mathrm{Cause} \rightarrow \mathrm{NoRel} \rightarrow \mathrm{Conjunction}$, are negatively correlated with coherence level. In the TOEFL P1 dataset,

[1]Spearman's correlation is particularly appropriate for this analysis as both coherence levels and n-gram frequencies represent ordinal variables.

	TOEFL P1		
		coef	p-value
2-gram	Disjunction → Cause	0.5242	<0.01
	Synchronous → Conjunction	0.4733	<0.01
	Instantiation → Level-of-detail	0.3483	<0.01
	Conjunction → Synchronous	0.3477	<0.01
3-gram	Level-of-detail → Conjunction → Instantiation	0.5239	<0.01
	Conjunction → Contrast → Conjunction	0.5234	<0.01
	Conjunction → Conjunction → Contrast	0.5227	<0.01
	Level-of-detail → Concession → Cause	0.4882	<0.01

Table 7.3 Correlation between discourse relation N-gram patterns and coherence levels. Only the top four patterns with the highest absolute correlation coefficients are shown.

essays containing *Cause* and *Level-of-detail* relations, e.g., $\mathrm{Disjunction} \rightarrow \mathrm{Cause}$, tend to be more coherent. This aligns with existing theories where discourse relations play a key role in achieving text coherence (Rohde et al., 2018). Relation 3-gram patterns seem to be more strongly correlated with text coherence than relation 2-gram ones. For instance, in the TOEFL P1 dataset, 3-gram patterns yield correlation coefficients predominantly exceeding 0.5, compared to approximately 0.4 for 2-grams. We also observe that the two corpora exhibit different relation n-gram patterns correlated with text coherence. This difference may be due to fundamental genre distinctions in discourse organization (Webber, 2009). In the TOEFL corpus, essays are viewpoint-oriented, using evidence (*Cause* relation) and examples (*Instantiation* relation) to support opinions. In contrast, the documents in the GCDC Enron dataset are narrative texts, typically employing *Conjunction* relations. These distributional differences are quantitatively shown in Table 7.4.

7.1.3 Text vs. Relations

To further investigate the role of discourse relations in coherence modeling, we conduct a comparison experiment between two BiLSTM-based classifiers, where the first uses the raw text of the document as input while the other inputs the discourse relation sequence parsed from the document.

Table 7.5 shows the accuracy and macro-F1 results on GCDC Enron and TOEFL P1 datasets. Surprisingly, the classifier built on the discourse relation sequence (Rel Sequence) can achieve comparable performance to that built on raw text. On the GCDC Enron dataset,

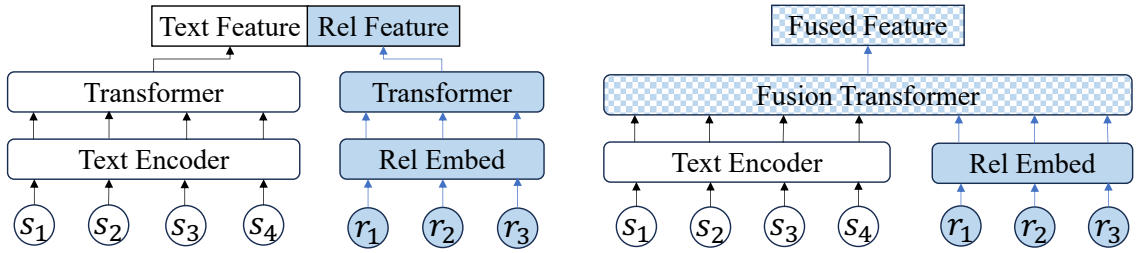
Relation	GCDC Enron	TOEFL P1
Conjunction	33.47%	19.29%
Cause	25.72%	37.40%
Concession	8.92%	7.60%
Level-of-detail	13.48%	11.99%
Asynchronous	4.51%	1.69%
Synchronous	0.80%	0.61%
Contrast	1.09%	4.65%
Instantiation	1.49%	10.58%
NoRel	8.55%	1.12%
Condition	0.32%	0.41%
Purpose	0.32%	0.17%
Substitution	0.57%	1.31%
Manner	0.01%	0.01%
Disjunction	0.01%	0.06%
Equivalence	0.39%	2.56%
Cause+belief	0.26%	0.37%
Negative-condition	0.08%	0.14%
Similarity	0.01%	0.04%

Table 7.4 The distribution of discourse relations parsed from GCDC Enron and TOEFL P1.

Input Type	GCDC Enron		TOEFL P1	
	Acc	F1	Acc	F1
Raw Text	$46.20_{0.77}$	$42.86_{0.97}$	$57.55_{1.24}$	$50.39_{0.78}$
Rel Sequence	$44.15_{0.92}$	$39.43_{1.24}$	$59.17_{0.87}$	$53.51_{0.99}$
Rel Sequence (shuffled)	$37.40_{1.05}$	$31.62_{1.07}$	$50.54_{0.92}$	$43.03_{1.53}$

Table 7.5 The performance (with std) of BiLSTM classifier when using text, relation sequence, and shuffled relation sequence as input, respectively.

the classifier based on the relation sequence only lags behind that on raw text by 2 to 3 points, despite the relation sequence being much shorter than the word sequence of the text. The results on the TOEFL P1 dataset are more encouraging, with the BiLSTM classifier using relations as input outperforming the counterpart based on raw text. These results indicate that discourse relations parsed from the document are useful for coherence modeling. We further investigate the importance of the relation order by training another classifier on the shuffled relation sequence, and show the result in Table 7.5. The results of the classifier



(a) Concatenation (b) Fusion

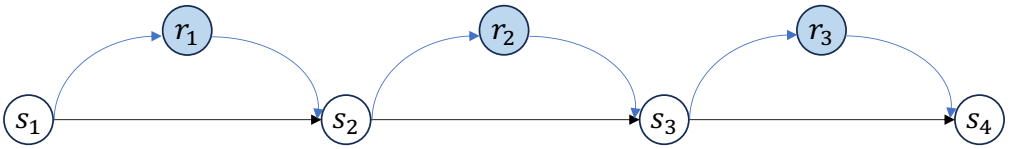
Fig. 7.2 Two ways to combine text- and relation-based features: concatenation vs. fusion.

trained on the shuffled relations lag behind the counterpart trained on the original relation sequence by more than 7 points, strongly indicating that transition patterns between relations are crucial for coherence modeling. In addition, our analysis reveals that only approximately 60% of correct predictions overlap between the text-based and relation-based classifiers. This suggests that raw text and the relation sequence provide different information for coherence assessment.

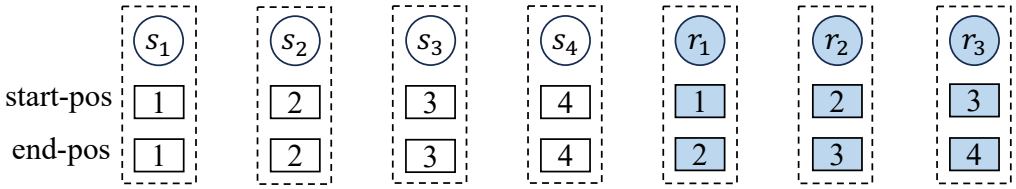
7.2 Discourse Relation-Enhanced Fusion Model

Inspired by the above analyses, we explore approaches in this section to combine text- and relation-based features for coherence modeling. A straightforward way to use both types of information is to extract text- and relation-based features separately, concatenate them, and feed them into a classifier (as shown in Figure 7.2a). However, this concatenation approach fails to capture potential interactions between these two types of features. Prior studies (Ji et al., 2016; Yu et al., 2022a) have demonstrated that incorporating discourse relations into language models can lead to better text representations. Therefore, we investigate a fusion model to facilitate the interaction between text and relation information.

Figure 7.2b shows the overall architecture of the proposed model. First, we use a text encoder and a relation embedding layer to generate sentence and relation representations, respectively. Specifically, given a text $d = \{s_1, s_2, ..., s_L\}$ with L sentences, we input the entire text to a text encoder to obtain representations of tokens $\{\mathbf{e}^t_1, \mathbf{e}^t_2, ..., \mathbf{e}^t_N\}$, where N is the number of tokens in the text. The text encoder can be a pre-trained language model (PLM), such as RoBERTa (Liu et al., 2019), or a large-scale language model (LLM), such as LLaMA (Touvron et al., 2023). Following previous work (Jeon and Strube, 2022), we



(a) Sentences with Relations



(b) Flat Sentences-Relations with Positions

Fig. 7.3 Converting original sentences and parsed relations (a) into a flat sentence-relation structure (b), where start_pos and end_pos denote the start and end positions of the node in the original sentence sequence.

derive the sentence representation by averaging representations of tokens[2] contained in each sentence, i.e., $\mathbf{e}_j^s = \frac{1}{M}\sum_{t_i \in s_j} \mathbf{e}_i^t$, where M is the number of tokens in sentence s_j. Regarding discourse relations $\{r_1, ..., r_{L-1}\}$ parsed from the text, we embed each relation r_j into a vector $\mathbf{e}_j^r = \mathrm{Embed}(r_j)$, where Embed denotes a relation embedding lookup table. Then, we input sentences and relations into a fusion transformer. The challenge here is how to promote the interaction between sentence and relation representations while ensuring that sentences attend to the right relations (and vice versa). We address this through three components: (1) a flat structure of sentences and relations with positional information, (2) a position-aware attention, and (3) a visibility matrix between sentences and relations.

7.2.1 Flat Structure with Positions

After applying the discourse parser, we obtain the sentences of the text (the lower part of Figure 7.3a) and discourse relations between adjacent sentences (the upper part of Figure 7.3a), forming a graph structure. However, since the Transformer is designed for sequence modeling (Vaswani et al., 2017), it is not straightforward for the Transformer to process graph-structured input. One possible solution is to insert relations into the sentence sequence, for example $[s_1, r_2, s_2, ...]$, but the resulting new sequence is no longer natural text.

To address these issues, we introduce a flat structure to organize sentences and relations, in which the two types of elements are concatenated and equipped with positional information

[2]We also experimented with [CLS] pooling but found average pooling is consistently better.

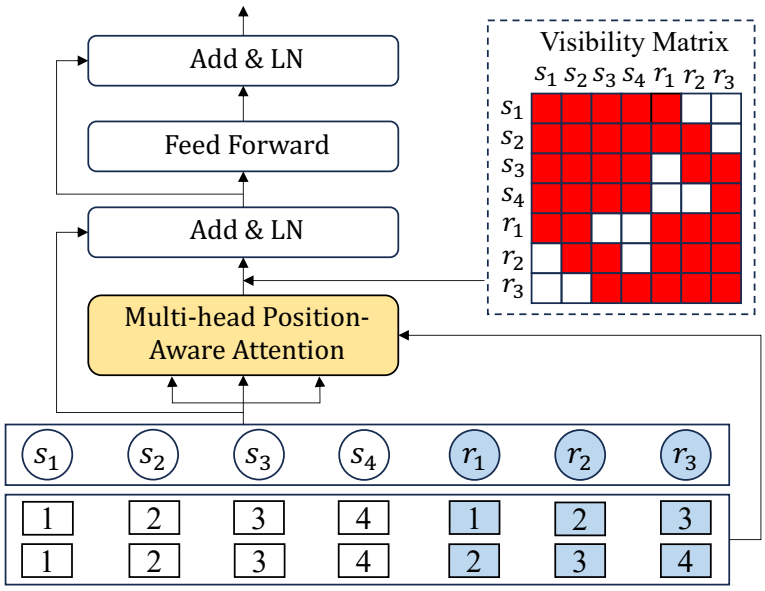


Fig. 7.4 Fusion Transformer.

(as shown in Figure 7.3b). Specifically, sentences and relations are represented as a sequence of triples, where each triple contains three elements: (1) a node, which can be either a sentence or a relation; (2) start_pos and (3) end_pos, denoting the start and end position of the node in the original sentence sequence, respectively. If the node is a sentence, the start and end positions are the same. If the node is a relation, the start and end positions are different, indicating which two sentences the relation connects. For example, (s_1, 1, 1) denotes that this is the first sentence in the text, while (r_1, 1, 2) means that this is a discourse relation connecting the first and second sentences of the text. With this flat structure, we can maintain the original order information of sentences while enabling the Transformer to process these two features (see next section).

7.2.2 Position-aware Attention

The vanilla Transformer encodes the sequence using absolute positions, which is not suitable for our flat structure input. Taking s_1 and r_1 in Figure 7.3b as an example, they are related, but their absolute positions are far apart. Inspired by the self-attention mechanisms proposed in Dai et al. (2019) and Li et al. (2020), we investigate position-aware attention to facilitate the interaction between relevant sentence and relation nodes. The position-aware attention

between the i-th and the j-th nodes is defined as:

$$\mathbf{A}_{ij} = \mathbf{q}_i \mathbf{k}_j^T + \mathbf{q}_i \mathbf{r}_{i-j}^T + \mathbf{u}\mathbf{k}_j^T + \mathbf{v}\mathbf{r}_{i-j}^T \tag{7.1}$$

where $\mathbf{q}_i, \mathbf{k}_j, \mathbf{r}_{i-j} = \mathbf{e}_i\mathbf{W}_q, \mathbf{e}_j\mathbf{W}_k, \mathbf{pe}_{i-j}\mathbf{W}_r$, $\mathbf{e}_i$ denotes the representation of the i-th node, $\mathbf{pe}_{i-j}$ denotes the relative position embedding between the i-th and the j-th nodes, and $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_r$, $\mathbf{u}$, $\mathbf{v}$ are trainable parameters. The first and third terms in Equation 8.1 are content-based addressing, where the former calculates weight between query and key, and the latter governs a global content bias (Dai et al., 2019). The second and last terms compute weight with relative positional information, which helps guide the attention between relevant sentences and relations. Specifically, since each triple in the flat structured input contains two positional information (i.e., `start_pos` and `end_pos`), we can calculate four types of relative distances between the i-th and the j-th nodes: (i) $\mathbf{s}\mathrm{tart_i} - \mathbf{s}\mathrm{tart_j}$; (ii) $\mathbf{s}\mathrm{tart_i} - \mathbf{e}\mathrm{nd_j}$; (iii) $\mathbf{e}\mathrm{nd_i} - \mathbf{s}\mathrm{tart_j}$; and (iv) $\mathbf{e}\mathrm{nd_i} - \mathbf{e}\mathrm{nd_j}$. Under the guidance of relative positional information, a sentence will not only attend to neighboring sentences but also the relation acting upon it. Taking s_1 and r_1 in Figure 7.3b as an example, the distance between the start positions (`start_pos`) of the two nodes is 0, indicating they are very related. The final relative position embedding between the i-th and the j-th nodes, i.e., $\mathbf{pe}_{i-j}$, is defined as a non-linear transformation over the four relative distances:

$$\mathbf{pe}_{i-j} = (\mathbf{p}_{s_i - s_j} \otimes \mathbf{p}_{s_i - e_j} \otimes \mathbf{p}_{e_i - e_j} \otimes \mathbf{p}_{e_i - e_j})\mathbf{W}_p \tag{7.2}$$

The position embedding $\mathbf{p}$ is initialized following the original Transformer formulation, where $\mathbf{p}_{pos}^{2k} = \sin\left(pos/10000^{2k/d_{model}}\right)$ and $\mathbf{p}_{pos}^{2k+1} = \cos\left(pos/10000^{2k/d_{model}}\right)$ (Vaswani et al., 2017).

7.2.3 Visibility Matrix

While relative position embeddings can effectively guide attention calculation, sentence nodes may still attend to irrelevant relation nodes, such as s_1 attending to r_3 (see Figure 7.3a), leading to a poor text representations. Thus, we further introduce a visibility matrix $\mathbf{M}$ (Mihaylov and Frank, 2019) to prevent this. The matrix $\mathbf{M}$ is defined as:

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{if } \mathrm{cond}_1 \mid \mathrm{cond}_2 \mid \mathrm{cond}_3 \\ -\infty, & \text{otherwise} \end{cases} \tag{7.3}$$

where cond_1 and cond_2 are defined as both nodes i and j being either sentences or relations, respectively; and cond_3 is defined as one of the nodes being a sentence and the other a relation, with the relation acting upon the sentence. We apply the visibility matrix to the attention calculation:

$$\mathbf{A}^* = \mathrm{Softmax}(\mathbf{A} + \mathbf{M}) \tag{7.4}$$

Then layer normalizations and a feed-forward network (as shown in Figure 7.4) are applied to produce the text representation $\mathbf{v}$. Finally, we input $\mathbf{v}$ into a softmax classifier and use the cross-entropy loss for training.

7.3 Experiments

We conduct experiments on the GCDC (Lai and Tetreault, 2018) and TOEFL (Blanchard et al., 2014) datasets to show the effectiveness of relation features for coherence modeling.

