Inauguraldissertation zur Erlangung der Doktorwürde der Neuphilologischen Fakultät der Ruprecht-Karls-Universität Heidelberg

Graph-based Patterns for Local Coherence Modeling

vorgelegt von

Mohsen Mesgar

Referent: Prof. Dr. Michael Strube

Korreferent: Prof. Dr. Katja Markert

Einreichung: Oktober 2018

Abstract

Coherence is an essential property of well-written texts. It distinguishes a multi-sentence text from a sequence of randomly strung sentences. The task of local coherence modeling is about the way that sentences in a text link up one another. Solving this task is beneficial for assessing the quality of texts. Moreover, a coherence model can be integrated into text generation systems such as text summarizers to produce coherent texts.

In this dissertation, we present a graph-based approach to local coherence modeling that accounts for the connectivity structure among sentences in a text. Graphs give our model the capability to take into account relations between non-adjacent sentences as well as those between adjacent sentences. Besides, the connectivity style among nodes in graphs reflects the relationships among sentences in a text.

We first employ the entity graph approach, proposed by Guinaudeau and Strube (2013), to represent a text via a graph. In the entity graph representation of a text, nodes encode sentences and edges depict the existence of a pair of coreferent mentions in sentences. We then devise graph-based features to capture the connectivity structure of nodes in a graph, and accordingly the connectivity structure of sentences in the corresponding text. We extract all subgraphs of entity graphs as features which encode the connectivity structure of graphs. Frequencies of subgraphs correlate with the perceived coherence of their corresponding texts. Therefore, we refer to these subgraphs as coherence patterns.

In order to complete our approach to coherence modeling, we propose a new graph representation of texts, rather than the entity graph. Our approach employs lexico-semantic relations among words in sentences, instead of only entity coreference relations, to model relationships between sentences via a graph. This new lexical graph representation of texts plus our method for mining coherence patterns make our coherence model.

We evaluate our approach on the readability assessment task because a primary factor of readability is coherence. Coherent texts are easy to read and consequently demand less effort from their readers. Our extensive experiments on two separate readability assessment datasets show that frequencies of coherence patterns in texts correlate with the readability ratings assigned by human judges. By training a machine learning method on our coherence patterns, our model outperforms its counterparts on ranking texts with respect to their readability. As one of the ultimate goals of coherence models is to use them in text generation systems, we show how our coherence patterns can be integrated into a graph-based text summarizer to produce informative and coherent summaries. Our coherence patterns improve the performance of the summarization system based on both standard summarization metrics and human evaluations. An implementation of the approaches discussed in this dissertation is publicly available[1].

[1] `https://github.com/MMesgar/`

Zusammenfassung

Kohärenz ist eine wesentliche Eigenschaft von gut geschriebenen Texten. Sie unterscheidet einen Text mit mehreren Sätzen von einer Folge von zufällig aufgereihten Sätzen. Bei der Aufgabe der lokalen Kohärenzmodellierung geht es darum, wie Sätze in einem Text miteinander verbunden sind. Die Lösung dieser Aufgabe ist nützlich für die Bewertung von Textqualität. Außerdem kann ein Kohärenzmodell in Textgenerierungssystemen wie z.B. Textzusammenfassungssystemen integriert werden, um zusammenhängende Texte zu erzeugen.

In dieser Doktorarbeit präsentieren wir einen graphbasierten Ansatz zur lokalen Kohärenzmodellierung, welcher die Verbindungsstruktur unter den Sätzen in einem Text darstellt. Die Graphen geben unserem Modell die Fähigkeit, sowohl die Verbindungen zwischen benachbarten als auch zwischen nicht benachbarten Sätzen zu berücksichtigen. Darüber hinaus spiegelt der Verbindungsstil unter Knoten in Graphen die Beziehungen zwischen den Sätzen in einem Text wider.

Zuerst verwenden wir den von Guinaudeau und Strube (2013) entwickelten Entity-Graph-Ansatz, um einen Text durch einen Graphen darzustellen. In diesem Ansatz werden Sätze durch Knoten repräsentiert, und Kanten zwischen Knoten repräsentieren koreferente Ausdrücke in zwei Sätzen. Danach entwickeln wir graph-basierte Eigenschaften zum Erfassen der Verbindungsstruktur von Knoten in einem Graphen, und von den Sätzen im dazugehörigen Text. Wir extrahieren alle Untergraphen der Entity-Graphen als Merkmale, die die Verbindungsstruktur der Graphen repräsentieren. Die Häufigkeit der Untergraphen korrelieren mit der wahrgenommenen Kohärenz ihrer entsprechenden Texte. Deshalb beziehen wir uns auf diese Untergraphen als Kohärenzmuster.

Um unseren Ansatz zur Kohärenzmodellierung zu vervollständigen, schlagen wir als Alternative zum Entity-Graphen eine neue Graphrepräsentation von Texten vor. Unser Ansatz nutzt lexiko-semantische Beziehungen zwischen Wörtern, und nicht nur Koreferenzbeziehungen, um semantische Beziehungen zwischen Sätzen als Graph zu Modellieren. Diese neue lexikalische Graphrepräsentation von Texten plus unsere Methode für die Kohärenzmusterextraktion bildet unser Kohärenzmodell.

Wir evaluieren unseren Ansatz überwiegend im Hinblick auf die Lesbarkeitsbewertung von

Texten, weil Textkohärenz ein Schlüsselfaktor für diese Aufgabe ist. Kohärente Texte sind einfach zu lesen und zu verstehen und erfordern folglich weniger Aufwand von ihren Lesern. Durch umfangreiche Versuche auf zwei verschiedenen Datensätzen zur Lesbarkeitsbewertung untersuchen wir die Korrelation zwischen den Häufigkeiten der Kohärenzmuster in Texten und von menschlichen Subjekten vorgenommenen Lesbarkeitsbewertungen.

Durch das Trainieren eines maschinellen Lernverfahrens auf unseren Kohärenzmustern übertrifft unser Modell seine Gegenstücke im Bewerten von Texten hinsichtlich ihrer Lesbarkeit. Da eines der eigentlichen Ziele der Kohärenzmodellierung der Einsatz in Texterzeugungssystemen ist, zeigen wir, wie unsere Kohärenzmuster in ein graphbasiertes Textzusammenfassungssystem zum Erzeugen von informativen und kohärenten Zusammenfassungen integriert werden kann. Unsere Kohärenzmuster verbessern die Leistung des Zusammenfassungssystems basierendn auf sowohl Standardzusammenfassungsmetriken als auch auf menschliche Bewertungen. Eine Implementierung der in dieser Doktorarbeit diskutierten Ansätze ist öffentlich verfügbar[2].

[2] https://github.com/MMesgar/

Acknowledgments

First and foremost, I would like to thank my supervisor, Michael Strube. It has been an honor for me to be a Ph.D. student of Michael. He is a true mentor, who has guided me in how to perform and present fascinating research. I appreciate all of his contributions of time, ideas and motivations to make my Ph.D. experience productive and stimulating. Moreover, his patient and careful editing of my research papers have considerably improved my skills in scientific writing and presentation.