7.3.1 Experimental Settings

Implementation Details. We implement our model using the PyTorch library, experiment with two different text encoders, a pre-trained language model $\mathrm{RoBERTa}_{\mathrm{base}}$ (Liu et al., 2019), and a large language model $\mathrm{Llama\text{-}2\text{-}7B}$ (Touvron et al., 2023), and initialize the relation embeddings with Glove (Pennington et al., 2014). We use the AdamW optimizer with an initial learning rate of 1e-3, a batch size of 32, and a maximum of 20 training epochs. Considering the training variability in GCDC, we follow the setting in Lai and Tetreault (2018) to perform 10-fold cross-validation over the training dataset. Regarding the TOEFL dataset, we conduct 5-fold cross-validation on the dataset of each prompt, which is a common setting for the AES task (Taghipour and Ng, 2016). Like previous work (Farag and Yannakoudakis, 2019; Jeon and Strube, 2022), we use standard accuracy (Acc, %) as our evaluation metric.

Baselines. To investigate the usefulness of discourse relations, we compare with a baseline using only textual input without any relation information:

- **TextOnly**. This model consists of a text encoder to obtain sentence representations, a sentence-level Transformer to extract coherence patterns, and a softmax classifier for prediction.

To show the effectiveness of our fusion strategy, we compare it with the concatenate baseline:

- **Concat**. This baseline simply concatenates text- and relation-based features without considering interactions between them (see Figure 7.2a).

Text: [Tom was late for the meeting this morning. However, it was ….]

Level of Coherence: **High**

Fig. 7.5 The template used for the Llama-Prompt baseline

Replace the MASK token by selecting only one of the following coherence labels: [Low, Medium, High].

Examples:

Text: [text_1]

Coherence level: **Low**

Text: [text_2]

Coherence level: **Medium**

Text: [text_3]

Coherence level: **High**

<other two examples for each coherence level>

Text: [target_text]

Coherence level: **[MASK]**

Fig. 7.6 The template used for the GPT4-Prompt baseline.

Recently, prompt-based methods using large language models (LLMs) have significantly impacted various NLP tasks. Therefore, we also compare our method with baselines following this trend:

- **Llama-Prompt**. Using LoRA (Hu et al., 2022) to tune Llama-2-7B, and predict the coherence of an input document with the designed template. Figure 2 presents the template employed in this baseline.
- **GPT4-Prompt**. Calling GPT-4o API and applying in-context learning (Min et al., 2022) for coherence assessment. Figure 3 illustrates the prompt utilized for in-context learning.

Further, we compare our method against previous state-of-the-art models on each corpus.

Model		Clinton	Enron	Yahoo	Yelp	Avg
Jeon and Strube (2022)		$64.20_{0.40}$	$55.30_{0.30}$	$58.40_{0.20}$	$57.30_{0.20}$	58.90
Liu et al. (2023a)		$66.20_{0.81}$	$57.00_{0.81}$	$\mathbf{63.65}_{0.74}$	$58.05_{1.21}$	61.23
Llama-Prompt		$62.60_{1.59}$	$57.35_{1.42}$	$60.05_{1.41}$	$57.50_{1.02}$	59.36
GPT4-Prompt		53.00	53.00	50.00	49.00	51.25
RoBERTa	TextOnly	$64.55_{0.69}$	$57.50_{0.89}$	$60.05_{0.35}$	$58.20_{0.75}$	60.10
	Concat	$65.45_{0.79}$	$58.30_{0.56}$	$61.35_{0.67}$	$59.05_{0.57}$	61.04
	Our Method	$\mathbf{66.25}_{0.64}$	$\mathbf{59.60}_{1.26}$	$63.05_{0.42}$	$\mathbf{60.20}_{0.95}$	**62.28**
LLaMA	TextOnly	$63.90_{0.49}$	$57.05_{0.79}$	$59.60_{0.49}$	$57.35_{0.74}$	59.47
	Concat	$64.10_{0.66}$	$57.15_{0.50}$	$61.15_{0.81}$	$58.35_{0.71}$	60.19
	Our Method	$65.75_{0.46}$	$59.30_{0.98}$	$61.70_{0.78}$	$59.45_{0.99}$	61.55

Table 7.6 Mean accuracy results (with std) on the GCDC dataset.

7.3.2 Overall Results

GCDC. Table 7.6 presents the results on the GCDC dataset, where the last two blocks show the results based on RoBERTa and LLaMA. When using RoBERTa as the text encoder, both Concat and Our Method outperform the TextOnly baseline, indicating that relation features are helpful for coherence assessment. The improvement of Concat over the TextOnly baseline is limited, with an increase in accuracy of less than one point (60.10 $\rightarrow$ 61.04). We argue that simply concatenating text- and relation-based features cannot fully utilize relation information since the two features are processed separately, without considering the interaction between them. Compared to Concat, Our Method shows a greater improvement, increasing by 2.18% in accuracy, suggesting that our approach is more efficient in utilizing relation information. When using LLaMA as the text encoder, similar results are observed, showing that relation features are useful across different encoders. Surprisingly, our method implemented with RoBERTa performs better than the counterpart with LLaMA (62.28 vs. 61.55) despite the latter having more parameters and being pre-trained on a larger corpus than the former. We suspect this is because RoBERTa learns bidirectional context-aware representations while LLaMA is limited by its uni-directional context (Yang et al., 2019). Recent work also observed similar results of RoBERTa and LLaMA on other text classification tasks (Rodriguez-Garcia et al., 2024).

The second block in Table 7.6 shows the results of prompt-based methods, including Llama-Prompt and GPT4-Prompt. Similar to using LLaMA as a text encoder, the performance of Llama-prompt also underperforms our method using RoBERTa, with an accuracy gap of 2.92%. The performance of GPT 4 on this task is even worse, lagging behind our

Model		Prompt								Avg
		1	2	3	4	5	6	7	8	
Jeon and Strube (2022)		$78.38_{0.00}$	$75.70_{0.30}$	$76.58_{0.00}$	$76.56_{0.00}$	$\mathbf{79.10}_{0.00}$	$76.41_{0.00}$	$75.03_{0.00}$	$74.54_{0.00}$	76.54
Liu et al. (2023a)		$75.79_{1.14}$	$76.25_{1.07}$	$74.14_{1.18}$	$75.81_{0.71}$	$77.01_{0.94}$	$77.08_{1.14}$	$73.55_{0.80}$	$72.91_{0.66}$	75.34
Llama-Prompt		$76.81_{1.36}$	$76.12_{1.12}$	$76.57_{1.23}$	$75.55_{1.06}$	$76.93_{1.16}$	$76.33_{1.04}$	$76.10_{0.96}$	$74.73_{1.37}$	76.14
GPT4-Prompt		59.21	58.65	64.28	58.27	58.48	65.10	60.23	59.34	57.25
RoBERTa	TextOnly	$76.36_{0.90}$	$75.10_{1.03}$	$75.29_{0.51}$	$75.33_{1.47}$	$75.90_{1.01}$	$75.61_{1.88}$	$73.76_{0.91}$	$73.34_{1.06}$	75.08
	Concat	$77.63_{1.31}$	$75.87_{0.36}$	$76.72_{0.93}$	$76.66_{1.87}$	$78.20_{1.14}$	$77.08_{1.31}$	$75.48_{0.69}$	$74.92_{1.15}$	76.57
	Our Method	$\mathbf{78.97}_{0.75}$	$\mathbf{77.21}_{0.99}$	$\mathbf{77.59}_{0.92}$	$\mathbf{77.19}_{0.90}$	$78.45_{1.14}$	$\mathbf{78.22}_{1.57}$	$\mathbf{76.78}_{0.96}$	$\mathbf{75.85}_{1.06}$	**77.49**
LLaMA	TextOnly	$74.96_{1.17}$	$74.45_{1.55}$	$74.71_{0.43}$	$73.81_{1.45}$	$75.65_{1.55}$	$75.62_{0.96}$	$74.64_{0.93}$	$73.34_{1.02}$	74.65
	Concat	$75.94_{0.75}$	$75.85_{1.21}$	$75.31_{0.56}$	$74.47_{1.47}$	$76.50_{1.19}$	$76.35_{0.98}$	$75.12_{0.74}$	$73.58_{1.23}$	75.39
	Our Method	$77.16_{1.12}$	$76.89_{1.33}$	$76.29_{0.71}$	$76.19_{1.04}$	$77.41_{1.12}$	$77.29_{1.06}$	$76.31_{0.82}$	$75.19_{0.94}$	76.59

Table 7.7 Mean accuracy results (with std) on the TOEFL dataset.

method (RoBERTa) by 11% in accuracy. This is consistent with previous findings that GPT-4 achieves a certain level of accuracy in scoring essays but still underperforms trained models (Mizumoto and Eguchi, 2023). To further show the importance of relation features and the efficiency of our method, we compare against two state-of-the-art models (Jeon and Strube, 2022; Liu et al., 2023a) on this corpus. Both models are entity-based, and their results are shown in the first block of Table 7.6. Our method, using relation features, outperforms the two entity-based models for coherence assessment, indicating its superiority for this task.

TOEFL. Results on the TOEFL dataset are shown in Table 7.7. Similar to the observations on the GCDC dataset, relation features contribute to coherence modeling. When using RoBERTa as the text encoder, Concat and Our Method outperform the TextOnly baseline by 1.49% and 2.41% in accuracy, respectively. The same results are observed when using LLaMA as a text encoder, where the improvement of Concat and Our Method over the TextOnly baseline is 0.72% and 2.05%, respectively. In both settings, Our Method outperforms Concat, demonstrating the effectiveness of fusion for text- and relation-based features. Despite being quite popular in recent research, prompt-based methods, including Llama-Prompt and GPT4-Prompt, perform comparably to other baselines and are slightly inferior to our method using RoBERTa or LLaMA as the text encoder. We further compare our method with previous entity-based approaches. Results in Table 7.7 show that our approach performs better than the two models, highlighting the usefulness of relation features for this task.

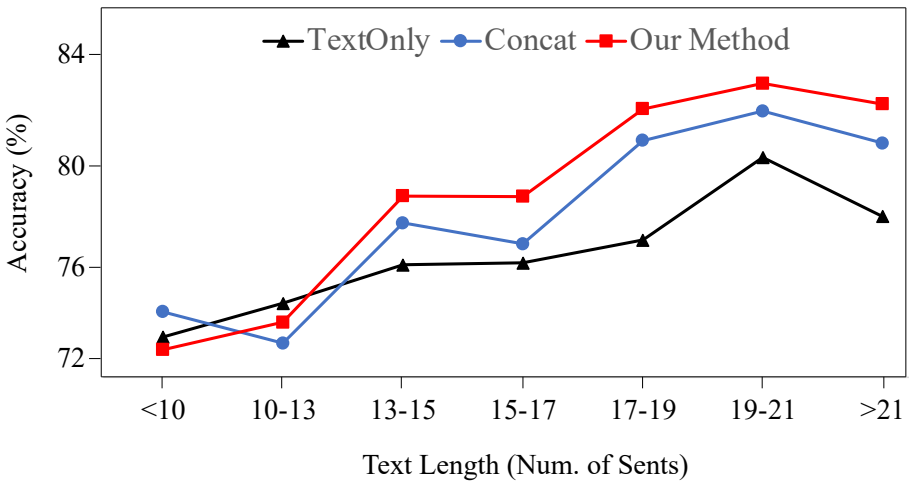


Fig. 7.7 Accuracy against text length.

7.3.3 Performance Analysis

We conducted two analyses to understand why relation features perform well in coherence assessment. First, we compare the performance of Our Method and Concat with the TextOnly baseline across different document lengths (measured by the number of sentences). Figure 7.7 shows the accuracy trends of these three models (using RoBERTa) on the TOEFL P1 dataset as the number of sentences increases. Our Method and Concat show comparable performance to the TextOnly baseline at the beginning, but gradually outperform it as the number of sentences increases, demonstrating that relation information contributes to learning better coherence patterns for longer documents. Our Method consistently outperforms Concat, indicating that it is more efficient in exploiting relation features.

To probe whether our model has truly learned better coherence patterns, we further examine its transferability in cross-domain settings. Specifically, we train TextOnly, Concat, and Our Method on Enron of GCDC (or Prompt 1 of TOEFL), and evaluate their performance on other parts of the GCDC (or other prompts of the TOEFL) datasets. Table 8.4 shows the results of these three models. With relation information, Concat and Our Method consistently show better performance than the TextOnly baseline in the cross-domain setting, indicating the relation sequence of texts can serve as domain-agnostic features for coherence assessment. Our Method outperforms the Concat baseline in all cross-domain experiments, showing the superiority of fusion over simple concatenation.

Model	Enron → Others		TOEFL P1 → Others	
	RoBERTa	LLaMA	RoBERTa	LLaMA
TextOnly	51.83	47.50	71.88	67.70
Concat	53.50(+1.67)	49.83(+2.33)	74.86(+2.98)	70.93(+3.23)
Our Method	56.33(+4.50)	52.33(+4.83)	75.52(+3.64)	72.49(+4.79)

Table 7.8 Cross-domain accuracy of models.

7.3.4 Ablation Study

We conduct ablation studies to evaluate the effectiveness of position-aware attention (PAA) and the visibility matrix (VM). Specifically, we first remove the visibility matrix, and then replace the position-aware attention with a vanilla attention mechanism. Table 7.9 shows the results on the GCDC Enron and TOEFL P1 datasets using RoBERTa. We observe that each component contributes to the performance, showing its essential role in achieving good performance. Furthermore, the performance drop from removing the position-aware attention mechanism is greater than that from eliminating the visibility matrix, indicating that relative position information is more important in guiding fusion.

Model	RoBERTa		LLaMA	
	Enron	TOEFL P1	Enron	TOEFL P1
Our Method	59.60	78.97	59.30	77.16
- VM	59.20	78.12	58.55	76.26
- VM, PAA	58.15	77.24	57.40	75.68

Table 7.9 Ablation study for visibility matrix (VM) and position-aware attention (PAA) in our method.

7.4 Summary

In this chapter, we provide empirical evidence to demonstrate the correlation between discourse relations and text coherence. Then, we introduce a novel fusion model to combine text- and relation-based features for coherence assessment. Experiments on two benchmarks show that our method consistently outperforms various baseline models, demonstrating the importance of relation features and the effectiveness of our approach.

Chapter 8

Coherence Modeling Using Entities and Discourse Relations

In linguistics, coherence can be achieved through various means, such as maintaining reference to the same set of entities across sentences and establishing discourse relations between them. As demonstrated in Chapters 4 and 7, both entity-based and discourse relation-based features independently contribute to the assessment of coherence. However, real-world texts often demand a more integrated perspective to fully account for coherence, as entity and discourse relation cues frequently coexist and interact in complex and interdependent ways. To illustrate this, we present an example in Figure 8.1, which contains four sentences and is considered highly coherent.

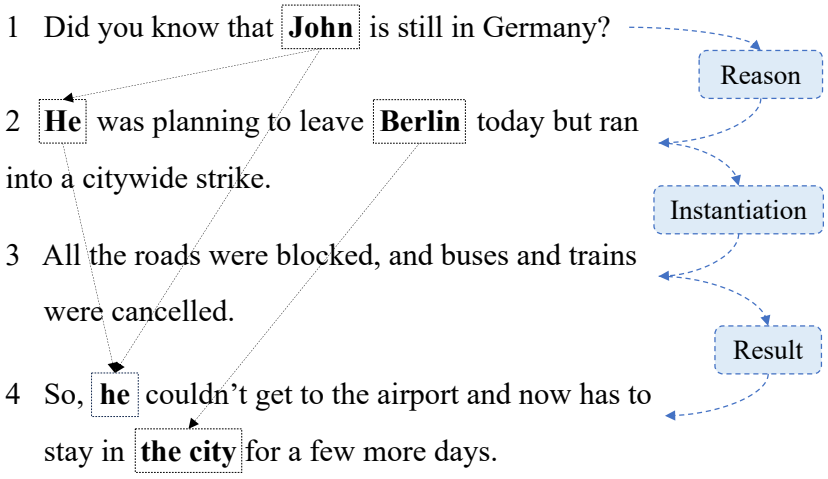


Fig. 8.1 An example of a coherent text, whose coherence should be explained using both entities and discourse relations. We bold the interlinked entities in the text and show the discourse relations between sentences.

Establishing coherence through entities is not straightforward in this case, as there are no overlapping entities between the second and third sentences. Instead, we must use a more complex linguistic phenomenon, namely bridging (Clark, 1975; Hou et al., 2018), to link "city" (in "citywide") and "road". Meanwhile, the connection between these sentences is more readily explained by a discourse relation (e.g., Instantiation), as the third sentence elaborates on the strike mentioned earlier. However, relying solely on discourse relations also has limitations, as it may compromise the smooth tracking of the protagonist if the referents are unclear. For example, if the final sentence were changed to "So, Maria couldn't get to the airport...", the discourse relation might still hold, but the referent switch (i.e., John $\rightarrow$ Maria) would disrupt the overall coherence. This underscores the need to jointly consider both entity continuity and discourse structure.