I give my gratitude to Anette Frank, Katja Markert, and their team members for their useful comments and questions about my research during my presentations in the Ph.D. colloquium series at Heidelberg University. I am eternally grateful to Katja Markert for agreeing to be the second reviewer of this dissertation.

Many thanks to the Klaus Tschira Foundation for financially supporting me by a Ph.D. scholarship during the time that I was conducting my research at the Heidelberg Institute for Theoretical Studies (HITS). I appreciate the German Research Foundation as part of the research training group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) for supporting me, as one of its associate researchers, to travel to a conference in the U.S. for presenting one of my papers, which is used in the research of this dissertation.

I want to express my heartfelt gratefulness to all HITS members for providing a friendly and inspiring environment. In particular, I thank my colleagues in the natural language processing (NLP) group at HITS: Alex Judea, Angela Fahrni, Benjamin Heinzerling, Daraksha Parveen, Nafise Moosavi, Mark-Christoph Müller, and Sebastian Martschat. People who were the first that listened to my research ideas and the last that read my papers before submissions.

My sincere thanks also go to Holger Gebert, one of the leaders of the NLP projects at SAP SE in Germany, and Ashish Gard, the manager of the Siri team at Apple Inc. in the U.S., for providing me an opportunity to join their teams as an intern. They gave me access to the laboratory and research facilities to deepen my knowledge of NLP and machine learning.

Last but not least, I want to thank my family and friends, who have always been supportive. Individually, my heartfelt thanks are for my spouse Shaghayegh Tavafi for all of her lovely supports in all easy and challenging moments of our residency in Germany.

Contents

1 Introduction 1

1.1 Motivation and Research Questions 2

1.2 Contributions 3

1.3 Outline 4

1.4 Published Work 5

2 Coherence Modeling 7

2.1 Problem Definition 7

2.2 The Linguistics of Coherence 9

2.3 Evaluation 18

3 Related Work 21

3.1 Entity-based Approaches to Local Coherence 21

3.2 Lexical Approaches to Local Coherence 35

3.3 Applications and Evaluations of Coherence Models 41

4 Graph-based Coherence Patterns 47

4.1 Why Graph-based Patterns? 47

4.2 Coherence Pattern Extraction 49

4.3 Coherence Modeling 55

4.4 Experiments 56

4.5 Summary 86

5 Lexical Cohesion Graph 89

5.1 Lexical Cohesion 89

5.2 LexGraph 91

5.3 Coherence Pattern Mining 94

5.4 Smoothing 95

5.5 Experiments 99

5.6 Summary 109

6 Conclusions 113

6.1 Contributions 113

6.2 Future Work 114

List of Figures 117

List of Tables 119

List of Algorithms 121

Bibliography 123

1 Introduction

Research in natural language processing intends to provide models for understanding and generating texts. One crucial aspect in the processing of multi-sentence texts is *coherence*: How sentences in such a text are related to one another to make the text a whole. As an example, consider the following text snippet[1]:

(1) A total of 248 people, including a dozen Americans, were killed in the terrorist bombing of the U.S. Embassy in Nairobi on August 7, 1998. A twin attack on the U.S. Embassy in Tanzania killed 11 people, all Africans. Osama bin Laden is the suspected mastermind of the bombings. Through the Saudis, the United States asked the Taliban, the Islamic movement that controls most of Afghanistan, to deport bin Laden, but they refused. Evidence suggests that the terror suspects accused in the bombings, regardless of their nationality or place of residence, are associates of bin Laden or associated with terrorist groups under his control.

The above text is a summary which is provided by a human from several documents. All sentences are attached together in a way that the whole text conveys a meaning. The first sentence gives some information about a "bombing". The second sentence takes this information and expands it to another instance. The third sentence uses the given information ("bombing") from its preceding sentences to introduce "Osama bin Laden" as new information. The rest of the sentences follow a similar structure of relationships. The text below[2] is a summary which is generated automatically from the same cluster of documents. It is less coherent than the text in Example (1), as its sentences are weakly related to each other.

(2) Solemn-faced Kenyans, whose relatives were killed in the terrorist bombing of a U.S. Embassy, collected benefits on Friday. They said failed to compensate for their losses. Nearly two months after the bombings of the American

[1] Article *D31038.M.100.T.B* Taken from `http://homepages.inf.ed.ac.uk/mlap/coherence/` accessed 28 May 2018.

[2] Article *D31038.M.100.T.16* taken from `http://homepages.inf.ed.ac.uk/mlap/coherence/` accessed 28 May 2018.

> Embassies in Kenya and Tanzania, a picture of those charged in the case is slowly emerging. Nine months before the attack on the American Embassy here, U.S. intelligence officials received a detailed warning that Islamic radicals were plotting to blow up the building, according to Kenyan and American officials.

A coherence model should first represent how sentences in a text are related to each other. It then uses the structure of relations to ideally rank and distinguish texts with respect to their perceived coherence. For example, consider the text snippets in Example (1) and Example (2), a coherence model should ideally rank the former text higher, in terms of coherence, than the latter one.

As it has been shown in the above examples, applications of a coherence model are in downstream tasks in natural language processing. One example is in readability assessment, in which coherence is employed as an essential factor in measuring the quality of texts. Coherent texts avoid confusion, so they are easy to read and follow. Another example is in text summarization, which can employ a coherence model in two ways: First, a coherence model can be used for evaluating the quality of outputs of automatic summarizers. In this case, the usage of coherence models for the summarization task is similar to their usage in general text quality assessment. Second, a coherence model can as one component be integrated into summarization systems to generate coherent summaries directly.

In the research presented in this dissertation, we aim to develop a computational model for text coherence. We also intend to evaluate our coherence model in extrinsic applications. In the remainder of this chapter, we take a further look into the motivation of the research conducted in this thesis and formulate main research questions (Section 1.1), briefly explain our contributions (Section 1.2), present the outline of this dissertation (Section 1.3), and describe which parts of this dissertation were published (Section 1.4).

1.1 Motivation and Research Questions

As we have described above, the goal of computational coherence models is to compare texts with respect to their coherence. It suggests that there should be certain features which are characteristic of coherent texts while these features are absent in incoherent texts. In the literature, we encounter different sets of features relying on relations which are extracted from a text. One type of relations prominently employed by coherence models is coreference. In such coherence models, relations among sentences represent the existence of noun phrases that refer to the same entity in sentences. The underlying premise of these models is that coherent texts reveal specific patterns in their relations. However, these models predefine and limit patterns to linear relations over adjacent sentences. This observation leads us to the first research question investigated in this dissertation: **Do there exist nonlinear connectivity patterns in coherent texts that take long-distance relations into account?** If we can answer this question by discovering frequent patterns, which involve long-distance relations, in coherent texts, a follow-up question arises, which is how the frequencies of these patterns correlate with the quality of texts. Another question is whether these features improve the performance of downstream natural language processing systems. The answers to these questions help to learn how coherence patterns proposed in the research in this dissertation compete with their peers, where they are evaluated in coherence related downstream tasks.

In order to develop a robust computational coherence model, we not only need an approach to extract coherence patterns, which represent connectivity structures of sentences, but we also require a computational method to encode semantic relations between sentences. Sentence relations are not limited to coreference between referring expressions in sentences; other semantic relations such as synonymy and antonymy among words in sentences can connect sentences as well. This fact motivates our second research question: **How can we model sentence relations in a text beyond coreference relations over entities by means of semantic relations among words of sentences?** In order to answer this question, we first need to define appropriate word representations. Word representations should give the model the capability to quantify lexico-semantic relations between words. The model is then required to encode connections between sentences based on the words that are semantically related. Finally, given such representations of relations among sentences in a text, we can use our approach to coherence pattern extraction for modeling the connectivity structure of sentences in texts. By such patterns, we then rank texts concerning their coherence.

1.2 Contributions

We answer the first question by introducing a graph-based representation of coherence patterns. The graph representation empowers our coherence model to take long-distance relations as well as relations among adjacent sentences into account. It further captures the connectivity style of relations among sentences. In such graphs, nodes encode sentences, and edges capture relationships between sentences. Then, we formulate the task of extracting coherence patterns from a set of texts as a subgraph mining problem from a set of graphs. We show how frequencies of subgraphs in a graph capture the connectivity style of nodes in the graph and consequently the coherence of the corresponding text. We illustrate how frequencies of patterns in texts correlate with the quality ratings that are assigned to those texts by human judges.

We answer the second question by motivating and developing an approach to coherence modeling based on lexical relations. This approach represents relations among sentences via a graph. Nodes in such graphs represent sentences, and edges represent the existence of lexico-semantic relations among words in sentences. We explain how word embeddings are employed to quantify the strength of semantic relations between words in sentences. We show that applying subgraph mining methods on such graph representations of texts leads to predictive coherence patterns. We further investigate the impact of the size of subgraphs on the performance of patterns. We discuss that the frequencies of large subgraphs highly correlate with the quality of texts as they are more informative about connectivity structures of graphs in comparison to small subgraphs. However, most of the large subgraphs only occur in a few graph representations of texts resulting in a sparsity problem. We show how smoothing methods, which are applied in statistical language models, can be adapted to solve this sparsity problem in the frequency of coherence patterns.

The implementation of graph representations, subgraph mining approaches, and the smoothing method discussed in this thesis are publicly available[3].

1.3 Outline

The remainder of this thesis is organized into five chapters.

In Chapter 2, we discuss the task of coherence modeling in detail. We give a formal definition of coherence modeling. Furthermore, we address linguistic properties, primary issues, and evaluation approaches.

In Chapter 3, we review the related work on which we mainly built our coherence model. We survey different tasks that have been employed to evaluate the coherence models presented in the research in this dissertation.

In Chapter 4, we present our approach to coherence pattern mining. We recast the problem of coherence pattern mining as extracting frequent subgraphs in graph representations of texts. We assess the usefulness of coherence patterns on the readability assessment task. We show how coherence patterns extracted from a set of news articles correlate with their readability ratings assigned by human judges. We observe that the frequencies of patterns as features,

[3] Available for download at `https://github.com/MMesgar/`

which encode coherence, are more predictive than other examined features for ranking texts concerning their coherence. A fundamental analysis of the size of extracted subgraphs leads us to the observation that by increasing the number of nodes in subgraphs the predictive power of coherence patterns for ranking texts improves. We furthermore evaluate our approach to coherence pattern mining in the summarization task. We show how subgraphs extracted from coherent summaries can improve the performance of an automatic summarization system to produce more coherent summaries.

In Chapter 5, we propose a graph-based representation method for modeling coherence based on lexico-semantic relations between words in sentences. We show that coherence patterns extracted from such graphs are more beneficial for the readability assessment task in comparison to the patterns extracted from the entity graph representations of texts. We investigate broader about the quality of coherence patterns extracted from such lexical graph representations of texts and the influence of their size on the overall performance of the model. We explain the sparsity problem in the frequencies of subgraphs and its impacts on the performance of our coherence model. Following smoothing methods utilized in statistical language models, we introduce an approach to solve the sparsity problem in graphs.

In Chapter 6, we summarize the answers that the research presented in this dissertation gives to the research questions formed in Section 1.1. Furthermore, we discuss possible research avenues for future work.

1.4 Published Work

Most research presented in this thesis is an extension of published research first-authored by the author of this thesis. Some parts of the presented research originated from the published research to which the author of this dissertation contributed.

The idea of using subgraphs as coherence patterns, presented in Chapter 4, was published in Mesgar and Strube (2015). A preliminary investigation of graph-based coherence modeling was presented in Mesgar and Strube (2014). The application of coherence patterns in summarization, also presented in Chapter 4, is published in Parveen et al. (2016). Our lexical approach to local coherence modeling, and the smoothing method, which both are described in Chapter 5, are proposed in Mesgar and Strube (2016). A follow-up paper to the research presented in this dissertation is published in Mesgar and Strube (2018).

2 Coherence Modeling

Local coherence modeling with varying specifications over the years is a crucial task for natural language processing. This chapter provides definitions related to this task as it is tackled in the research presented in this dissertation. We formally define the problem of coherence modeling (Section 2.1) and then explain the linguistics of coherence (Section 2.2). Finally, we discuss our evaluation approach for assessing the coherence models (Section 2.3) presented in this dissertation.

2.1 Problem Definition

In this research, we tackle the problem of local coherence modeling. The simplified definition of this task is to model how text units (or segments) in a text are related to one another. This task has been the focus of the majority work in text processing (see Chapter 3). Variations of this task consider different types of relations, such as rhetorical (Hovy and McCoy, 1989) or lexical (Morris and Hirst, 1991), between different spans of texts, such as clauses (Strube, 1998) or sentences (Halliday and Hasan, 1976), in various text types, such as dialogue (Wang et al., 2013) or monologue (Barzilay and Lapata, 2008). Here we formally define this task as it is investigated in this research with the goal to use this definition for establishing the next chapters of this dissertation.

2.1.1 Formal Modeling

In order to provide a formal definition of the task, i.e. local coherence modeling in texts, we first need to define what we refer to as a text. In this research, we assume that a text consists of two sentences or more.

Definition 1. *Text T is a sequence of a finite number of sentences $[s_0, s_1, s_2, ..., s_n]$, where the number of sentences is greater than 1.*

Each sentence in the above definition of a text is a list of words.

Definition 2. *Sentence s_i is a sequence of words $[w_0, w_1, w_2, ..., w_n]$ that forms a sentence structure in a text.*

An underlying assumption in research on text processing is that a text is more than the sum of its sentences (Webber et al., 2012). It is not sufficient to collect an arbitrary sequence of sentences in order to obtain a text. Sentences in a text are supposed to be related to one another to make a whole. Therefore, a relationship function is required to check if two sentences are related.

Definition 3. *Relationship function $R(s_i, s_j)$ indicates whether two sentences s_i and s_j in a text are related. The domain of this function is a pair of sentences, and its range is a number.*

The output of the relationship function indicates the strength of the relation between the input sentences. The output of this function can, however, be limited to a binary value $\{0, 1\}$. In this case, the value indicates if there is a relationship between sentences or not.

Given the above definition of the relationship function, relationships across all sentences in a text can be represented by a set P, which contains all relationships between any pair of sentences in the text.