Although entities and discourse relations offer complementary perspectives on coherence, few studies have empirically examined whether integrating them yields more effective coherence assessment. In this chapter, we propose two approaches to jointly model entities and discourse relations for coherence assessment. Experiments conducted on three benchmark datasets demonstrate that our methods significantly outperform strong baseline models, highlighting the benefits of incorporating both entity- and discourse-based features. Further analysis indicates that this integrated modeling approach facilitates the learning of more robust coherence patterns, helping to alleviate the effects of imbalanced data distributions and enhance the generalization ability of models across domains.

8.1 Method

In this section, we describe how to identify entities and discourse relations in a document, and then present two methods that use them to evaluate coherence.

Given a document, we use Stanza (Qi et al., 2020) to identify all nouns and coreference chains, and to segment the text into sentences. We focus on nouns rather than named entities, as previous studies have shown that nouns yield better performance in coherence modeling (Elsner and Charniak, 2011; Tien Nguyen and Joty, 2017). For discourse relations, we follow prior work (Lin et al., 2011) that adopts the Penn Discourse Treebank (PDTB) framework (Prasad et al., 2006). Specifically, we use the discourse parser `discopy`, developed by Knaebel (2021), to extract relations between adjacent sentences, with a few modifications. First, we use PDTB 3.0 (Webber et al., 2019b) instead of PDTB 2.0 (Prasad et al., 2006), as the newer version includes more relation types and offers several improvements. Second, for implicit discourse relation classification, we use the model proposed by Liu and Strube (2023), which achieves state-of-the-art performance.

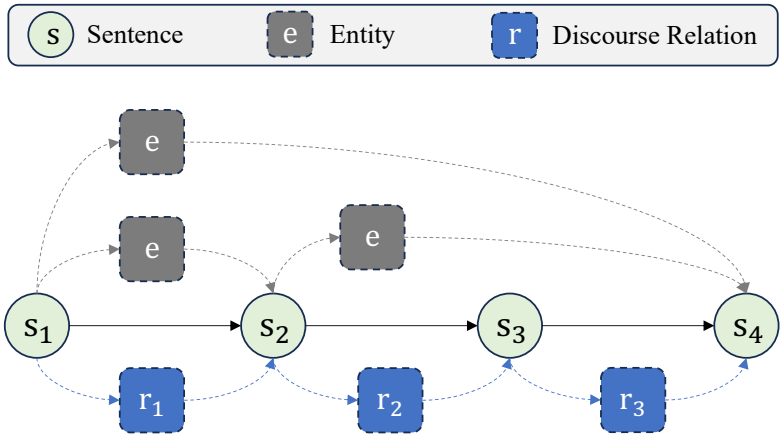


Fig. 8.2 Sentences (in Figure 8.1) linked by entities and discourse relations.

After identifying nouns, coreference relations, and discourse relations, we link two sentences if (1) they share the same nouns or have a coreference link between their mentions, or (2) they are connected by a discourse relation. In the first case, we add an edge labeled "entity" between the sentences; in the second, we add an edge labeled with the specific discourse relation type. Figure 8.2 illustrates how the sentences in Figure 8.1 are linked via these identified entities and discourse relations, forming a graph structure.

However, since the Transformer is designed for sequence modeling (Vaswani et al., 2017), it does not naturally handle graph-structured input. One possible solution is to use Graph Neural Networks (GNNs); however, standard GNNs are permutation-invariant and cannot capture order information (Wu et al., 2021b), which is crucial for coherence modeling (Lapata, 2003). Below, we introduce two approaches to address these issues.

8.1.1 Method I: Fusion

In this approach, we introduce a flat structure to organize sentences, entities, and discourse relations, and design a fusion Transformer to jointly model these elements. Figure 8.3 provides an overview.

In the flat structure, sentences, entities, and discourse relations are concatenated into a sequence. Each element in this sequence is assigned a two-dimensional position (see the bottom part of Figure 8.3), indicating its **start** and **end** positions within the original sentence sequence. Take s_1 and r_1 as an example: their positions are $(1, 1)$ and $(1, 2)$, respectively, meaning that s_1 is the first sentence in the text, and r_1 links the first and second sentences. This flat structure preserves sentence order as well as the connections among sentences, entities,

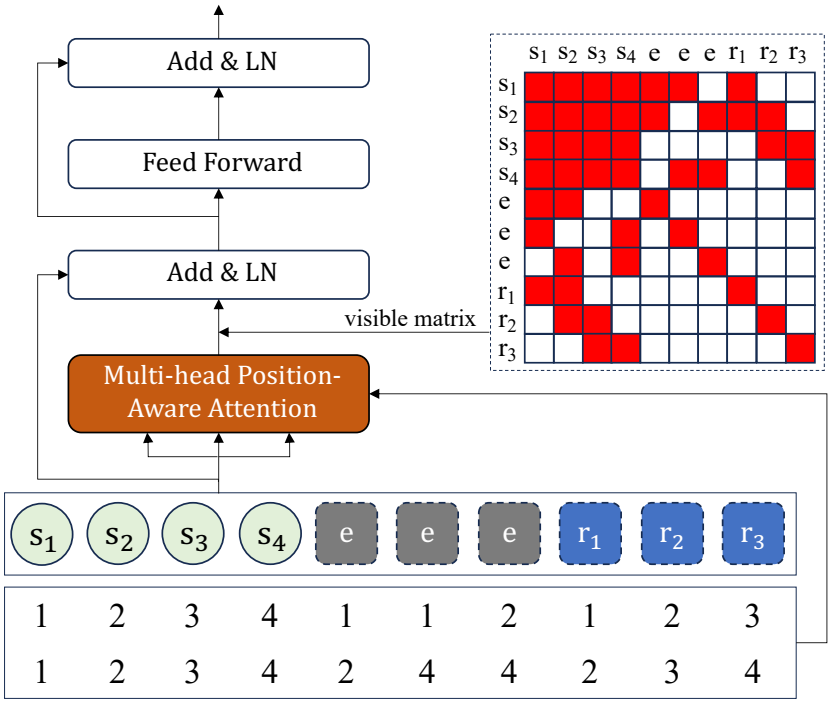


Fig. 8.3 The sentences, entities, and discourse relations in Figure 8.2 are organized into a flat structure, in which each element is assigned a two-dimensional position indicating its start and end within the original sentence sequence. This flat input is then processed by a fusion Transformer.

and discourse relations. Its sequential format also makes it well-suited for Transformer models.

To handle this flat structure, we propose a fusion Transformer that enhances the vanilla Transformer with a novel position-aware attention mechanism and a visibility matrix. Specifically, we first use a text encoder, such as RoBERTa or LLaMA, to obtain representations of sentences, entities, and discourse relations. Then, we feed all elements along with their two-dimensional positions into the position-aware attention module. The position-aware attention between the i-th and j-th elements in the sequence is defined as:

$$\mathbf{A}_{ij} = \mathbf{q}_i \mathbf{k}_j^T + \mathbf{q}_i \mathbf{r}_{i-j}^T + \mathbf{u}\mathbf{k}_j^T + \mathbf{v}\mathbf{r}_{i-j}^T \tag{8.1}$$

where $\mathbf{q}_i, \mathbf{k}_j, \mathbf{r}_{i-j} = \mathbf{e}_i\mathbf{W}_q, \mathbf{e}_j\mathbf{W}_k, \mathbf{pe}_{i-j}\mathbf{W}_r$. Here, $\mathbf{e}_i$ denotes the representation of the i-th element, $\mathbf{pe}_{i-j}$ denotes the relative position embedding between the i-th and j-th elements, and $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_r$, $\mathbf{u}$, and $\mathbf{v}$ are trainable parameters. The first and third terms in Eq.8.1 are content-based addressing: the former computes the attention weight between the query and

key, while the latter introduces a global content bias (Dai et al., 2019). The second and last terms compute weights using relative positional information, which guides attention between relevant elements. Since each element in the flat structure has a 2D position, we compute four types of relative distances between the i-th and j-th elements: (i) $\mathrm{start_i} - \mathrm{start_j}$; (ii) $\mathbf{s}\mathrm{start_i} - \mathbf{e}\mathrm{nd_j}$; (iii) $\mathbf{e}\mathrm{nd_i} - \mathbf{s}\mathrm{start_j}$; (iv) $\mathbf{e}\mathrm{nd_i} - \mathbf{e}\mathrm{nd_j}$. The final relative position embedding $\mathbf{pe}_{i-j}$ is defined as a non-linear transformation over these four relative distances:

$$\mathbf{pe}_{i-j} = (\mathbf{p}_{s_i-s_j} \otimes \mathbf{p}_{s_i-e_j} \otimes \mathbf{p}_{e_i-e_j} \otimes \mathbf{p}_{e_i-e_j})\mathbf{W}_p \tag{8.2}$$

The position embedding $\mathbf{p}$ is initialized following the standard Transformer formulation, where $\mathbf{p}_{pos}^{2k} = \sin\left(pos/10000^{2k/d_{model}}\right)$ and $\mathbf{p}_{pos}^{2k+1} = \cos\left(pos/10000^{2k/d_{model}}\right)$ (Vaswani et al., 2017).

Although relative positional information can effectively guide how nodes attend to one another, the model may still assign attention to irrelevant nodes, for example, a discourse relation node attending to an entity node. To mitigate this issue, we further introduce a visibility matrix $\mathbf{M}$ to guide the attention mechanism:

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{if } \mathrm{C_1} \mid \mathrm{C_2} \mid \mathrm{C_3} \mid \mathrm{C_4} \\ -\infty, & \text{otherwise} \end{cases} \tag{8.3}$$

where $\mathrm{C_1}$ corresponds to $i = j$ (i.e., self-connection), $\mathrm{C_2}$ indicates that both i-th and j-th elements are sentences (text content), $\mathrm{C_3}$ refers to that one element is a sentence and the other is an entity, and the sentence links to the entity (entity patterns), and $\mathrm{C_4}$ is defined as nodes i and j is one sentence and one relation, and the relation works on the sentence (discourse relation patterns). We apply the visibility matrix to the attention calculation:

$$\mathbf{A}^* = \mathrm{Softmax}(\mathbf{A} + \mathbf{M}) \tag{8.4}$$

Then, layer normalization and a feed-forward network (as shown in Figure 8.3) are applied to produce the text representation. Finally, the resulting representation is fed into a softmax classifier, and cross-entropy loss is used for training.

8.1.2 Method II: Prompt

While the first approach can model coherence using entity and discourse relation information, it relies on an additional fusion module and cannot fully leverage the generative capabilities of Large Language Models (i.e., it merely treats LLMs as feature extractors). Inspired

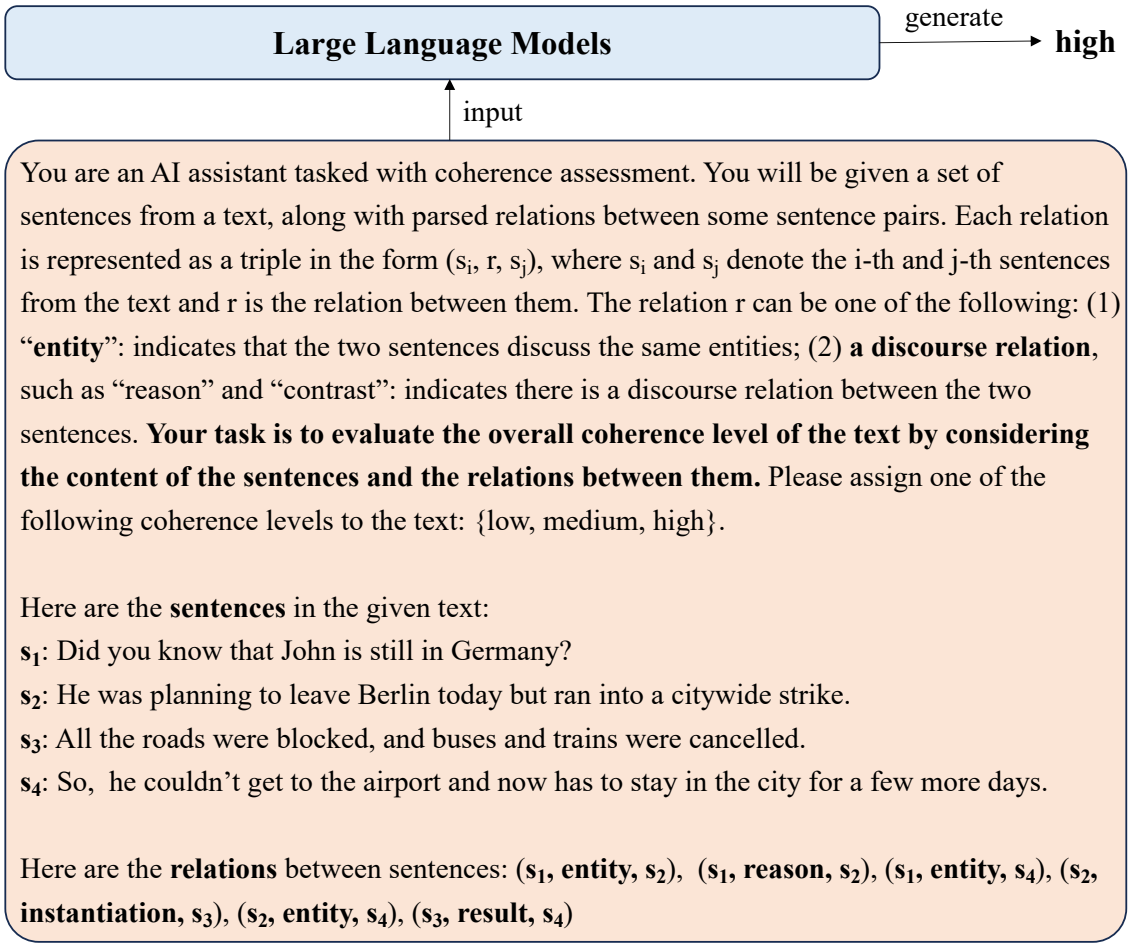


Fig. 8.4 Illustration of our second approach. We use natural language to describe the relationships between sentences, entities, and discourse relations in Figure 8.2, presenting the graph structure in a concise and intuitive way. We then instruct LLMs to consider these elements for coherence assessment.

by Ye et al. (2024), we explore a second approach that uses natural language to describe the connections among sentences, entities, and discourse relations, and then prompts LLMs to take this information into account for coherence assessment. Figure 8.4 illustrates this approach using the example from Figure 8.1 and its corresponding connection graph from Figure 8.2.

Given a graph composed of sentences, entities, discourse relations, and their connections, we traverse all sentence nodes in the order they appear in the text, from left to right. Sentences are added to the prompt and labeled with their positions (e.g., s_1, s_2, etc.; see Figure 8.4). For each sentence node, we perform a depth-first search to find all two-hop neighboring

nodes that are bridged by an entity or a discourse relation. This allows us to break down the graph into a list of triples, where each triple (s_i, r_{ij}, s_j) includes two sentences, s_i and s_j, along with the relation r_{ij} between them. We only retain triples where $i < j$, following the natural left-to-right reading order of humans, as suggested by Liu et al. (2023a). For example, the graph in Figure 8.2 is broken down into the following triples: (s_1, entity, s_2), (s_1, reason, s_2), (s_1, entity, s_4), (s_2, instantiation, s_3), (s_2, entity, s_4), and (s_3, result, s_4). These triples are expressed in natural language format, making them easy for LLMs to process. More importantly, they retain all the connection information between sentences, entities, and discourse relations. Finally, we include the list of triples in the prompt and instruct the LLMs to assess coherence by considering both the content of the sentences and the patterns of entities and discourse relations between them (see Figure 8.4).

8.2 Experiments

8.2.1 Experimental Settings

Datasets. We conduct experiments on three widely used corpora for coherence modeling: GCDC (Lai and Tetreault, 2018), CoheSentia (Maimon and Tsarfaty, 2023), and TOEFL (Blanchard et al., 2014). GCDC is designed for evaluating discourse coherence, containing texts from four distinct domains: **Yahoo**, **Enron**, **Clinton**, and **Yelp**. CoheSentia is another dataset used to assess discourse coherence. Unlike GCDC, which consists of real-world texts, CoheSentia contains stories generated by GPT-3 and is annotated by humans with coherence scores ranging from 1 to 5. However, the score distribution is highly imbalanced,[1] which makes it difficult for models to converge during training (Maimon and Tsarfaty, 2023). To address this, we group scores 1 and 2 as low coherence, scores 3 and 4 as medium coherence, and score 5 as high coherence. The TOEFL dataset was originally created for automated essay scoring but has since been widely used to evaluate coherence models (Burstein et al., 2010; Jeon and Strube, 2020a). See Section 2.2.1 for detailed descriptions of GCDC and TOEFL.