Definition 4. *Let $r_{ij} = R(s_i, s_j)$ indicate the relationship between a sentence pair (s_i, s_j) in a text T; the set $P = \{r_{ij} | (s_i, s_j) \in T^2, i \neq j\}$ contains all r_{ij} for any pair of sentences in T.*

Although we define P as a set, it can be partially structured, e.g., where sentence s_i has to precede sentence s_j in a text. However, we do not make any assumption in this regard in our formulation to give coherence models the freedom to make it concrete.

While some texts can easily be recognized as coherent or incoherent, often local coherence is a matter of degree (Halliday and Hasan, 1976). A text can be less coherent when compared to one text, but more coherent when compared to another. As such, since the notion of coherence is relative, coherence assessment is better to be performed as a ranking problem. Given a pair of texts, a coherence model ideally ranks the texts with respect to their coherence. In order to rank texts concerning their coherence, we should capture patterns that frequently occur in more coherent texts and rarely in less coherent ones. Coherence patterns are templates of relationships among sentences in texts where their frequencies assist in distinguishing coherent texts from incoherent ones.

Definition 5. *A coherence pattern is a subset of relations $p \subseteq P$ occurring among sentences in a text.*

We define a function to model how coherence patterns are extracted from a corpus of texts.

Definition 6. *Given a corpus C, a pattern mining method M extracts all subsets of relations that occur in any text in C as a set of coherence patterns.*

The output of the pattern mining process from a corpus of texts is a set of coherence patterns. Frequencies of these patterns in a text encode the coherence of the text.

Definition 7. *Let $P = \{p_0, p_1, p_2, ..., p_m\}$ be a set of patterns extracted from a corpus of texts C; the perceived coherence of text T is represented by vector $\phi =< f_0, f_1, f_2, ..., f_m >$, where f_k is the frequency of pattern p_k in text T.*

A vector representation of coherence allows us to employ machine learning models to rank texts with respect to their coherence. In Chapter 3, we describe several approaches to modeling relationships between a pair of sentences, i.e. $R(s_i, s_j)$. We then explain how these approaches represent the set of all relationships in a text, i.e., P. We additionally review how different computational models derive method M in our formulation. We employ a plausible representation for texts, i.e. the entity graph representation (Guinaudeau and Strube, 2013), from the literature and develop our approach to extracting coherence patterns in Chapter 4. We further improve the predictive power of coherence patterns by a new approach to representing relationships among sentences, i.e. P, in Chapter 5.

2.2 The Linguistics of Coherence

The aforementioned formal definitions are sufficient for the research presented in this dissertation to develop a representation of cohesive relations, extract coherence patterns, and model coherence computationally. However, since coherence is a semantic property of text, its definition requires to be related to the text linguistic properties. Therefore, in this section, we explain the linguistics of coherence.

We start with the linguistic properties of what we refer to as a text in this research. As we aim to represent cohesive relations among sentences, we explain what aspects of sentences serve to relate them in a text. We finally discuss how coherence patterns are approached in the linguistics of coherence, since they are the core of our coherence model.

2.2.1 Text

The first definition in our formal model of coherence is about text. In linguistics, the word “text” is used to refer to any passage, spoken or written, of whatever length that forms a unified whole. In this dissertation, we follow other coherence models (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013) and use the word "text" to denote a monologue written passage, which includes more than one sentence.

One-sentence texts, of course, do exist, such as public notices, proverbs, and advertising slogans. For instance, a sample text with only one sentence is shown in Example (3)[1]:

(3) A journey of a thousand miles begins with a single step.

However, in the research presented in this thesis[2], we assume that texts contain at least two sentences. We also assume that texts are written in formal register, in contrast to informal register like language used in tweets[3].

2.2.2 Coherence

Coherence is a vital factor for distinguishing a well-written text from a sequence of unrelated sentences. A coherent text discusses a sequence of topics in a structured way by which a reader can recognize the relationships among topics, and collectively render the text as a unified whole (Stede, 2012). Lautamatti (1978) defines the term "topic", generally, as what text units are mainly about. Each topic tends to occupy a (topical) segment in a text. Coherence is the result of the relations and the structures of topical segments in a text (Hearst, 1997). This structure is sometimes referred to as global coherence since it is coarse-grained and may span the entire text (Elsner et al., 2007). In general, however, it is not straightforward, first, to define the notion of topics and, second, to recognize topics and their boundaries across text segments (Stede, 2012).

2.2.3 Local Coherence

From a linguistic viewpoint, a (coherent) text employs linguistic devices, which are readily identifiable linguistic signals, to relate sentences[4] of a text to each other. These devices signal readers to interpret each sentence while considering its relationships with other linked sentences (Dijk, 1977). Therefore, understanding a text implies uncovering such relationships among its sentences. Local coherence is about the way that linguistic devices are utilized to relate sentences in a text (Stede, 2012). Halliday and Hasan (1976) refer to this phenomenon

[1] Taken from `https://www.engvid.com/english-resource/50-common-proverbs-sayings/`, accessed 1 June 2018.

[2] Texts that consist of one sentence do not exist in the datasets employed for experiments of the research presented in this thesis.

[3] A post made on the social media application Twitter.

[4] We limit the text units in linguistics to sentences.

as “cohesion”. Stoddard (1991) argues that local coherence and cohesion are not distinguishable and can be used interchangeably. With reference to Stoddard (1991), we use the term “local coherence”.

Stede (2012) states that signals of local coherence serve as indicators of topicsl structure in a text. Exisiting of these signals is a sign of topic continuity, and the absence of a surface relation is a sign of a topic shift. Barzilay and Lapata (2008) note that the local coherence of a text implies its global coherence. Since the research presented in this dissertation is about “local coherence modeling”, henceforth we refer to it shortly as “coherence modeling”. We explicitly distinguish these terms where they are not distinguishable from the context.

Cohesive devices, which are used to connect sentences, can be grouped into grammatical and lexical relations between elements of sentences (Halliday and Hasan, 1976). Grammatical relations are reference, substitution, ellipsis and conjunctions. Lexical relations include any lexico-semantic relation such as repetition, synonym, antonym, and the like between words of sentences. Among the grammatical relations, reference devices, which are also known as entity relations, have widely been studied. The intuition behind the leveraging of entity relations for coherence modeling is that related sentences in a text contain text pieces that are used to refer to the same entities. Since the core of these models is the entity, we follow Barzilay and Lapata (2005) and Elsner and Charniak (2010), and define an entity as follows:

Definition 8. *An entity is perceived as a person, a physical object, a concept, or an abstraction that exists (or may exist) in the world external to a text.*

An entity can be referred to in different ways by various expressions, or mentions.

Definition 9. *The pieces of a text that are used to refer to an entity are called mentions.*

Given these definitions, one way of representing an entity is to group all mentions that refer to that entity. Each cluster of mentions represents an entity. The task of identifying mentions that refer to the same entity is known as coreference resolution.

Definition 10. *The task of detecting all mentions in a text and clustering all mentions that refer to the same entities is called coreference resolution.*

The text in Example (4)[5] shows how coreference relations among mentions of an entity relate its sentences.

[5] Taken from `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1162/handouts/cs224n-lecture11-coreference-6up.pdf`, accessed 2 June 2018.

(4) **Mr. Obama** visited the city.
The president talked about Milwaukee's economy.
He mentioned new jobs.

In the text shown in Example (4), "Mr. Obama", "The president" and "He" are three mentions that refer to the 44th president of the United States. The three sentences of this text are connected because they contain mentions that refer to the same entity.

One of the popular entity-based frameworks for local coherence modeling is centering theory (Grosz et al., 1995). The heart of this theory is the concept of the "text center". The text center at any given point in a text is the most "salient" entity at that point. For instance, at the end of the last sentence in the text that is shown in Example (4), the text center is on the entity "Barack Obama". Centering theory (for English texts) takes the grammatical roles of mentions of entities in sentences as the most significant linguistic signs for the saliency of entities. Precisely, the grammatical subject is preferred as the default position for the text center. So the text center can be various entities at different points in a text.

Centering theory accounts for the process of the center flow in a text as the text center captures the focus of attention in the reader's mind through the text (Grosz et al., 1995). Depending on the configuration of the grammatical roles of mentions of an entity in adjacent sentences, Brennan et al. (1987) define four different types of transitions for the text center across sentences. These transitions capture the smoothness of the center move from one sentence to another. Therefore they encode the local coherence of a text. Where a sentence focuses on the topic, i.e. the text center, that is discussed in sentences preceding that sentence, the transition is Retain or Continue. Other transitions involve a topic shift: Smooth Shift, or Rough Shift. The center transitions presented in centering theory have directly motivated several coherence models, e.g., the model proposed by Karamanis et al. (2004). However, since centering theory requires human annotations for center transitions across sentences, some computational models preferred to employ principles of centering theory as soft constraints or features in a probabilistic framework. An example of such models is the entity grid model (Barzilay and Lapata, 2005, 2008), which is discussed in more detail in Chapter 3.

Further linguistic research shows that grammatical role information is far less predictive for tracking the text center in other languages such as German, which is a free word order language (Strube and Hahn, 1996). Instead of that, Strube and Hahn (1996) consider the "functional information structure" (Daneš, 1974). The idea behind this structure is to capture the flow of information within a text. Some information in a sentence is known (or old) for readers because sentences preceding that sentence discuss it, and some information is new. The way that the information status changes within a text reveals how smoothly topics flow across sentences, or how coherent the text is (Daneš, 1974).

The other perspective of local coherence is lexical cohesion. It is the cohesive effect based on lexico-semantic relations between words in a text (Halliday and Hasan, 1976). An advantage of lexical cohesion models, in contrast to entity-based models, is that they require no semantic annotation. The insight of local coherence resulting from lexical relations is that content words in a text do not occur independently of one another, but rather bear semantic relatedness. So in a coherent text, content words of sentences are expected to be semantically related. A form of lexical cohesion is reiteration, which involves different types of lexical relations such as repeating a word, using a synonym of a word, and employing a superordinate word. Example (5) is taken from Halliday and Hasan (1976) to illustrate these relations:

(5) (a) Repetition:
There was a large **mushroom** growing near her, about the same height as herself; and when she had looked under it, it occurred to her that she might as well look and see what was on the top of it.
She stretched herself up on tiptoe and peeped over the edge of the **mushroom**, [...]

(b) Synonymy:
Accordingly [...] I took leave, turned to the **ascent** of the peak.
The **climb** is perfectly easy.

(c) Superordinate:
Henry's bought himself a new **Jaguar**.
He practically lives in the **car**.

In Example (5)(a), the word "mushroom" is exactly repeated in the sentences. In (b) the two words "ascent" and "climb" carry the same meaning. In (c), the word "car" is a superordinate, any word whose meaning includes that of the earlier one, of "Jaguar" since a vehicle is a superordinate of a car. The relationship between "spoon" and "teaspoon" is another example of the superordinate relation. The boundary between the reiteration type in lexical cohesion and reference type in entity relations is by no means clearcut (Halliday and Hasan, 1976). It clarifies why local coherence and cohesion are, for purposes of the research presented in this dissertation, largely synonymous.

In summary, there is a local coherence relation between any pair of lexical items that stand to each other in some lexico-semantic relations, even including relations between word pairs shown in Example (6) (Halliday and Hasan, 1976).

(6) rail ... road
car ... brake
try ... succeed
walk ... drive
Tuesday ... Thursday
like ... hate
red ... green

For local coherence modeling, it is (almost) sufficient to know that a pair of words is in a relationship, apart from the type of the relation (Halliday and Hasan, 1976). Hoey (1991) also examined how lexical elements make a text organized and contribute to local coherence. He showed that relationships between semantically related words in a text follow similar patterns in texts. In Chapter 3, we survey some related computational models of lexical cohesion in texts.

2.2.4 Coherence Patterns

Numerous researchers and practitioners in natural language processing deal with whole texts rather than individual sentences. While it is evident that text must have a coherent structure, its characteristics are less explicit, making it more difficult to exploit in applications (Webber et al., 2012). A text commonly comprises a sequence of sentences. Text structures are the patterns that one observes in connections among multi-sentence texts (Webber et al., 2012). Recognizing these patterns, which are referred to as coherence patterns, in terms of the elements that construct them is essential to derive and interpret information in a text correctly. Coherence patterns can also be characteristics of particular types of texts and therefore be of value in assessing the quality of texts generated automatically.

The concept of coherence patterns is linguistically derived from the “texture” of texts (Halliday and Hasan, 1976). The texture of text is the semantics perceived because of the combination of the different text units. Stoddard (1991) defines a text as “a phenomenon of seemingly infinite complexity due to its synergistic nature”, where units of a text are supposed to cooperate for an enhanced effect. It is synergism that makes texts more than consecutive words and sentences. The dynamics of synergism, which is because of its multi-dimensionality, is beyond the linear and sequential structure of texts. One cause of the multi-dimensionality of synergism is a global component which is referred to as texture.

From Stoddard (1991)'s perspective, the texture of text is interpretable by means of elements that occur in texts and distinguish them. She refers to these elements as "coherence patterns". So, texture manifests itself in coherence patterns that occur in similar texts. We have discussed that texture is one cause of the multi-dimensionality of synergism in texts; therefore, texture involves the quality of depth, which may range from minimal (approximating "flatness") to maximal or at any level in between. In other words, coherence patterns can be as basic as the linear relations among consecutive text units, or be more complicated and non-linear by incorporating long-distance relations between non-adjacent units.

The texture, which is a composite of patterns, is like the "fingerprint" of a text (Stoddard, 1991), which can be used to distinguish texts. In this sense, the texture of text is mostly related to "style", which is undoubtedly associated with the local coherence of text (Barzilay and Lapata, 2008). This explains why such patterns are referred to as coherence patterns. Furthermore, this linguistically supports our first research question that concerns the non-linearity of patterns in distinguishing coherent texts from incoherent ones.

Coherence patterns occurring in a text should be recognizable by readers of the text to assist them to understand the text easily. The unity of text is the result of the interactiveness of text units. Such interactiveness seems to have a degree of consistency across texts (Stoddard, 1991), so they should be identifiable as patterns. Moreover, it would be easier for readers to smoothly process texts in which patterns of interactions of text units are not only clearly recognizable, but also familiar to readers.