Implementation Details. We implement our models using the PyTorch library. For Method I, we experiment with two widely used text encoders (Abhishek et al., 2022; Parmar et al., 2024): the pre-trained language model $\mathrm{RoBERTa_{base}}$ (Liu et al., 2019) and the large language model $\mathrm{Llama\text{-}3.1\text{-}8B\text{-}Instruction}$ (Grattafiori et al., 2024).[2] Training is performed using the

[1] Over 50% of the data is labeled with a score of 5.

[2] We use the 8B LLaMA model instead of the 70B version due to memory limitations that prevent fine-tuning larger models. However, our resources do support zero-shot experiments with the 70B model. To maintain

AdamW optimizer with an initial learning rate of 1e-3, a batch size of 32, and a maximum of 20 epochs.

For Method II, which is specifically designed for large language models (LLMs), we evaluate it using Llama-3.1-8B-Instruction.[2] The evaluation is conducted under two settings: **zero-shot** and **fine-tuned**. In the **zero-shot** setting, the model is not trained beforehand; instead, it is directly prompted to generate labels. This setup tests whether incorporating entity and discourse relation features can assist coherence evaluation in cold-start scenarios. In the **fine-tuned** setting, we fine-tune the LLaMA model using LoRA (Hu et al., 2022) for 3 epochs with a learning rate of 5e-5 and a batch size of 2. This setup evaluates whether instruction-tuning the LLM to consider entities and discourse relations can enhance its performance.

To account for training variability, we perform 10-fold cross-validation on the GCDC training dataset (Lai and Tetreault, 2018), 5-fold cross-validation on the CoheSentia corpus, and 5-fold cross-validation on each prompt-specific dataset in the TOEFL corpus (Taghipour and Ng, 2016). Following prior work, we use standard accuracy (Acc, %) as our primary evaluation metric.

Baselines. To validate the importance of modeling entities and discourse relations simultaneously, we compare our approach with the following baselines:

- **TextOnly**. This baseline relies solely on textual information for coherence modeling. In Method I, it uses a text encoder to obtain sentence representations, a sentence-level Transformer to capture coherence patterns, and a softmax classifier for prediction. In Method II, it prompts LLMs to evaluate coherence based only on the text.
- **TextEnty**. This is an ablated version of our approach in which the discourse relation elements are removed from the sentence-entity-discourse relation graph.
- **TextRel**. This is another ablated version of our method, where we remove the entity elements from the graph.

Furthermore, we compare our approaches against previous state-of-the-art models on each corpus.

8.2.2 Overall Results

GCDC / CoheSentia. Table 8.1 presents the results on the GCDC and CoheSentia datasets. The "Fusion" block reports the results obtained using an additional fusion module to integrate

consistency across settings, we use the 8B model throughout the main text but include zero-shot results for the 70B model in the Appendix D.2.

Model			GCDC					CoheSentia
			Clinton	Enron	Yahoo	Yelp	Avg	
Jeon and Strube (2022)			$64.20_{0.4}$	$55.30_{0.3}$	$58.40_{0.2}$	$57.30_{0.2}$	58.90	-
Liu et al. (2023a)			$66.20_{0.8}$	$57.00_{0.8}$	$63.65_{0.7}$	$58.05_{1.2}$	61.23	-
Fusion	RoBERTa	TextOnly	$64.55_{0.7}$	$57.50_{0.9}$	$60.05_{0.4}$	$58.20_{0.8}$	60.10	$60.64_{1.5}$
		TextEnty	$66.20_{0.8}$	$58.80_{1.1}$	$63.15_{0.9}$	$59.20_{1.1}$	61.83	$63.13_{2.0}$
		TextRel	$66.45_{0.9}$	$59.70_{1.0}$	$63.35_{1.1}$	$60.40_{1.3}$	62.48	$63.74_{1.8}$
		Our Method I	$\mathbf{67.60}_{0.5}$	$60.50_{0.3}$	$63.75_{0.5}$	$61.10_{0.4}$	**63.24**	$66.24_{1.6}$
	LLaMA	TextOnly	$63.55_{0.5}$	$56.65_{0.8}$	$59.45_{0.8}$	$57.45_{1.0}$	59.27	$63.13_{1.2}$
		TextEnty	$64.80_{0.8}$	$58.10_{0.4}$	$62.10_{0.5}$	$57.90_{0.8}$	60.73	$65.80_{1.5}$
		TextRel	$65.10_{0.7}$	$58.75_{0.4}$	$62.85_{0.3}$	$59.35_{0.5}$	61.51	$66.65_{1.6}$
		Our Method I	$67.25_{0.4}$	$60.10_{0.3}$	$\mathbf{64.10}_{0.5}$	$\mathbf{61.30}_{0.5}$	63.18	$\mathbf{69.12}_{1.5}$
Prompt	LLaMA zero-shot	TextOnly	54.50	38.00	34.00	40.50	40.88	50.10
		TextEnty	55.00	39.00	41.50	44.50	45.00	51.35
		TextRel	57.50	41.00	42.00	45.50	46.50	52.17
		Our Method II	56.50	41.00	42.00	48.00	46.88	53.83
	LLaMA fine-tuned	TextOnly	$63.55_{0.8}$	$56.80_{0.9}$	$60.05_{1.0}$	$55.45_{1.2}$	58.96	$64.95_{1.4}$
		TextEnty	$65.00_{1.2}$	$57.60_{0.5}$	$60.45_{1.0}$	$56.30_{0.9}$	59.84	$65.38_{1.5}$
		TextRel	$64.55_{0.7}$	$59.10_{0.5}$	$61.10_{0.7}$	$57.25_{0.5}$	60.50	$66.42_{1.4}$
		Our Method II	$65.15_{0.6}$	$\mathbf{60.55}_{1.2}$	$62.05_{1.2}$	$57.55_{0.5}$	61.33	$67.28_{1.1}$

Table 8.1 Mean accuracy results (with std) on GCDC and CoheSentia.

entity and discourse relation features, while the "Prompt" block shows the results based on incorporating entity and discourse relation patterns into the input prompt of LLMs using natural language.

For the Fusion style, we report results based on RoBERTa and LLaMA. Regardless of which encoder is used, TextEnty and TextRel consistently outperform the TextOnly baseline on GCDC and CoheSentia. This suggests that incorporating entity or discourse relation features enhances coherence assessment, aligning with the findings of previous entity-based (Jeon and Strube, 2022) and discourse relation-based studies (Wu et al., 2023). The improvement of TextRel over TextOnly is greater than that of TextEnty over TextOnly, likely because discourse relations are more commonly used than entity cues to connect sentences in both GCDC and CoheSentia. For instance, discourse relations like cause and concession are frequently employed in CoheSentia to make stories more compact and engaging (Chaturvedi et al., 2017). Our Method I significantly outperforms both the TextEnty and TextRel baselines, showing a 1-2% improvement on GCDC and approximately a 3% gain on CoheSentia. These

Model			P1	P2	P3	P4	P5	P6	P7	P8	Avg
Jeon and Strube (2022)			78.38	75.70	76.58	76.56	79.10	76.41	75.03	74.54	76.54
Liu et al. (2023a)			$75.79_{1.1}$	$76.25_{1.1}$	$74.14_{1.2}$	$75.81_{0.7}$	$77.01_{0.9}$	$77.08_{1.1}$	$73.55_{0.8}$	$72.91_{0.7}$	75.34
Fusion	RoBERTa	TextOnly	$76.36_{0.9}$	$75.10_{1.0}$	$75.29_{0.5}$	$75.33_{1.5}$	$75.90_{1.0}$	$75.61_{1.9}$	$73.76_{0.9}$	$73.34_{1.1}$	75.08
		TextEnty	$79.05_{1.4}$	$77.15_{1.2}$	$77.73_{0.8}$	$76.98_{1.3}$	$77.64_{1.6}$	$78.32_{1.5}$	$76.49_{1.3}$	$75.79_{1.0}$	77.39
		TextRel	$78.94_{0.8}$	$77.41_{0.7}$	$77.80_{0.8}$	$77.55_{0.8}$	$78.49_{0.9}$	$78.33_{1.5}$	$77.08_{1.2}$	$76.25_{0.5}$	77.73
		Our Method I	$79.92_{0.8}$	$\mathbf{78.46}_{0.9}$	$\mathbf{78.68}_{0.9}$	$\mathbf{78.25}_{1.2}$	$79.23_{1.1}$	$79.42_{1.27}$	$78.21_{0.9}$	$77.13_{1.1}$	**78.66**
	LLaMA	TextOnly	$75.17_{0.8}$	$73.88_{1.3}$	$73.63_{1.6}$	$73.67_{1.4}$	$75.89_{1.0}$	$75.10_{0.9}$	$73.67_{1.4}$	$72.87_{1.5}$	74.24
		TextEnty	$77.03_{0.8}$	$75.59_{1.4}$	$75.14_{1.5}$	$75.20_{1.5}$	$77.07_{0.9}$	$77.12_{0.8}$	$75.48_{0.6}$	$74.17_{1.4}$	75.85
		TextRel	$76.35_{0.9}$	$76.40_{0.7}$	$75.98_{0.5}$	$75.40_{1.2}$	$76.64_{1.7}$	$76.65_{1.6}$	$75.18_{1.1}$	$75.16_{1.3}$	75.97
		Our Method I	$78.24_{1.7}$	$78.11_{1.9}$	$77.01_{1.1}$	$76.59_{1.1}$	$79.23_{1.3}$	$\mathbf{79.47}_{1.6}$	$77.32_{1.1}$	$76.50_{1.8}$	77.81
Prompt	LLaMA zero-shot	TextOnly	51.39	55.19	52.72	50.63	54.37	50.62	46.92	49.44	51.41
		TextEnty	56.85	53.78	54.48	54.00	53.83	57.15	55.89	54.64	55.08
		TextRel	58.51	56.45	54.73	55.59	56.43	57.19	57.41	53.72	56.25
		Our Method II	59.90	57.75	56.73	56.13	57.28	58.02	58.19	55.91	57.49
	LLaMA fine-tuned	TextOnly	$79.03_{1.1}$	$76.76_{1.4}$	$76.24_{1.5}$	$77.52_{1.4}$	$79.49_{1.4}$	$76.02_{1.4}$	$76.69_{1.1}$	$75.28_{0.9}$	77.13
		TextEnty	$\mathbf{80.13}_{1.2}$	$76.63_{1.2}$	$75.64_{1.3}$	$77.73_{1.0}$	$79.55_{1.5}$	$76.57_{1.6}$	$\mathbf{78.95}_{1.4}$	$76.41_{1.3}$	77.70
		TextRel	$79.35_{1.5}$	$77.15_{1.6}$	$77.16_{1.4}$	$76.61_{1.2}$	$80.15_{1.1}$	$75.41_{1.5}$	$78.29_{1.3}$	$76.89_{1.4}$	77.63
		Our Method II	$80.02_{1.6}$	$77.92_{1.5}$	$77.58_{1.2}$	$78.13_{1.3}$	$\mathbf{81.13}_{1.5}$	$77.29_{1.3}$	$77.88_{1.0}$	$\mathbf{77.18}_{1.5}$	78.39

Table 8.2 Mean accuracy results (with std) on TOEFL dataset.

results highlight the value of jointly modeling entity and discourse relation features for effective coherence assessment.

For the Prompt style, we present the results of LLaMA in both zero-shot and fine-tuned settings. In the zero-shot setting, incorporating entity and discourse relation information enhances LLaMA's performance in coherence assessment. On GCDC, TextEnty and TextRel outperform the TextOnly baseline by over 4-5%. In contrast, the improvement on CoheSentia is more modest, with gains of about 1-2%. Combining these features further boosts performance, resulting in improvements of over 6% on GCDC and 3.5% on CoheSentia, compared to the TextOnly baseline. These results suggest that prior knowledge of entity- and discourse relation-based coherence can be effectively leveraged for coherence assessment in cold-start scenarios. When fine-tuning LLaMA with LoRA, the performance improvements of TextEnty, TextRel, and EntyRel over TextOnly still exist, but the gains are smaller than in the zero-shot setting. We speculate that this is because fine-tuning allows the model to somewhat implicitly capture coherence-relevant signals, such as entity transitions and discourse relations (Xiao et al., 2021), so the explicit incorporation of them leads to limited improvement.

		Low	Medium	High	Range
Fusion (LLaMA)	TextOnly	66.67	78.99	77.88	12.32
	TextEnty	73.24	80.44	76.79	7.20
	TextRel	74.36	80.45	78.41	6.09
	Our Method I	81.16	81.99	77.19	4.80
Prompt (fine-tuned)	TextOnly	68.22	83.29	82.93	15.07
	TextEnty	71.70	85.23	85.49	13.79
	TextRel	70.59	84.09	84.05	13.50
	Our Method II	73.47	85.39	84.71	11.92

Table 8.3 Accuracy results for each coherence level on TOEFL P5. Range indicates the difference between the highest and lowest values.

TOEFL. Results on TOEFL are shown in Table 8.2. Similar to the findings on GCDC and CoheSentia, both entity and discourse relation patterns contribute positively to the task in the fusion setting. Specifically, TextEnty and TextRel outperform the TextOnly baseline by 2-3% when using RoBERTa or LLaMA as the text encoder. Combining entity and discourse relation features further enhances performance. Our Method I, using RoBERTa as the text encoder, achieves an average accuracy of 78.66%, significantly outperforming the previous state-of-the-art model (Jeon and Strube, 2022). We observe similar results in the prompt setting: in the zero-shot scenario, Method II achieves an accuracy of 6.08% higher than the TextOnly baseline, and 1.26% higher in the fine-tuned setting.

8.2.3 Analysis

To understand how jointly modeling entities and discourse relations contributes to coherence assessment, we analyze the accuracy of different models across each coherence level. Table 8.3 presents the models' performance on the TOEFL P5 dataset in both the Fusion setting (with LLaMA as the text encoder) and the fine-tuned Prompt setting. TextOnly exhibits a strong bias, achieving high accuracy on "medium" and "high" coherence labels but significantly lower accuracy on the "low" label. We attribute this to the imbalanced label distribution in the TOEFL P5 dataset, where over 90% of samples are annotated as "medium" or "high" coherence. TextEnty and TextRel help mitigate this bias by incorporating entity and discourse relation information, respectively. For example, in the Fusion setting, they improve accuracy on low-coherence data by 6.57% and 7.69%, respectively. Our Methods I and II go further by jointly modeling entities and discourse relations, resulting in the smallest performance gap across all three coherence levels. These results suggest that incorporating

[There's two older men here that are always in a lousy mood.]$_1$ [They can't wait for 62.]$_2$ [Andrea M. is correct about the cart corrals.]$_3$ …[Really saves us money in that dept.]$_7$ [But I can only give 3 stars on account that their pricing for vegetables and other items is a ripoff.]$_8$ [Why is there only one dollar sign showing above?]$_9$ [People are SOOOO brainwashed into thinking that Fry's and Walmart save you money.]$_{10}$ …

(s_1, cause, s_2), (s_2, norel, s_3), (s_3, level-of-detail, s_4), (s_4, norel, s_5), (s_5, norel, s_6), (s_6, cause, s_7), (s_7, norel, s_8), (s_7, entity, s_{10}), …

Model	Prediction	Explanation
TextOnly	High	…… The reviewer's tone is conversational and informal, but their points are clearly stated and easy to follow.
TextEnty	Low	The relation between s_7 and s_{10} is "entity", indicating that they discuss the same entities. However, s_7 is discussing the cost-effectiveness of prescriptions, while s_{10} is discussing the cost-effectiveness of Fry's and Walmart. This lack of coherence between the entities being discussed suggests a low coherence level.
TextRel	Medium	The relations between sentences provide a clear narrative flow, but the contrast between the customer's satisfaction with the store's prices and their dissatisfaction with the pricing of vegetables creates a sense of inconsistency.
Ours	Low	The relations between sentences provide a clear narrative flow, but the content of the sentences are often abrupt and lack cohesion, indicating a low level of coherence.

[Look at you...such a hoax spreader...ha]$_1$ [Anyway, i was so busy yesterday i didn't have a chance to do anything about it anyway.]$_2$ [so no harm, no foul.]$_3$ [how are you doing?]$_4$ [as for Enron, as you well know our stock's not doing so great these days, but i'm not overly stressed.]$_5$ … [Can't wait to hear how your preparations are coming along...for the wedding and the move.]$_7$ [btw, have you set a firm date yet?]$_8$

(s_1, norel, s_2), (s_2, cause, s_3), (s_3, norel, s_4), (s_4, conjunction, s_5), (s_5, conjunction, s_6), (s_5, entity, s_6), (s_6, conjunction, s_7), (s_7, conjunction, s_8)

TextOnly	Medium	… seemingly unrelated statements, …. the presence of a few cohesive elements, … indicates some level of coherence.
TextEnty	Medium	… various unrelated topics… s5 and s6 discuss the same entities, which is a positive sign of coherence.
TextRel	Low	The text discusses various unrelated topics … The relations between sentences are mostly conjunctions and sometimes norel, which indicate a simple sequence of ideas rather than a deeper connection between them.
Ours	Low	The text discusses various unrelated topics… The relations between sentences are mostly conjunctions, indicating a lack of strong logical connections between the ideas.