Stoddard (1991) graphically illustrates[6] the way that text units interact in a few texts. The results of her text analysis show that local coherence manifests itself in the connectivity structure of text units in the examined graphical illustrations of texts. The key results of her study can be summarized as follows: First, the unity of a text is better to be modeled by means of patterns that span through adjacent and non-adjacent text units, and second, both typologies (i.e. graphical structure) and counting should be involved to gain a better understanding of the local coherence of text. So, one factor that must be considered in describing coherence patterns as the input to texture is the likelihood of patterns occurring over the broad stretch of a text. The facts that cohesive patterns occur in texts and cohesiveness is relative in texts provide useful validation of the intuition used in the research presented in this thesis: Coherent texts reveal some regularities in their structure that can be encoded with the frequency of coherence

[6] The term of "networks of cohesion" proposed by Halliday and Hasan (1976) can be interpreted (almost) equivalent to this graphical illustration.

patterns.

Daneš (1974) had spoken about the concepts of text structure and coherence patterns more generally and earlier than Stoddard (1991). Daneš (1974) describes the structure of texts by the concept of "thematization", which has also been noticed by Halliday and Hasan (1976) as "given-new information structure". At each point of a text, "given information" is what has been talked about earlier than that point in the text and "new information" is been mentioned now. A theme, from Daneš (1974)'s perspective, is a point of departure where a text flows from a topic towards a rheme or another topic. In simple words, theme can be realized as given information, and rheme is new information. The contextual determination of givenness is far from being a simple phenomenon. Daneš (1974) explains that given information can be realized either directly with an identically worded expression or indirectly with a synonymous one. The indirect mentioning is based on semantic inference. For instance, the expression "illness" occurring in a sentence might convey a piece of given information if in one of its preceding sentences "disease" has been somehow mentioned. In opposite, the new information may neither be mentioned in its proceeding context nor be related to any given information.

Thematization is about the structure of transitions between themes and rhemes in a text. Daneš (1974) illustrates the relation between a theme and a rheme in a text unit by $T \rightarrow R$. This notation encodes that the flow of information in a text unit is from given information (or theme, T) to new information (or rheme, R). Daneš (1974) states that the inquiry into the thematic organization of the text is highly connected with the investigation of the so-called "text coherence" or "text connectivity". He analyzes Czech scientific and other professional texts, as well as some other materials in the German and English languages. He ascertains several essential types of organizational patterns in the examined texts, represented in Table 2.1. Daneš (1974) interprets these patterns as follows:

- Pattern 1: This patterns illustrates a linear transition between themes and rhemes. In this pattern, each text unit takes the rheme presented in the preceding context of the unit as given information and transfers it to new information or a new rheme. In other words, each R (i.e. new information) in a text unit becomes T (i.e. given information) in its next unit.

- Pattern 2: This pattern depicts a constant theme continuation across text units. One theme appears in a series of text units, each of which, however, presents new information about the presented theme.

- Pattern 3: In this pattern, $[T]$ indicates a hypertheme, which is a global theme of a text

Pattern ID	**Pattern**
Pattern 1	$$\begin{tikzcd} T_1 \arrow[r] & R_1 \arrow[d] & & \\ & T_2(= R_1) \arrow[r] & R_2 \arrow[d] & \\ & & T_3(= R_2) \arrow[r] & R_3 \end{tikzcd}$$
Pattern 2	$$\begin{tikzcd} T_1 \arrow[r] \arrow[d] & R_1 \\ T_1 \arrow[r] \arrow[d] & R_2 \\ T_1 \arrow[r] & R_3 \end{tikzcd}$$
Pattern 3	$$\begin{tikzcd} & & {[T]} \arrow[dll] \arrow[ddl] \arrow[dddr] & & \\ T_1 \arrow[r] & R_1 & & & \\ & T_2 \arrow[r] & R_2 & & \\ & & & T_3 \arrow[r] & R_3 \end{tikzcd}$$
Pattern 4	$$\begin{tikzcd} & T_1 \arrow[r] & R_1 \; (= R_1' + R_2'') \arrow[ddl] \arrow[ddddl, bend left] \\ & \vdots & \\ & T_2' \arrow[r] & R_2' \\ & \vdots & \\ & T_2'' \arrow[r] & R_2'' \end{tikzcd}$$

Table 2.1: Coherence patterns that are defined by Daneš (1974). A horizontal arrow indicates a transition in an utterance, while a vertical one indicates a contextual connection within utterances.

unit. This pattern shows that different units can be connected because their themes, or their given information, are semantically related to a hypertheme.

- Pattern 4: Different combinations of patterns can emerge in different texts. Some of these combinations are such frequent that they can be taken as particular types of theme-rheme transitions of a higher order. This pattern is one of the high-order patterns, where a text unit presents two (which can potentially be several) rhemes, R' and R'', in connection with a given theme. First, R' is expanded, and when its progression completes, R'' becomes the theme for another transition. In-between transitions for extending each rheme may follow their own patterns.

Daneš (1974) notes that one of the crucial properties of these patterns is their missing link. For example, in Pattern 1 there is no link between the earliest and latest text units. Those are connected because of the middle unit that makes transitions between themes and rhemes smoother. In contrast, all text units in Pattern 3 are linked to each other because they all have T_1 as shared given information.

2.3 Evaluation

The goal of the research presented in this dissertation is to provide an approach to coherence modeling and compare it with other models. In order to accomplish this, it is essential to have a method to evaluate the performance of coherence models. In this section, we complete our definition by describing the evaluation methods we employ for assessing coherence models that are examined in our experiments.

2.3.1 Intrinsic vs. Extrinsic

Intrinsic and extrinsic are two types of evaluation methodologies for computational methods. In an intrinsic evaluation, system outputs are directly evaluated in terms of a set of norms or predefined criteria about the desired functionality of the system itself. In an extrinsic evaluation, system outputs are assessed on their impacts on a task external to the system itself.

Some research papers on local coherence modeling use intrinsic evaluation approaches such as sentence ordering (Lapata, 2003; Mihalcea and Tarau, 2004; Karamanis et al., 2004; Barzilay and Lee, 2004; Barzilay and Lapata, 2008). Such an evaluation method is primarily designed to model violations of restrictions in centering theory (Karamanis et al., 2004). The goal of the sentence ordering task is to check if a coherence model can recognize the original order of sentences in a text as the best order of its sentences. The main underlying assumption for this task is that perturbing the order of sentences in a text disturbs its coherence. However, datasets used for this task are artificially created (Lai and Tetreault, 2018), which eases the task for coherence models.

Some other approaches take the coherence of a text as a factor of the text quality and extrinsically evaluate the coherence model in downstream tasks (Miltsakaki and Kukich, 2004; Yannakoudakis and Briscoe, 2012). In this dissertation, we follow extrinsic evaluation methods, and evaluate our coherence model based on its performance for the readability assessment task and the automatic single document summarization task.

In the readability assessment task (Miltsakaki and Kukich, 2000; Pitler and Nenkova, 2008; Petersen and Ostendorf, 2009; Flor et al., 2013), a coherence model is used to assess the readability of texts. The insight of this task is that coherent texts contain less complexity than other ones; therefore, they are easy to read and understand (Pitler and Nenkova, 2008). We use readability assessment as an extrinsic evaluation task for coherence models.

Automatic text summarization has received a lot of attention by researchers in natural language processing because of its potential for various information access applications. For instance, it is useful for tools that aid users to navigate and digest web content (e.g. news, social media, and product reviews), question answering, and personalized recommendation engines. Single document summarization is the task of producing a shorter version of a text while preserving its information content (Nenkova and McKeown, 2011). A basic approach to single document summarization is extractive, in which a summary is produced by identifying and concatenating the most important sentences in a text. Ideally, information in selected sentences for a summary should be the most important information in the input text. This information, however, should have satisfactory variance (or minimum redundancy), and be presented coherently in the summary to be readable. Developing an extractive summarizer that jointly optimizes these three crucial factors – importance, diversity, and coherence – is a challenging task because the inclusion of relevant sentences relies not only on properties of the sentences themselves, but also the properties of every other sentence in a summary. Moreover, since the length of a summary is limited[7], making a balance between these three factors is difficult. For example, a summarizer may select a sentence which contains less important information in comparison to other sentences just to make other selected sentences coherent.

[7] Forcing summaries to obey a length constraint is a typical setup in summarization as it allows for a fair empirical comparison between different possible outputs. Furthermore, it represents a real world scenario where summaries are supposed to be shown on small screens.

2.3.2 Ranking as Classification

Coherence is not a binary property of a text that either exists or not. It is a comparative attribute of texts: Is a text more coherent than other one? Even for humans, it might be ambiguous to decide if a text is coherent or not; however, they can rank texts with respect to their coherence (Halliday and Hasan, 1976). This fits to the task of text ranking with respect to readability, where a coherence model is evaluated by comparing its rankings with rankings provided by human judges for readability.

From a computational point of view, the core of the evaluation method in this dissertation is a pairwise ranking task: Given a pair of texts, which one is more coherent? For being convenient for machine learning models, the pairwise ranking task is recast as a classification task, where each text pair is associated with a label. The value of the label represents which text in a pair should be ranked higher; we use label $+1$ where the first text in a pair is ranked higher, and -1 otherwise. This binary classification task can be solved by a machine learning approach, such as Support Vector Machines (SVMs) (Bishop, 2006). The details of experimental setups for machine learning models are explained in Chapter 4 and Chapter 5.

2.3.3 Evaluation Metrics

In order to perform a quantitative analysis of labels predicted by a coherence model for text pairs, we employ different metrics.

For the readability assessment task we use accuracy and F1-measure. Accuracy quantifies how often a coherence model makes a correct decision on text pairs in test data. A decision is correct if the label predicted by a model for a text pair is identical with the label that is assigned by human judges.

F1-measure is the harmonic mean between precision and recall. Precision is the ratio of the number of correct predictions over the number of all predicted labels. If a model predicts a label for each text pair in test data, then precision and accuracy are identical. However, it is also likely that a model does not rank a pair of texts and sees the texts equally coherent. In such cases, precision and accuracy are not identical. Recall is the number of correct predictions among the number of pairs with the desired label in test data.

For the summarization task, we use ROUGE metrics to evaluate the performance of the examined summarizers. ROUGE is a standard metric for text summarization. It compares a summary generated by a summarizer with a gold summary, which usually is generated by a human, based on word overlaps between summaries. We explain these metrics in more detail in related chapters of this thesis.

3 Related Work

The task of coherence modeling has received much attention due to its significant impact on other natural language processing tasks. In this dissertation, we propose a novel approach to local coherence modeling based on a graph representation of texts. We apply our approach to two types of graphs. The first type captures coreference relations among entity mentions in a text. The second one captures lexical relations among words in a text. We evaluate our coherence model by examining its impact on readability assessment and text summarization.

Accordingly, we first review entity-based approaches to local coherence (Section 3.1). We then survey lexical approaches (Section 3.2). Finally, we review approaches that use a local coherence model for the readability assessment and summarization tasks (Section 3.3).

3.1 Entity-based Approaches to Local Coherence

The preliminary steps of the research presented in this dissertation are inspired by entity-based coherence models. In this section, we explain the details of two popular entity-based models: the entity grid model and the entity graph model. We also discuss their extensions.

Historical review. Entity-based approaches to local coherence modeling have a long history within the linguistics literature (Kuno, 1972; Halliday and Hasan, 1976; Prince, 1981; Joshi and Weinstein, 1998). Most approaches share a common primary assumption: Coherence is perceived based on how entities are introduced and discussed within a text (Barzilay and Lapata, 2008). Texts that keep referring to similar entities are supposed to be more coherent than those with random and unexpected switches from one entity to another. Different linguistic theories support this premise. One of them is centering theory (Grosz et al., 1995; Joshi and Weinstein, 1998), which is discussed in Chapter 2.

A great deal of research has been devoted to implementing centering theory directly (Miltsakaki and Kukich, 2000; Karamanis et al., 2004). However, it is a challenging task because computational models need to determine how to instantiate the parameters of the theory as they are often underspecified. Interestingly, Poesio et al. (2004) noticed that even for basic parameters of centering theory such as “utterance”, “realization”, and “ranking”, multiple interpretations have been developed, as centring theory does not formulate its parameters explicitly. For example, in some studies entities are ranked with respect to the grammatical roles of their mentions (Brennan et al., 1987; Grosz et al., 1995) whereas in some other studies entities are ranked with respect to the position of their mentions (Prince, 1981). In some other studies they are ranked concerning their familiarity status (the thematic role) (Strube and Hahn, 1999; Moens, 2008). Therefore, two instantiations of the same theory make different predictions for the same input.

Another vein of research tries to avoid this by finding an instantiation of parameters so that the parameters are the most consistent with observable data (Strube and Hahn, 1999; Karamanis et al., 2004; Poesio et al., 2004). Some others adopt a specific instantiation in a way that the performance of the coherence model improves for a specific task. For example, Miltsakaki and Kukich (2000) annotate a corpus of student essays with entity transition information and then show that the distribution of transitions correlates with human grades. Analogously, Hasler et al. (2003) investigate whether centering theory can be used in evaluating the readability of summaries, which were produced by humans or machines, by annotating them with the entity transition information. Poesio et al. (2004) demonstrate that the predictive power of the models that directly implement centering theory is highly sensitive to their parameter instantiations, no matter for which task such instantiations are specified.

Barzilay and Lapata (2005, 2008) propose a general framework for coherence modeling. The primary goal of this framework is to eliminate the need for human annotations for parameters in centering theory, regardless of what the evaluation task is. Inspired by that theory, this model hypothesizes that the distribution of entities within coherent texts reveals certain regularities that make these texts distinguishable from incoherent ones. Machine learning approaches can learn these regularities. Since the entity grid model has been the core of many research papers in the area of coherence modeling (including the research presented in this dissertation), we explain details of this model.