Fig. 8.5 Two examples (truncated) showing how entities and discourse relations aid coherence assessment. Both texts are labeled as low coherence. We use a zero-shot prompt setting, and the "explanation" refers to LLaMA's brief justification for its prediction.

entities and discourse relations helps the model learn more effective coherence patterns and improves its robustness to imbalanced data distributions.

To better understand how entities and discourse relations influence model behavior, we present two case studies in Figure 8.5. The two examples are from the GCDC corpus and annotated as low coherence. In both cases, we use a zero-shot prompt setting, asking LLaMA to evaluate the coherence level of a given text and provide a brief explanation[3] for its assessment (see Appendix D.1 for details). As shown in the first example, without entity and discourse relation information (i.e., TextOnly), LLaMA evaluates the text as having high coherence. TextRel identifies some inconsistencies but still fails to classify it as medium coherence. In contrast, TextEnty and Our Method II correctly assess the text as having low coherence, due to the lack of cohesion, specifically, missing entity-based signals. In the second example, all models recognize that the sentences in the text cover various unrelated topics. However, TextOnly and TextEnty are slightly influenced by the presence of cohesive elements, leading them to predict the text as medium coherence. In contrast, TextRel and Our Method II correctly and confidently classify it as low coherence, due to the lack of logical connections between the sentences. These two cases effectively illustrate the importance of modeling both entity and discourse relation patterns for accurate coherence assessment.

		Enron → Others	TOEFL P1 → Others
Fusion (LLaMA)	TextOnly	47.48	68.79
	TextEnty	50.62 (+3.14)	72.02 (+3.23)
	TextRel	50.98 (+3.55)	72.87 (+4.08)
	Our Method I	53.82 (+6.34)	74.40 (+5.61)
Prompt (fine-tuned)	TextOnly	52.50	76.72
	TextEnty	53.67 (+1.17)	78.42 (+1.70)
	TextRel	54.75 (+2.25)	78.15 (+1.43)
	Our Method II	56.00 (+3.50)	78.60 (+1.88)

Table 8.4 Accuracy of models in a cross-domain setting.

To assess whether our models have truly learned more robust coherence patterns, we further evaluate their transferability in cross-domain settings. Specifically, we train TextOnly, TextEnty, TextRel, and Our Method in both the Fusion and Prompt settings on the Enron subset of GCDC (or Prompt 5 of TOEFL) and test their performance on other subsets of GCDC (or other TOEFL prompts). Table 8.4 presents the results. Both TextEnty and TextRel

[3]While LLMs can generate plausible rationales for their outputs, these explanations should **not be assumed to faithfully reflect** the underlying mechanisms driving their decisions.

consistently outperform the TextOnly baseline in cross-domain settings, indicating that entity and discourse relation patterns are effective domain-agnostic features for coherence assessment. Moreover, our methods achieve the best performance across all cross-domain experiments, demonstrating the effectiveness of jointly modeling entities and discourse relations.

8.3 Summary

In this chapter, we explore whether combining entity and discourse relation information improves coherence modeling. We propose two novel methods that jointly model entities and discourse relations for coherence assessment. Experiments on three benchmark datasets demonstrate that our approaches consistently outperform strong baselines, emphasizing the value of integrating both features. Additionally, we demonstrate that these features enhance model robustness in scenarios with imbalanced labels and across different domains.

Chapter 9

Conclusions & Future Work

9.1 Conclusions

This thesis explores various linguistically inspired features for coherence modeling from three perspectives: (i) entity-based features (Chapter 4), (ii) discourse relation-based patterns (Chapters 5, 6, and 7), and (iii) a combination of both (Chapter 8).

Part I focuses on entity-based features. We begin by illustrating, through an example, how structural similarity between documents can be potentially useful for coherence modeling. To explicitly model this similarity, we propose a graph-based approach that connects structurally similar documents and leverages Graph Neural Networks to model their connectivity for coherence assessment. Experimental results demonstrate that our method significantly outperforms baselines that do not incorporate such structural information, highlighting its effectiveness. Further analysis reveals that in highly coherent texts, sentences tend to be more densely connected, whereas in less coherent texts, sentences are more isolated.

Part II centers on discourse relation-based patterns. This part comprises three contributions: (1) a novel model to improve implicit discourse relation classification; (2) a detailed analysis of the poor performance of *explicit to implicit discourse classification*; and (3) a discourse relation-enhanced approach for coherence assessment.

- First, we identify one major reason for the limited use of discourse relations in coherence modeling: the poor performance of previous discourse parsers, particularly for implicit relations. To address this, we propose a novel connective-enhanced model inspired by the annotation process of implicit relations in the Penn Discourse Treebank. Our model significantly improves classification performance, achieving over 76% accuracy on top-level relations in PDTB 3.0.

- Next, we systematically investigate an unanswered question in the discourse community: why classifiers trained on explicit examples (with connectives removed) perform poorly in real implicit scenarios. We show that a key reason is a label shift introduced during the construction of the *implicit-like* dataset. Through both manual and empirical analysis, we demonstrate the existence of such a shift and investigate several factors that lead to the occurrence of label shift. We then propose two strategies to mitigate the effects of label shift, showing consistent improvements not only on PDTB data but also on corpora annotated with RST relations.

- Building on the improved discourse relation classifier, we extract discourse relation sequences from coherence corpora and conduct a correlation analysis between relation n-grams and coherence levels. Based on this insight, we propose a fusion Transformer model that integrates both text-based and discourse relation-based features. Our model significantly outperforms competitive baselines, and further analysis shows that discourse relation features enhance model robustness, especially on long documents.

Part III investigates the joint modeling of entities and discourse relations, motivated by the observation that these cues often co-occur and interact in complex ways to establish textual coherence. We propose two approaches: a fusion model and a prompt-based method. Experiments on three benchmark datasets show that both approaches significantly outperform strong baselines, demonstrating the effectiveness of modeling these features jointly. Further analysis indicates that integrating entities and discourse relations facilitates better learning of coherence patterns, mitigates the impact of data imbalance, and improves generalization across domains.

9.2 Future Work

In this section, we outline two potential directions for future research:

- **Designing Linguistically-Inspired Prompts to Guide Large Language Models (LLMs)**.

  Recent advancements in large language models (LLMs), such as GPT-4 (OpenAI et al., 2024) and DeepSeek (DeepSeek-AI et al., 2025), have demonstrated remarkable capabilities and fundamentally reshaped the landscape of natural language processing (NLP). The field is witnessing a paradigm shift from training task-specific models to leveraging general-purpose LLMs via inference through prompt design. Instead of fine-tuning a model for individual tasks, researchers are increasingly focused on

how to formulate effective prompts that can elicit the desired reasoning behavior from LLMs without additional training.

The most prominent approach in LLM prompting is chain-of-thought reasoning (Wei et al., 2022), where complex problems are decomposed into a sequence of simpler reasoning steps. For instance, when asked how much food an adult should consume daily without gaining weight, an LLM might first estimate the average daily energy expenditure, then retrieve the caloric content of various foods per 100 grams, and finally calculate appropriate food portions to remain within the caloric limit.

In this thesis, we have demonstrated that entity-based and discourse relation-based features are highly beneficial for coherence assessment. A promising direction for future research is to design chain-of-thought prompts that guide LLMs to first analyze entity-based connectivity patterns within a document, then examine inter-sentential discourse relations, and finally integrate these cues, along with the document content, to assess overall coherence. This linguistically-informed reasoning process could enhance the interpretability and effectiveness of LLM-based coherence evaluation.

- **Investigating the Benefits of Linguistically-Inspired Features in Extremely Long Documents**

  Our experimental results show that models jointly modeling entities and discourse relations perform particularly well on long documents. However, current LLMs often struggle with very long texts due to input length limitations and degraded performance over extended contexts (Wang et al., 2024; Li et al., 2025). This opens up an interesting avenue for future work: exploring whether linguistically inspired features can help improve the performance of LLMs when handling extremely long documents.

  By explicitly modeling structural and semantic cues, such as entity salience and discourse structure, it may be possible to augment LLMs' capabilities on long-document tasks, either by pre-processing documents into more coherent segments or by designing specialized prompting strategies that incorporate these linguistic signals.

Appendix

A Structural-similarity Enhanced Coherence Modeling

A.1 Subgraph Examples

We show several text pieces with their corresponding constructed subgraphs in Figure 1 (from the GCDC Clinton dataset) and Figure 2 (from the TOEFL P1 dataset). In each example, the corresponding subgraph is shown on the left. Blue boxes indicate the recognized nouns in each sentence, and semantically related nouns across different sentences are connected by directed edges between the boxes. Two sentences are connected if they contain semantically related nouns.

B Annotation-Inspired Implicit Relation Classification

B.1 Experimental Settings

Connectives. For PDTB 2.0, PDTB 3.0, and PCC, we retain only connectives whose frequencies exceed 100, 100, and 5, respectively. In addition, we introduce a default connective, <unk>, into the connective inventory. Any instance containing a connective whose frequency falls below the corresponding threshold is mapped to this default value. As a result, the resulting dataset size is consistent with that used in previous work.

Baselines. We primarily compare our approach with the following baselines:

- **RoBERTa**. This baseline fine-tunes RoBERTa using only the two discourse arguments, (Arg1, Arg2), as input to predict discourse relations. At inference time, the model similarly takes (Arg1, Arg2) as input and outputs a relation prediction.
- **RoBERTaConn**. In this setting, RoBERTa is fine-tuned with both discourse arguments and the explicit connective, i.e., (Arg1, Connective, Arg2), to predict the discourse relation.

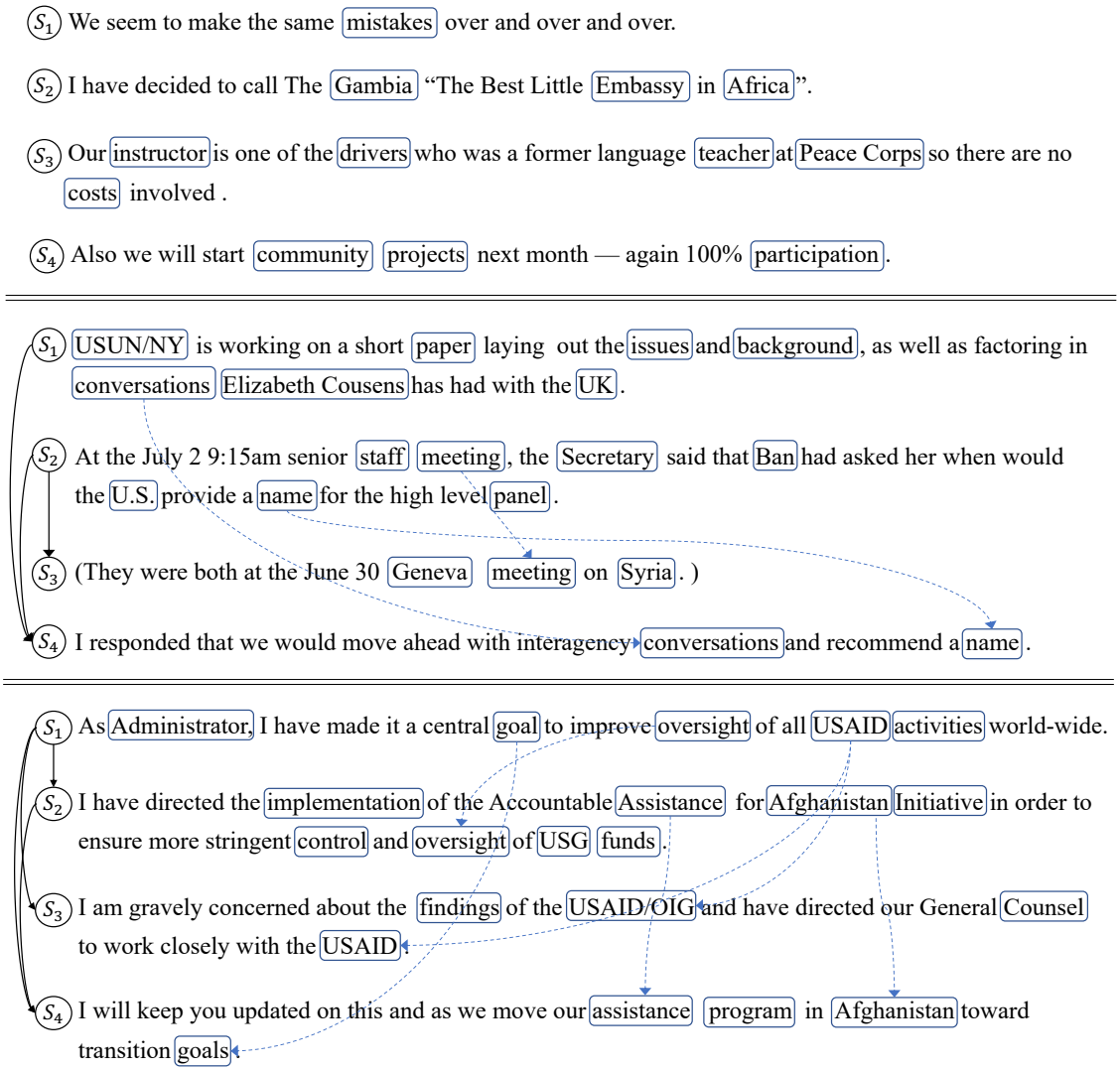


Fig. 1 Three text examples with their corresponding constructed subgraphs from the GCDC Clinton dataset. The subgraph for each text example is displayed to the left of the example.

During evaluation, however, only the two arguments (Arg1, Arg2) are provided as input for relation prediction.

- **Adversarial**. This baseline consists of two RoBERTa-based encoders. The first encoder takes only the two arguments (Arg1, Arg2) as input, while the second encoder additionally incorporates the connective (Arg1, Connective, Arg2). Both encoders are trained to predict discourse relations. An adversarial loss is further introduced to encourage the representations produced by the two encoders to be indistinguishable, such that a discriminator cannot determine which encoder generated a given representation (see Figure 3.7). At test time, only the first encoder is used to predict discourse relations based on (Arg1, Arg2).

- **Multitask**. This baseline uses the two arguments (Arg1, Arg2) as input and employs two classification heads: one for discourse relation prediction and the other for connective prediction (as shown in Figure 3.8). During evaluation, only the discourse relation classifier is used to predict relations from (Arg1, Arg2).
- **Pipeline**. This baseline is a variant of our approach in which the connective generation module and the relation prediction module are trained separately. The connective generation module takes (Arg1, Arg2) as input and generates a plausible connective, while the relation prediction module uses (Arg1, Generated_Connective, Arg2) to predict the discourse relation.

C Explicit to Implicit Discourse Relation Classification

C.1 Manual Analysis

We sample 100 examples from the explicit corpus of PDTB 2.0 and remove connectives from these instances. Then two students[1] familiar with discourse relations are asked to independently annotate these 100 examples with the connective removed, separately. They annotate each instance with one of 12 relations, including *Comparison.Concession*, *Comparison.Contrast*, *Contingency.Cause*, *Contingency.Pragmatic cause*, *Expansion.Conjunction*, *Expansion.Instantiation*, *Expansion.Alternative*, *Expansion.List*, *Expansion.Restatement*, *Temporal.Asynchronous*, *Temporal.Synchrony*, and *NonRel*. If there is disagreement in the annotation of any example, we ask them to discuss and provide a final annotation. The final result can be either just one relation or two different ones (indicating ambiguous examples).

D Joint Modeling of Entities and Discourse Relations

D.1 Prompt with Explanation

In the case studies presented in Section 8.2.3 of Chapter 8, we prompt LLaMA not only to evaluate the coherence level of a given text but also to provide a brief explanation for its judgment. This is achieved by modifying the instruction template used with LLaMA. Figure 3 shows the prompt used in these case studies for Our Method II. Similar prompts are used for TextOnly, TextEnty, and TextRel.

[1]Both students are from the Computational Linguistics department.

Model			GCDC					CoheSentia
			Clinton	Enron	Yahoo	Yelp	Avg	
Prompt	LLaMA zero-shot	TextOnly	56.50	51.00	43.50	47.50	49.63	55.07
		TextEnty	57.50	51.50	45.50	52.00	51.63	56.11
		TextRel	59.50	52.50	49.50	52.50	53.50	56.73
		Our Method II	60.00	53.50	52.50	53.00	54.75	57.56

Table 1 Mean accuracy results of **Llama-3.3-70B** on GCDC and CoheSentia in the **zero-shot setting**.

Models			P1	P2	P3	P4	P5	P6	P7	P8	Avg
Prompt	LLaMA zero-shot	TextOnly	57.25	58.51	54.58	54.67	57.95	56.46	53.62	54.37	55.93
		TextEnty	60.51	58.26	56.30	58.05	58.25	60.42	60.26	56.80	58.61
		TextRel	61.05	59.35	56.88	58.45	59.83	60.21	61.33	56.51	59.20
		Our Method II	62.56	60.24	59.74	59.91	61.35	62.19	61.80	58.23	60.75

Table 2 Mean accuracy results of **Llama-3.3-70B** on TOEFL dataset in the **zero-shot setting**.

D.2 Zero-shot Results Using Llama-3.3-70B

Coherence assessment involves processing entire documents as input, which are typically quite lengthy (see Table 2.6). As a result, training and inference require GPUs with substantial memory capacity. Due to hardware limitations, we employ Llama-3.1-8B as the language model for implementing Method II in Section 8.2 of Chapter 8. We also experimented with the more advanced Llama-3.3-70B model, but it caused out-of-memory errors during fine-tuning. However, our GPU can run Llama-3.3-70B in a zero-shot setting for Method II. Accordingly, we report zero-shot results using Llama-3.3-70B in Tables 1 and 2. As shown, the results are consistent with those obtained using Llama-3.1-8B: incorporating entity and discourse relations improves the model's performance in coherence assessment, and jointly modeling both types of information yields the best results.

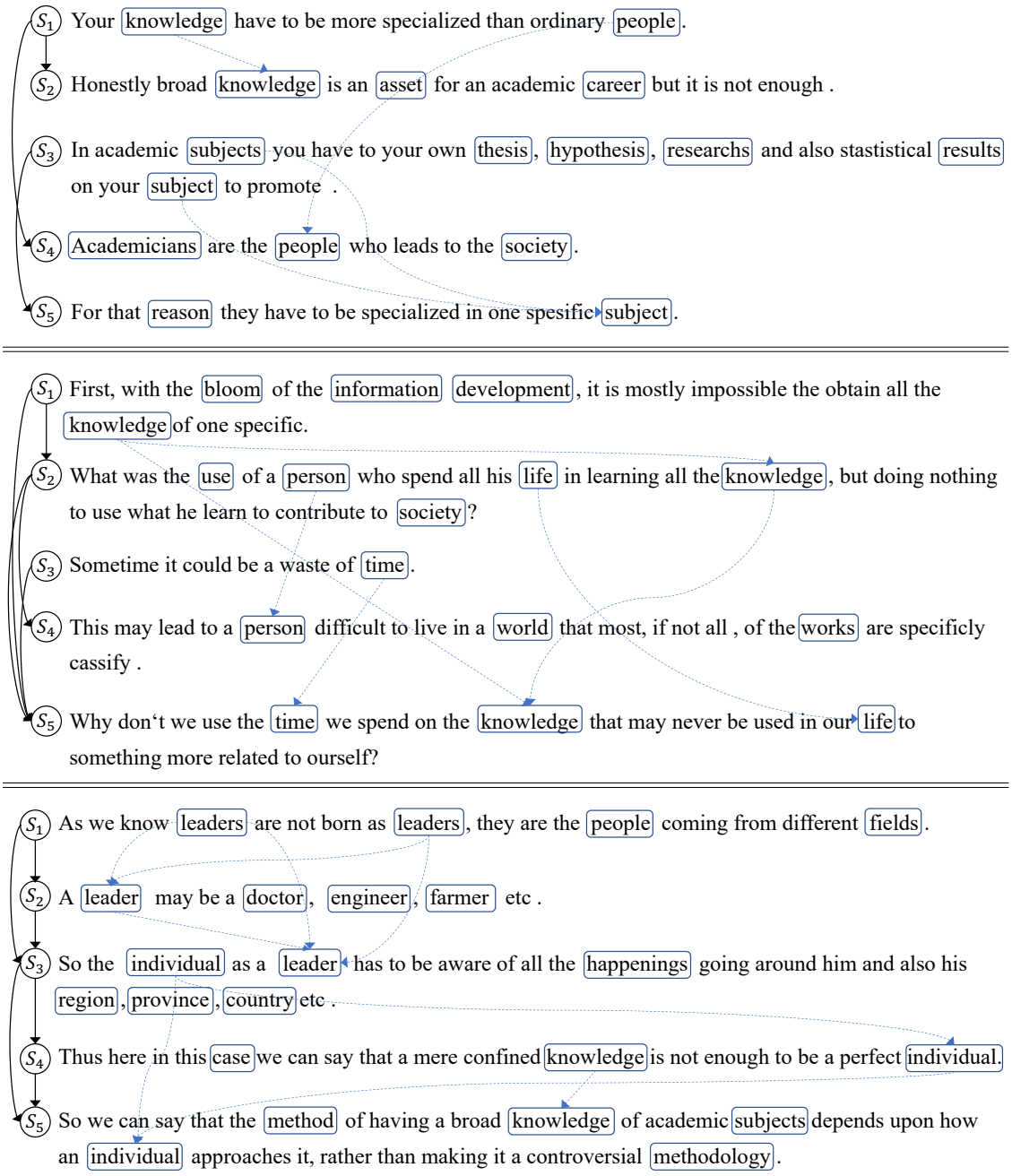


Fig. 2 Three text examples with their corresponding constructed subgraphs from the TOEFL P1 dataset.

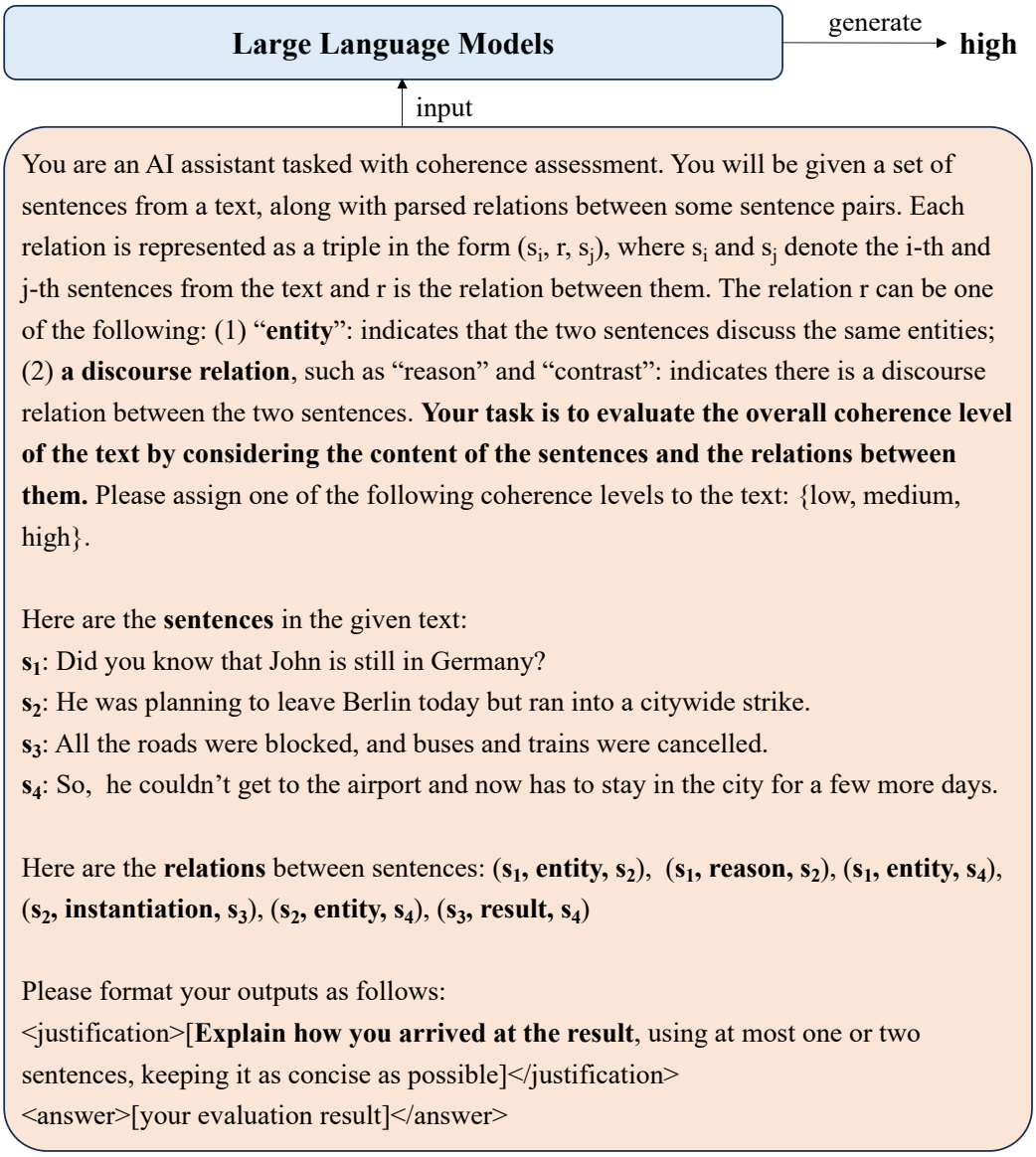


Fig. 3 Prompt with explanation.

E Code and Data Used in this Thesis

The code and data used in this thesis has been published with the following digital object identifiers:

- Wei Liu (2025). Source code and data for the PhD Thesis "Linguistically-Inspired Neural Coherence Modeling". DOI: 10.11588/data/ZBNUCG, URL: https://doi.org/10.11588/data/ZBNUCG

List of Figures

1.1 Overview of this thesis. It comprises three components: entity-based coherence modeling, discourse relation-based coherence modeling, and a combined approach that integrates both types of features. 7

2.1 A rhetorical structure representation of the text in Example (2.9). 16
2.2 Two forms of the task of Coherence Modeling. 21
2.3 An overview of the Transformer. 30
2.4 An overview of Message Passing. 33
2.5 A schematic comparison between GPT (decoder-only), BERT (encoder-only) and BART (encoder-decoder). 35

3.1 An illustrative example of the entity grid. Given a text, all entities and their grammatical roles, subject (S), object (O), or other (X), are identified. The text is then represented as a two-dimensional grid, where each row corresponds to a sentence and each column to an entity. Each cell (s_i, e_j) in the grid denotes the grammatical role of the j-th entity e_j in the i-th sentence s_i. 42
3.2 An illustrative example of the entity graph. Given a text with identified entities, a bipartite graph is constructed linking sentences and entities. This bipartite graph is subsequently transformed into a sentence graph via a one-mode projection onto the sentence nodes. 44
3.3 Neural entity grid model proposed by Tien Nguyen and Joty (2017). The model is trained using a pairwise ranking approach with shared parameters for positive and negative documents. 45

3.4 An overview of the model proposed by Jeon and Strube (2020a). The approach approximates Centering Theory to track shifts in discourse focus across segments and constructs hierarchical discourse structures that represent relationships between different segments of the text (see the Discourse Segment Parser). These structures are then utilized by a structure-aware Transformer for coherence assessment. 46
3.5 An illustrative example of the method proposed by Lin et al. (2011). Given a text, the method first identifies all discourse arguments and determines the discourse relations between each pair of arguments. It then extracts, for each sentence, the discourse arguments involving a given term, along with their associated discourse relations. 48
3.6 The overall idea of Long and Webber (2022). Given an anchor instance, positive and negative examples are identified within a training batch according to the PDTB sense hierarchy. The contrastive objective encourages instances sharing higher-level discourse senses to be closer in the representation space, while pushing apart instances belonging to different branches of the sense hierarchy. 51
3.7 The architecture of the adversarial model proposed by Qin et al. (2017). The framework contains three main components: 1) an implicit relation network *i*-encoder over raw sentence arguments, 2) a connective-augmented relation network *a*-encoder whose inputs are augmented with implicit connectives, and 3) a discriminator distinguishing between the features by the two networks. The features are fed to the final classifier for relation classification. The discriminator and *i*-encoder form an adversarial pair for feature imitation. At test time, the implicit network *i*-encoder with the classifier is used for prediction. 52
3.8 Overview of the multi-task model proposed by Kishimoto et al. (2020). The input is an implicit argument pair randomly selected from the training data, where annotators have provided an implicit connective for each pair. BERT is trained to predict the implicit connective and the discourse relation. . . . 53

4.1 An example of two highly coherent texts exhibiting similar entity connectivity structures. Recognized nouns are highlighted in bold. 58

4.2 Overview of the proposed graph-based approach. Our method identifies the graph structure of each document, converts the graph into a set of subgraphs, constructs a corpus-level graph based on the shared subgraphs between structurally similar documents, and finally encodes those connections using a Graph Convolutional Network (GCN). For simplicity, we illustrate this process with only three documents and five subgraphs, limiting the number of sentences per document. s_u, d_i, and g_j denote the u-th sentence in a document, the i-th document in the training corpus, and the j-th defined subgraph. 59
4.3 An example of subgraphs, in which graph (b) and graph (c) are 3-node subgraphs of graph (a). 61
4.4 Predicted label distribution in TOEFL P1 dataset. 69
4.5 Accuracy against essay length. 70
4.6 The top two most positively correlated subgraphs for each coherence level on the GCDC Clinton and TOEFL P1. r denotes the correlation coefficient value, and p is the p_value ($p < 0.05$ means statistically significant). 71

5.1 An example illustrating the two-step annotation procedure for implicit discourse relations in the Penn Discourse Treebank (PDTB) 2.0. 75
5.2 An overview of the proposed approach. The left part is the connective generation module, which generates a connective at the masked position between arguments (Arg1, Arg2). The right part is the relation classification module, which predicts the relation based on both arguments and the generated connective. The two modules share the embedding layer and transformer blocks, and the entire model is trained in an end-to-end manner. 76
5.3 Level-1 classification results on PDTB 2.0 (Ji split) when annotated connectives are fed to connective-enhanced models. "Increase" denotes performance gain compared to the model with default settings ("Base"). 84
5.4 Level-1 classification results on PDTB 2.0 (Ji split). "Remove" denotes the generated connectives are removed from the original model ("Base"). . . . 85

6.1 An illustration of the process for training an implicit discourse relation classifier using explicit relation examples. 92
6.2 The RoBERTa classifier used in our analyses. 93
6.3 Performance comparison of implicit discourse relation classification between the classifier trained on explicit examples (Exp2Imp) and one trained on real implicit examples (Imp2Imp), using the PDTB 2.0 dataset. 93

6.4 Examples of suffering and not suffering from label shift. 95
6.5 Different cases suffering from label shift. 96
6.6 Percentage of examples in **Exp**licit and **Imp**licit corpora that receive the same and different predictions when the input contains and not contains a connective. 97
6.7 Visualization of example representations in PDTB 2.0 with and without connectives. 98
6.8 Feature Importance of the XGBoost Model in predicting the label shift metric on PDTB 2.0 and 3.0. 101
6.9 The architecture of the joint learning model. 103

7.1 A coherent text with discourse relations. 109
7.2 Two ways to combine text- and relation-based features: concatenation vs. fusion. 115
7.3 Converting original sentences and parsed relations (a) into a flat sentence-relation structure (b), where start_pos and end_pos denote the start and end positions of the node in the original sentence sequence. 116
7.4 Fusion Transformer. 117
7.5 The template used for the Llama-Prompt baseline 120
7.6 The template used for the GPT4-Prompt baseline. 120
7.7 Accuracy against text length. 123

8.1 An example of a coherent text, whose coherence should be explained using both entities and discourse relations. We bold the interlinked entities in the text and show the discourse relations between sentences. 125
8.2 Sentences (in Figure 8.1) linked by entities and discourse relations. 127
8.3 The sentences, entities, and discourse relations in Figure 8.2 are organized into a flat structure, in which each element is assigned a two-dimensional position indicating its start and end within the original sentence sequence. This flat input is then processed by a fusion Transformer. 128
8.4 Illustration of our second approach. We use natural language to describe the relationships between sentences, entities, and discourse relations in Figure 8.2, presenting the graph structure in a concise and intuitive way. We then instruct LLMs to consider these elements for coherence assessment. 130

8.5 Two examples (truncated) showing how entities and discourse relations aid coherence assessment. Both texts are labeled as low coherence. We use a zero-shot prompt setting, and the "explanation" refers to LLaMA's brief justification for its prediction. 136

1 Three text examples with their corresponding constructed subgraphs from the GCDC Clinton dataset. The subgraph for each text example is displayed to the left of the example. 144
2 Three text examples with their corresponding constructed subgraphs from the TOEFL P1 dataset. 147
3 Prompt with explanation. 148

List of Tables

2.1 Four types of transitions in Certering Theory, from Brennan et al. (1987). . 14
2.2 PDTB 3.0 Sense Hierarchy. The leftmost column contains the Level-1 senses, and the middle column, the Level-2 senses. For asymmetric relations, Level-3 senses are located in the rightmost column. 19
2.3 An example of shuffle test from Barzilay and Lapata (2008), where the first is the original document and the second is the shuffled one. 22
2.4 Statistics of the shuffle test dataset created from the Wall Stree Journal portion of Penn TreeBank. 23
2.5 Topic prompts in the TOEFL dataset. 24
2.6 The statistics of the TOEFL dataset. 24
2.7 The statistics of the GCDC dataset. 25
2.8 Label distribution in TOEFL and GCDC (%). 25
2.9 The top-level (L1) and second-level (L2) discourse relations in PDTB 2.0 and PDTB 3.0 commonly used in the literature. 27
2.10 The statistics of PDTB 2.0 and PDTB 3.0. 28
2.11 The statistics of the GUM corpus. 28
2.12 Examples of Pre-trained Language Models (PLMs) and Large Language Models (LLMs) used in this thesis. 36
2.13 Comparison of model adaptation strategies, including fine-tuning, zero-shot prompting, and in-context learning. 37

4.1 Mean accuracy (standard deviation) on GCDC. * indicates that our model significantly outperforms the XLNet+DNN baseline ($p < 0.05$). 67
4.2 Mean accuracy (standard deviation) on TOEFL. * indicates that our model significantly outperforms the XLNet+DNN baseline ($p < 0.05$). 68
4.3 Ablation study for different edges on the GCDC Clinton and TOEFL P1 dataset. 70

5.1 Second-level (L2) relations of PCC used in our experiments. 80
5.2 Results on PDTB **2.0**. Subscripts are the standard deviation of the mean performance. 82
5.3 Results on PDTB **3.0**. 83
5.4 Results on **PCC**. 83
5.5 Level-1 classification results on PDTB 2.0 (Ji split) when connectives are correctly and incorrectly generated (or predicted). "+" and "-" denote the increase and decrease compared to the RoBERTa baseline (Base). 86
5.6 F1 results for each second-level relation of PDTB 2.0. 87
5.7 Ablation study for **S**cheduled **S**ampling and connective generation loss $\mathcal{L}_{Conn}$. 88

6.1 Pearson correlation between each individual factor and the label shift metric. 100
6.2 Results on **PDTB 2.0** (with standard deviation). E2I-Entire is the typical setting for explicit to implicit discourse relation recognition, serving as the baseline, and I2I-Entire is the upper bound for implicit relation classification. Our two strategies can effectively close the gap between the baseline and the upper bound. 105
6.3 Results on **PDTB 3.0** (with standard deviation). 106
6.4 Results on the RST GUM corpus. 107

7.1 Explicit and Implicit relations used in this study and their distribution in the training set of PDTB 3.0. 111
7.2 Correlation between discourse relation N-gram patterns and coherence levels. Only the top four patterns with the highest absolute correlation coefficients are shown. 112
7.3 Correlation between discourse relation N-gram patterns and coherence levels. Only the top four patterns with the highest absolute correlation coefficients are shown. 113
7.4 The distribution of discourse relations parsed from GCDC Enron and TOEFL P1. 114
7.5 The performance (with std) of BiLSTM classifier when using text, relation sequence, and shuffled relation sequence as input, respectively. 114
7.6 Mean accuracy results (with std) on the GCDC dataset. 121
7.7 Mean accuracy results (with std) on the TOEFL dataset. 122
7.8 Cross-domain accuracy of models. 124
7.9 Ablation study for visibility matrix (VM) and position-aware attention (PAA) in our method. 124

8.1 Mean accuracy results (with std) on GCDC and CoheSentia. 133
8.2 Mean accuracy results (with std) on TOEFL dataset. 134
8.3 Accuracy results for each coherence level on TOEFL P5. Range indicates the difference between the highest and lowest values. 135
8.4 Accuracy of models in a cross-domain setting. 137

1 Mean accuracy results of **Llama-3.3-70B** on GCDC and CoheSentia in the **zero-shot setting**. 146
2 Mean accuracy results of **Llama-3.3-70B** on TOEFL dataset in the **zero-shot setting**. 146

References

Abhishek, T., Rawat, D., Gupta, M., and Varma, V. (2022). Transformer models for text coherence assessment.

Asher, N. and Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Barzilay, R. and Lapata, M. (2005). Modeling local coherence: An entity-based approach. In Knight, K., Ng, H. T., and Oflazer, K., editors, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148, Ann Arbor, Michigan. Association for Computational Linguistics.

Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28.

Biran, O. and McKeown, K. (2015). Discourse planning with an n-gram model of relations. In Màrquez, L., Callison-Burch, C., and Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1973–1977, Lisbon, Portugal. Association for Computational Linguistics.

Blair-Goldensohn, S., McKeown, K., and Rambow, O. (2007). Building and refining rhetorical-semantic relation models. In Sidner, C., Schultz, T., Stone, M., and Zhai, C., editors, *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 428–435, Rochester, New York. Association for Computational Linguistics.

Blanchard, D., Tetreault, J., Higgins, D., Cahill, A., and Chodorow, M. (2014). ETS corpus of non-native written english. Web Download, Linguistic Data Consortium (LDC2014T06).

Blühdorn, H. (2017). Subordination and coordination in syntax, semantics and discourse. evidence from the study of connectives. In Fabricius-Hansen, C. and Ramm, W., editors, *'Subordination' versus 'Coordination' in Sentence and Text. A cross-linguistic perspective*, number 98 in Studies in Language Companion Series, pages 59 – 85.

Bourgonje, P. (2021). *Shallow Discourse Parsing for German*. Doctoral Thesis, Universität Potsdam.

Bourgonje, P. and Stede, M. (2020). The Potsdam commentary corpus 2.2: Extending annotations for shallow discourse parsing. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1061–1066, Marseille, France. European Language Resources Association.

Brennan, S. E., Friedman, M. W., and Pollard, C. J. (1987). A centering approach to pronouns. In *25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162, Stanford, California, USA. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Burstein, J., Tetreault, J., and Andreyev, S. (2010). Using entity-based features to model coherence in student essays. In Kaplan, R., Burstein, J., Harper, M., and Penn, G., editors, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 681–684, Los Angeles, California. Association for Computational Linguistics.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Carlson, L., Marcu, D., and Okurovsky, M. E. (2001). Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Chaturvedi, S., Peng, H., and Roth, D. (2017). Story comprehension for predicting what happens next. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614, Copenhagen, Denmark. Association for Computational Linguistics.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

Clark, H. H. (1975). Bridging. In Nash-Webber, B. and Schank, R., editors, *Theoretical Issues in Natural Language Processing*.

Crothers, E. J. (1978). Inference and coherence. *Discourse Processes*, 1:51–71.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Danes, F. (1974). Functional sentence perspective and the organization of the text. *Papers on functional sentence perspective*, 23:106–128.

DeepMind, G. (2024). Gemini 1.5: Unlocking multimodal understanding across 10m context length. *arXiv preprint arXiv:2403.05530*.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu, J., Li, J., Song, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang, T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Song, X., Shan, X., Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhu, Y. X., Zhang, Y., Xu, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y., Yan, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Huang, Z., Zhang, Z., Xie, Z., Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu, Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Gao, Z., and Pan, Z. (2025). Deepseek-v3 technical report.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dong, F., Zhang, Y., and Yang, J. (2017). Attention-based recurrent convolutional neural network for automatic essay scoring. In Levy, R. and Specia, L., editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Elsner, M. and Charniak, E. (2011). Extending the entity grid with entity-specific features. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129, Portland, Oregon, USA. Association for Computational Linguistics.

Farag, Y. and Yannakoudakis, H. (2019). Multi-task learning for coherence modeling. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 629–639, Florence, Italy. Association for Computational Linguistics.

Farag, Y., Yannakoudakis, H., and Briscoe, T. (2018). Neural automated essay scoring and coherence modeling for adversarially crafted input. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 263–271, New Orleans, Louisiana. Association for Computational Linguistics.

Feng, V. W. and Hirst, G. (2014). A linear-time bottom-up discourse parser with constraints and post-editing. In Toutanova, K. and Wu, H., editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.

Feng, V. W., Lin, Z., and Hirst, G. (2014). The impact of deep hierarchical discourse structures in the evaluation of text coherence. In Tsujii, J. and Hajic, J., editors, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 940–949, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Filippova, K. and Strube, M. (2007). Extending the entity-grid coherence model to semantically related entities. In Busemann, S., editor, *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 139–142, Saarbrücken, Germany. DFKI GmbH.

Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse processes*, 25(2-3):285–307.

Ghazvininejad, M., Shi, X., Choi, Y., and Knight, K. (2016). Generating topical poetry. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191, Austin, Texas. Association for Computational Linguistics.

Graff, D. and Cieri, C. (2003). English gigaword. Web Download. LDC Catalog No. LDC2003T05.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis,

I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. (2024). The llama 3 herd of models.

Grosz, B. J., Joshi, A. K., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Guinaudeau, C. and Strube, M. (2013). Graph-based local coherence modeling. In Schuetze, H., Fung, P., and Poesio, M., editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103, Sofia, Bulgaria. Association for Computational Linguistics.

Guo, F., He, R., Jin, D., Dang, J., Wang, L., and Li, X. (2018). Implicit discourse relation recognition using neural tensor network with interactive attention and sparse learning. In Bender, E. M., Derczynski, L., and Isabelle, P., editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 547–558, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Guo, Z. and Nguyen, M. L. (2020). Document-level neural machine translation using BERT as context encoder. In Shmueli, B. and Huang, Y. J., editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 101–107, Suzhou, China. Association for Computational Linguistics.

Halliday, M. A. K. and Hasan, R. (1976). *Cohesion in English*. Longman.

Hamner, B., Morgan, J., lynnvandev, Shermis, M., and Ark, T. V. (2012). The hewlett foundation: Automated essay scoring. https://kaggle.com/competitions/asap-aes. Kaggle.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

Hernault, H., Prendinger, H., duVerle, D. A., and Ishizuka, M. (2010). Hilda: A discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hou, Y., Markert, K., and Strube, M. (2018). Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Huang, H.-P. and Li, J. J. (2019). Unsupervised adversarial domain adaptation for implicit discourse relation classification. In Bansal, M. and Villavicencio, A., editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 686–695, Hong Kong, China. Association for Computational Linguistics.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017*.

Jeon, S. and Strube, M. (2020a). Centering-based neural coherence modeling with hierarchical discourse segments. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7458–7472, Online. Association for Computational Linguistics.

Jeon, S. and Strube, M. (2020b). Incremental neural lexical coherence modeling. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6752–6758, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jeon, S. and Strube, M. (2021). Countering the influence of essay length in neural essay scoring. In Moosavi, N. S., Gurevych, I., Fan, A., Wolf, T., Hou, Y., Marasović, A., and Ravi, S., editors, *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 32–38, Virtual. Association for Computational Linguistics.

Jeon, S. and Strube, M. (2022). Entity-based neural local coherence modeling. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7787–7805, Dublin, Ireland. Association for Computational Linguistics.

Ji, H., Yang, C., Shi, C., and Li, P. (2021). Heterogeneous graph neural network with distance encoding. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1138–1143, Los Alamitos, CA, USA. IEEE Computer Society.

Ji, Y. and Eisenstein, J. (2014). Representation learning for text-level discourse parsing. In Toutanova, K. and Wu, H., editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Ji, Y. and Eisenstein, J. (2015). One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344.

Ji, Y., Haffari, G., and Eisenstein, J. (2016). A latent variable recurrent neural network for discourse-driven language models. In Knight, K., Nenkova, A., and Rambow, O., editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 332–342, San Diego, California. Association for Computational Linguistics.

Ji, Y., Zhang, G., and Eisenstein, J. (2015). Closing the gap: Domain adaptation from explicit to implicit discourse relations. In Màrquez, L., Callison-Burch, C., and Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2219–2224, Lisbon, Portugal. Association for Computational Linguistics.

Jiang, C., Qian, T., Chen, Z., Tang, K., Zhan, S., and Zhan, T. (2021). Generating pseudo connectives with mlms for implicit discourse relation recognition. In *PRICAI 2021: Trends in Artificial Intelligence*, volume 13034 of *Lecture Notes in Computer Science*, pages 114–127. Springer.

Joty, S., Mohiuddin, M. T., and Tien Nguyen, D. (2018). Coherence modeling of asynchronous conversations: A neural entity grid approach. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568, Melbourne, Australia. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Third edition draft edition.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Ke, Z. and Ng, V. (2019). Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6300–6308.

Kiela, D. and Clark, S. (2014). A systematic study of semantic vector space model parameters. In Allauzen, A., Bernardi, R., Grefenstette, E., Larochelle, H., Manning, C., and Yih, S. W.-t., editors, *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden. Association for Computational Linguistics.

Kim, N., Feng, S., Gunasekara, C., and Lastras, L. (2020). Implicit discourse relation classification: We need to talk about evaluation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5404–5414, Online. Association for Computational Linguistics.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Kishimoto, Y., Murawaki, Y., and Kurohashi, S. (2018). A knowledge-augmented neural network model for implicit discourse relation classification. In Bender, E. M., Derczynski, L., and Isabelle, P., editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 584–595, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kishimoto, Y., Murawaki, Y., and Kurohashi, S. (2020). Adapting BERT to implicit discourse relation classification with a focus on discourse connectives. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1152–1158, Marseille, France. European Language Resources Association.

Knaebel, R. (2021). discopy: A neural system for shallow discourse parsing. In Braud, C., Hardmeier, C., Li, J. J., Louis, A., Strube, M., and Zeldes, A., editors, *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 128–133, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Knott, A., Oberlander, J., O'Donnell, M., and Mellish, C. (2000). Beyond elaboration: The interaction of relations and focus in coherent text. In Sanders, T., Schilperoord, J., and Spooren, W., editors, *Text Representation: Linguistic and Psycholinguistic Aspects, chapter 7*, Human Cognitive Processing, pages 181–196. John Benjamins Publishing Company.

Kobayashi, N., Hirao, T., Kamigaito, H., Okumura, M., and Nagata, M. (2022). A simple and strong baseline for end-to-end neural RST-style discourse parsing. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6725–6737, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22170–22183.

Kondor, R., Shervashidze, N., and Borgwardt, K. M. (2009). The graphlet spectrum. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 529–536.

Kurfalı, M. and Östling, R. (2021). Let's be explicit about that: Distant supervision for implicit discourse relation classification via connective prediction. In Roth, M., Tsarfaty, R.,

and Goldberg, Y., editors, *Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language*, pages 1–10, Online. Association for Computational Linguistics.

Kwon, M., Xie, S. M., Bullard, K., and Sadigh, D. (2023). Reward design with language models. In *The Eleventh International Conference on Learning Representations*.

Laban, P., Dai, L., Bandarkar, L., and Hearst, M. A. (2021). Can transformer models measure coherence in text: Re-thinking the shuffle test. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1058–1064, Online. Association for Computational Linguistics.

Lai, A. and Tetreault, J. (2018). Discourse coherence in the wild: A dataset, evaluation and methods. In Komatani, K., Litman, D., Yu, K., Papangelis, A., Cavedon, L., and Nakano, M., editors, *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223, Melbourne, Australia. Association for Computational Linguistics.

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 545–552, Sapporo, Japan. Association for Computational Linguistics.

Lapata, M. and Barzilay, R. (2005). Automatic evaluation of text coherence: Models and representations. In *IJCAI*, volume 5, pages 1085–1090.

Lapata, M. and Lascarides, A. (2004). Inferring sentence-internal temporal relations. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 153–160, Boston, Massachusetts, USA. Association for Computational Linguistics.

Lei, W., Wang, X., Liu, M., Ilievski, I., He, X., and Kan, M.-Y. (2017). Swim: A simple word interaction model for implicit discourse relation recognition. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4026–4032.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, J. and Hovy, E. (2014). A model of coherence based on distributed sentence representation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048, Doha, Qatar. Association for Computational Linguistics.

Li, J. and Jurafsky, D. (2017). Neural net models of open-domain discourse coherence. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209, Copenhagen, Denmark. Association for Computational Linguistics.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3538–3545.

Li, T., Zhang, G., Do, Q. D., Yue, X., and Chen, W. (2025). Long-context LLMs struggle with long in-context learning. *Transactions on Machine Learning Research*.

Li, X., Yan, H., Qiu, X., and Huang, X. (2020). FLAT: Chinese NER using flat-lattice transformer. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online. Association for Computational Linguistics.

Li, Z., Zhou, Q., Li, C., Xu, K., and Cao, Y. (2021). Improving BERT with syntax-aware local attention. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 645–653, Online. Association for Computational Linguistics.

Lin, X., Joty, S., Jwalapuram, P., and Bari, M. S. (2019). A unified linear-time framework for sentence-level discourse parsing. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200, Florence, Italy. Association for Computational Linguistics.

Lin, Z., Kan, M.-Y., and Ng, H. T. (2009). Recognizing implicit discourse relations in the Penn Discourse Treebank. In Koehn, P. and Mihalcea, R., editors, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351, Singapore. Association for Computational Linguistics.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2011). Automatically evaluating text coherence using discourse relations. In Lin, D., Matsumoto, Y., and Mihalcea, R., editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 997–1006, Portland, Oregon, USA. Association for Computational Linguistics.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2014). A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Liu, W., Fu, X., and Strube, M. (2023a). Modeling structural similarities between documents for coherence assessment with graph convolutional networks. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7792–7808, Toronto, Canada. Association for Computational Linguistics.

Liu, W. and Strube, M. (2023). Annotation-inspired implicit discourse relation classification with auxiliary discourse connective generation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15696–15712, Toronto, Canada. Association for Computational Linguistics.

Liu, W., Wan, S., and Strube, M. (2024). What causes the failure of explicit to implicit discourse relation recognition? In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2738–2753, Mexico City, Mexico. Association for Computational Linguistics.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. (2020a). K-BERT: enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908. AAAI Press.

Liu, X., Ou, J., Song, Y., and Jiang, X. (2020b). On the importance of word and sentence representation learning in implicit discourse relation classification. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3830–3836. International Joint Conferences on Artificial Intelligence Organization. Main track.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Liu, Y. J., Aoyama, T., and Zeldes, A. (2023b). What's hard in English RST parsing? predictive models for error analysis. In Stoyanchev, S., Joty, S., Schlangen, D., Dusek, O., Kennington, C., and Alikhani, M., editors, *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–42, Prague, Czechia. Association for Computational Linguistics.

Long, W. and Webber, B. (2022). Facilitating contrastive learning of discourse relational senses by exploiting the hierarchy of sense relations. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10704–10716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maekawa, A., Hirao, T., Kamigaito, H., and Okumura, M. (2024). Can we obtain significant success in RST discourse parsing by using large language models? In Graham, Y. and Purver, M., editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2803–2815, St. Julian's, Malta. Association for Computational Linguistics.

Maimon, A. and Tsarfaty, R. (2023). COHESENTIA: A novel benchmark of incremental versus holistic assessment of coherence in generated texts. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5328–5343, Singapore. Association for Computational Linguistics.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

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

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

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Mesgar, M., Ribeiro, L. F. R., and Gurevych, I. (2021). A neural graph-based local coherence model. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2316–2321, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mesgar, M. and Strube, M. (2015). Graph-based coherence modeling for assessing readability. In Palmer, M., Boleda, G., and Rosso, P., editors, *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 309–318, Denver, Colorado. Association for Computational Linguistics.

Mesgar, M. and Strube, M. (2016). Lexical coherence graph modeling using word embeddings. In Knight, K., Nenkova, A., and Rambow, O., editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1414–1423, San Diego, California. Association for Computational Linguistics.

Mesgar, M. and Strube, M. (2018). A neural local coherence model for text quality assessment. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339, Brussels, Belgium. Association for Computational Linguistics.

Mihaylov, T. and Frank, A. (2019). Discourse-aware semantic self-attention for narrative reading comprehension. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2541–2552, Hong Kong, China. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Milajevs, D., Sadrzadeh, M., and Purver, M. (2016). Robust co-occurrence quantification for lexical distributional semantics. In He, H., Lei, T., and Roberts, W., editors, *Proceedings of the ACL 2016 Student Research Workshop*, pages 58–64, Berlin, Germany. Association for Computational Linguistics.

Miltsakaki, E. and Kukich, K. (2000). The role of centering theory's rough-shift in the teaching and evaluation of writing skills. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 408–415, Hong Kong. Association for Computational Linguistics.

Miltsakaki, E., Prasad, R., Joshi, A., and Webber, B. (2004). The Penn Discourse Treebank. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., and Silva, R., editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mizumoto, A. and Eguchi, M. (2023). Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.

Mohiuddin, T., Jwalapuram, P., Lin, X., and Joty, S. (2021). Rethinking coherence modeling: Synthetic vs. downstream tasks. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3528–3539, Online. Association for Computational Linguistics.

Moon, H. C., Mohiuddin, T., Joty, S., and Xu, C. (2019). A unified neural coherence model. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

Nguyen, T.-T., Nguyen, X.-P., Joty, S., and Li, X. (2021). RST parsing from scratch. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1613–1625, Online. Association for Computational Linguistics.

OpenAI (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar,

N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). Gpt-4 technical report.

Parmar, M., Deilamsalehy, H., Dernoncourt, F., Yoon, S., Rossi, R. A., and Bui, T. (2024). Towards enhancing coherence in extractive summarization: Dataset and experiments with LLMs. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19810–19820, Miami, Florida, USA. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In Augenstein, I., Gella, S., Ruder, S., Kann, K., Can, B., Welbl, J., Conneau, A., Ren, X., and Rei, M., editors, *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Pitler, E., Louis, A., and Nenkova, A. (2009). Automatic sense prediction for implicit discourse relations in text. In Su, K.-Y., Su, J., Wiebe, J., and Li, H., editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 683–691, Suntec, Singapore. Association for Computational Linguistics.

Pitler, E. and Nenkova, A. (2009). Using syntax to disambiguate explicit discourse connectives in text. In Su, K.-Y., Su, J., Wiebe, J., and Li, H., editors, *Proceedings of the*

ACL-IJCNLP 2009 Conference Short Papers, pages 13–16, Suntec, Singapore. Association for Computational Linguistics.

Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., and Joshi, A. (2008). Easily identifiable discourse relations. In Scott, D. and Uszkoreit, H., editors, *Coling 2008: Companion volume: Posters*, pages 87–90, Manchester, UK. Coling 2008 Organizing Committee.

Poesio, M., Stevenson, R., Di Eugenio, B., and Hitzeman, J. (2004). Centering: A parametric theory and its instantiations. *Computational Linguistics*, 30(3):309–363.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The Penn Discourse TreeBank 2.0. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A. K., Robaldo, L., and Webber, B. L. (2006). The penn discourse treebank 2.0 annotation manual.

Prasad, R., Miltsakaki, E., Joshi, A., and Webber, B. (2004). Annotation and data mining of the Penn Discourse TreeBank. In *Proceedings of the Workshop on Discourse Annotation*, pages 88–95, Barcelona, Spain. Association for Computational Linguistics.

Prasad, R., Webber, B., and Lee, A. (2018). Discourse annotation in the PDTB: The next generation. In Bunt, H., editor, *Proceedings of the 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 87–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. In Celikyilmaz, A. and Wen, T.-H., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Qin, L., Zhang, Z., and Zhao, H. (2016a). Implicit discourse relation recognition with context-aware character-enhanced embeddings. In Matsumoto, Y. and Prasad, R., editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1914–1924, Osaka, Japan. The COLING 2016 Organizing Committee.

Qin, L., Zhang, Z., and Zhao, H. (2016b). A stacking gated neural architecture for implicit discourse relation classification. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2263–2270, Austin, Texas. Association for Computational Linguistics.

Qin, L., Zhang, Z., Zhao, H., Hu, Z., and Xing, E. (2017). Adversarial connective-exploiting networks for implicit discourse relation classification. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1017, Vancouver, Canada. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Technical Report.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2016). Sequence level training with recurrent neural networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016*.

Reinhart, T. (1980). Conditions for text coherence. *Poetics Today*, 1(4):161–180.

Rodriguez-Garcia, R., Reyes Montesinos, J., Fraile-Hernandez, J. M., and Peñas, A. (2024). HAMiSoN-ensemble at ClimateActivism 2024: Ensemble of RoBERTa, llama 2, and multi-task for stance detection. In Hürriyetoğlu, A., Tanev, H., Thapa, S., and Uludoğan, G., editors, *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 118–124, St. Julians, Malta. Association for Computational Linguistics.

Rohde, H., Johnson, A., Schneider, N., and Webber, B. (2018). Discourse coherence: Concurrent explicit and implicit relations. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2257–2267, Melbourne, Australia. Association for Computational Linguistics.

Rutherford, A. and Xue, N. (2014). Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In Wintner, S., Goldwater, S., and Riezler, S., editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–654, Gothenburg, Sweden. Association for Computational Linguistics.

Saito, M., Yamamoto, K., and Sekine, S. (2006). Using phrasal patterns to identify discourse relations. In Moore, R. C., Bilmes, J., Chu-Carroll, J., and Sanderson, M., editors, *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 133–136, New York City, USA. Association for Computational Linguistics.

Salle, A. and Villavicencio, A. (2019). Why so down? The role of negative (and positive) pointwise mutual information in distributional semantics. *arXiv preprint arXiv:1908.06941*.

Sanders, T. J., Demberg, V., Hoek, J., Scholman, M. C., Asr, F. T., Zufferey, S., and Evers-Vermeul, J. (2021). Unifying dimensions in coherence relations: How various annotation frameworks are related. *Corpus Linguistics and Linguistic Theory*, 17(1):1–71.

Sanders, T. J. M., Spooren, W. P. M., and Noordman, L. G. M. (1992). Toward a taxonomy of coherence relations. *Discourse Processes*, 15(1):1–35.

Sarzhoska-Georgievska, E. (2016). Coherence: Implications for teaching writing. *English Studies at NBU*, 2(1):17–30.

Schwarz, M. (2001). Establishing coherence in text. conceptual continuity and text-world models. *Logos and Language*, 2(1):15–24.

Shang, E., Liu, X., Wang, H., Rong, Y., and Liu, Y. (2019). Research on the application of artificial intelligence and distributed parallel computing in archives classification. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 1267–1271.

Shen, A., Mistica, M., Salehi, B., Li, H., Baldwin, T., and Qi, J. (2021). Evaluating document coherence modeling. *Transactions of the Association for Computational Linguistics*, 9:621–640.

Sheng, Z., Zhang, T., Jiang, C., and Kang, D. (2024). Bbscore: A brownian bridge based metric for assessing text coherence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14937–14945.

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. (2009). Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR.

Shi, J., Ding, X., Du, L., Liu, T., and Qin, B. (2021). Neural natural logic inference for interpretable question answering. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3673–3684, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shi, W. and Demberg, V. (2019a). Learning to explicitate connectives with Seq2Seq network for implicit discourse relation classification. In Dobnik, S., Chatzikyriakidis, S., and Demberg, V., editors, *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 188–199, Gothenburg, Sweden. Association for Computational Linguistics.

Shi, W. and Demberg, V. (2019b). Next sentence prediction helps implicit discourse relation classification within and across domains. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5790–5796, Hong Kong, China. Association for Computational Linguistics.

Sileo, D., Van De Cruys, T., Pradel, C., and Muller, P. (2019). Mining discourse markers for unsupervised sentence representation learning. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota. Association for Computational Linguistics.

Sporleder, C. and Lascarides, A. (2005). Exploiting linguistic cues to classify rhetorical relations. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-05)*, pages 532–539. Unknown Publisher. Pagination: 8.

Sporleder, C. and Lascarides, A. (2008a). Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369–416.

Sporleder, C. and Lascarides, A. (2008b). Using automatically labelled examples to classify rhetorical relations: an assessment. *Natural Language Engineering*, 14(3):369–416.

Strube, M. and Ponzetto, S. P. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1419–1424. AAAI Press.

Taghipour, K. and Ng, H. T. (2016). A neural approach to automated essay scoring. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Tang, L. and Liu, H. (2009). Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 817–826.

Tien Nguyen, D. and Joty, S. (2017). A neural local coherence model. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330, Vancouver, Canada. Association for Computational Linguistics.

Torabi Asr, F. and Demberg, V. (2012). Implicitness of discourse relations. In Kay, M. and Boitet, C., editors, *Proceedings of COLING 2012*, pages 2669–2684, Mumbai, India. The COLING 2012 Organizing Committee.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

van Dijk, T. A. and Kintsch, W. (1983). *Strategies of Discourse Comprehension*. Academic Press, New York.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, M., Chen, L., Cheng, F., Liao, S., Zhang, X., Wu, B., Yu, H., Xu, N., Zhang, L., Luo, R., Li, Y., Yang, M., Huang, F., and Li, Y. (2024). Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

Wang, X., Li, S., Li, J., and Li, W. (2012). Implicit discourse relation recognition by selecting typical training examples. In Kay, M. and Boitet, C., editors, *Proceedings of COLING 2012*, pages 2757–2772, Mumbai, India. The COLING 2012 Organizing Committee.

Webber, B. (2009). Genre distinctions for discourse in the Penn TreeBank. In Su, K.-Y., Su, J., Wiebe, J., and Li, H., editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 674–682, Suntec, Singapore. Association for Computational Linguistics.

Webber, B., Prasad, R., and Lee, A. (2019a). Ambiguity in explicit discourse connectives. In Dobnik, S., Chatzikyriakidis, S., and Demberg, V., editors, *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 134–141, Gothenburg, Sweden. Association for Computational Linguistics.

Webber, B., Prasad, R., Lee, A., and Joshi, A. (2019b). The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 35:108.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Wu, C., Cao, L., Ge, Y., Liu, Y., Zhang, M., and Su, J. (2022). A label dependence-aware sequence generation model for multi-level implicit discourse relation recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11486–11494.

Wu, H., Shen, X., Lan, M., Mao, S., Bai, X., and Wu, Y. (2023). A multi-task dataset for assessing discourse coherence in Chinese essays: Structure, theme, and logic analysis. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6673–6688, Singapore. Association for Computational Linguistics.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021a). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021b). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Xiang, W. and Wang, B. (2023). A survey of implicit discourse relation recognition. *ACM Comput. Surv.*, 55(12).

Xiang, W., Wang, Z., Dai, L., and Wang, B. (2022). ConnPrompt: Connective-cloze prompt learning for implicit discourse relation recognition. In Calzolari, N., Huang, C.-R., Kim, H., Pustejovsky, J., Wanner, L., Choi, K.-S., Ryu, P.-M., Chen, H.-H., Donatelli, L., Ji, H., Kurohashi, S., Paggio, P., Xue, N., Kim, S., Hahm, Y., He, Z., Lee, T. K., Santus, E., Bond, F., and Na, S.-H., editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 902–911, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiao, W., Huber, P., and Carenini, G. (2021). Predicting discourse trees from transformer-based neural summarizers. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4139–4152, Online. Association for Computational Linguistics.

Xu, P., Saghir, H., Kang, J. S., Long, T., Bose, A. J., Cao, Y., and Cheung, J. C. K. (2019). A cross-domain transferable neural coherence model. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, Florence, Italy. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yao, L., Mao, C., and Luo, Y. (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Ye, R., Zhang, C., Wang, R., Xu, S., and Zhang, Y. (2024). Language is all a graph needs. In Graham, Y. and Purver, M., editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1955–1973, St. Julian's, Malta. Association for Computational Linguistics.

Yin, W., Radev, D., and Xiong, C. (2021). DocNLI: A large-scale dataset for document-level natural language inference. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Yu, C., Zhang, H., Song, Y., and Ng, W. (2022a). CoCoLM: Complex commonsense enhanced language model with discourse relations. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1175–1187, Dublin, Ireland. Association for Computational Linguistics.

Yu, N., Zhang, M., Fu, G., and Zhang, M. (2022b). RST discourse parsing with second-stage EDU-level pre-training. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4269–4280, Dublin, Ireland. Association for Computational Linguistics.

Zeldes, A. (2017a). The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zeldes, A. (2017b). The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zhang, B., Su, J., Xiong, D., Lu, Y., Duan, H., and Yao, J. (2015). Shallow convolutional neural network for implicit discourse relation recognition. In Màrquez, L., Callison-Burch, C., and Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Lisbon, Portugal. Association for Computational Linguistics.

Zhang, L., Xing, Y., Kong, F., Li, P., and Zhou, G. (2020). A top-down neural architecture towards text-level parsing of discourse rhetorical structure. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6386–6395, Online. Association for Computational Linguistics.

Zhou, H., Lan, M., Wu, Y., Chen, Y., and Ma, M. (2022). Prompt-based connective prediction method for fine-grained implicit discourse relation recognition. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3848–3858, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhou, W., Huang, K., Ma, T., and Huang, J. (2021). Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14612–14620.

Zhou, Z. M., Lan, M., Niu, Z. Y., Xu, Y., and Su, J. (2010). The effects of discourse connectives prediction on implicit discourse relation recognition. In Katagiri, Y. and Nakano, M., editors, *Proceedings of the SIGDIAL 2010 Conference*, pages 139–146, Tokyo, Japan. Association for Computational Linguistics.