3.1.1 The Entity Grid Model

Barzilay and Lapata (2005, 2008) are the first researchers who proposed a general computational approach to local coherence modeling based on the entity relations across adjacent sentences. Supported by some linguistic work such as centering theory (Grosz et al., 1995) and other entity-based theories of text (Prince, 1981), they assume that the distribution of entities within coherent texts exhibits certain regularities that can be reflected in a grid topology, which is named “entity grid”. In this dissertation, we refer to this model as the entity grid model because its main idea is to represent the distribution of entities (see Chapter 2 for the definition of entity) across sentences in a text via a grid. In practice, mentions of an entity are linked in order to show that they are referring to the same entity. Connections between mentions not only show that they are used to refer to the same entity, but also indicate that sentences that contain those mentions are (almost) about the same topic or information (Barzilay and Lapata, 2008). So entity coreference relations can be seen as signals for local coherence.

3.1.1.1 Text Representation: Entity Grids

In the entity grid model, each text is represented by a grid, which is a two-dimensional array, whose rows correspond to entities and whose columns correspond to sentences. An entry $r_{i;j}$ in a grid describes the grammatical role of entity i in sentence j if the entity is mentioned in the sentence. The grammatical roles are categorized as subject (S), object (O), or all other grammatical roles (X). Besides, if an entity is not mentioned in a sentence, a special marker (-) fills the corresponding entry $r_{i;j}$ in the grid. Finally, if a sentence contains several mentions of one entity, the corresponding entry describes the most important grammatical role of the mentions in the sentence: subject if possible, then object, or finally other.

The discussion of entity grids develops around two essential questions: Which textual units should be considered mentions of an entity? How should different mentions be linked to represent an entity? A perfect solution in this regard would use a coreference resolution system to recognize mentions, to link arbitrary mentions to the same entities, and to discard noun phrases which do not correspond to an entity. Since coreference resolution systems are far from perfect, and tend to work even more poorly on incoherent texts, this approach is not generally the one utilized. Moreover, a non-perfect coreference resolution system introduces more noisy connections to a coherence model than what it fixes (Barzilay and Lapata, 2008). As an alternative, implementations of the entity grid model tend to employ all noun phrases as mentions and apply a heuristic coreference resolution to them. This coreference model connects all mentions that have an identical head noun as one entity. However, such a coreference resolution system is quite strict and straightforward. Detailed discussions of this heuristic are given in Poesio and Kabadjov (2004) and Elsner and Charniak (2010).

Example (7) shows a sample text[1]. In this example, noun phrases are marked with brackets as an indication of mentions. Mentions in a sentence are associated with their grammatical

[1] The text with ID D31010, taken from the Document Understanding Conference (DUC 2002) dataset, which we use in one of our summarization experiments. Numbers are not marked because they are filtered out in preprocessing.

roles in the sentence, annotated with a letter (S: subject, O: object, and X: others) next to the brackets.

(7)

s_0: [An arctic cold wave]$_\mathrm{S}$, [the worst]$_\mathrm{X}$ in [10 years]$_\mathrm{X}$, hit [parts]$_\mathrm{O}$ of [Europe]$_\mathrm{X}$, bringing [sub-zero temperatures]$_\mathrm{O}$ and killing [scores]$_\mathrm{O}$ of [people]$_\mathrm{X}$.

s_1: Hardest hit were [Poland]$_\mathrm{S}$, [Bulgaria]$_\mathrm{S}$, and [Romania]$_\mathrm{S}$ as well as [parts]$_\mathrm{S}$ of [central]$_\mathrm{X}$ and [eastern France]$_\mathrm{X}$.

s_2: In [Poland]$_\mathrm{X}$, [three weeks]$_\mathrm{X}$ of [sub-zero temperatures]$_\mathrm{X}$ killed [at least 85 people]$_\mathrm{O}$ in [November]$_\mathrm{X}$, 29 more than in [all]$_\mathrm{X}$ of [the previous winter]$_\mathrm{S}$.

s_3: [Most]$_\mathrm{S}$ of [the victims]$_\mathrm{X}$ were homeless [whose deaths]$_\mathrm{X}$ by [exposure]$_\mathrm{X}$ were alcohol related.

s_4: [Blizzards]$_\mathrm{X}$ and [cold temperatures]$_\mathrm{S}$ also hit [Bulgaria]$_\mathrm{X}$ and [Romania]$_\mathrm{O}$, stranding [hundreds]$_\mathrm{O}$ in [their cars]$_\mathrm{X}$.

s_5: Elsewhere, [snow]$_\mathrm{S}$ blanketed [the Italian island]$_\mathrm{O}$ of [Capri]$_\mathrm{X}$ for [the first time]$_\mathrm{X}$ in [10 years]$_\mathrm{X}$.

The corresponding entity grid for the text that is shown in Example (7) is presented in Table 3.1. For constructing this grid, we follow Barzilay and Lapata (2005, 2008) and consider head nouns of noun phrases to represent the entities. The coreferent mentions are detected by string matching over head nouns.

It is worth noting that although an entity in the original version of the entity grid is indicated by the head of a noun phrase, Elsner and Charniak (2011a) show that adding non-head nouns of a noun phrase to a grid improves the representation power of the entity grid. This enables the model to involve both head nouns and pre-modifiers in noun phrases to link sentences. Therefore, Elsner and Charniak (2011a) consider all nouns as entities in the entity grid representation. The non-head mentions are given the role X.

3.1.2 Pattern Definition: Grammatical Transitions

The key hypothesis in the entity grid model is that the way that entities are distributed as well as the way that the grammatical roles of entity mentions change through a text reveal similar patterns in coherent texts. Barzilay and Lapata (2005, 2008) define all possible transitions that may occur for an entity in a text as patterns.

More concretely, they define a transition pattern as a sequence of symbols, which are employed to demonstrate grammatical roles of mentions, with size n, i.e., $\{S, O, X, –\}^n$.

Entity	s_0	s_1	s_2	s_3	s_4	s_5
WAVE	S	-	-	-	-	-
WORST	X	-	-	-	-	-
YEARS	X	-	-	-	-	X
PARTS	O	O	-	-	-	-
EUROPE	X	-	-	-	-	-
TEMPERATURES	O	-	X	-	S	-
SCORES	O	-	-	-	-	-
PEOPLE	X	-	O	-	-	-
POLAND	-	O	X	-	-	-
BULGARIA	-	X	-	-	-	-
ROMANIA	-	X	-	-	O	-
CENTRAL	-	X	-	-	-	-
FRANCE	-	X	-	-	-	-
NOVEMBER	-	-	X	-	-	-
WEEKS	-	-	S	-	-	-
ALL	-	-	X	-	-	-
WINTER	-	-	X	-	-	-
MOST	-	-	-	S	-	-
VICTIMS	-	-	-	X	-	-
DEATHS	-	-	-	X	-	-
EXPOSURE	-	-	-	X	-	-
BLIZZARDS	-	-	-	-	S	-
HUNDREDS	-	-	-	-	O	-
CARS	-	-	-	-	X	-
TIME	-	-	-	-	-	X
SNOW	-	-	-	-	-	S
ISLAND	-	-	-	-	-	O
CAPRI	-	-	-	-	-	X

Table 3.1: The entity grid representation of the text presented in Example (7). The rows represent entities, and the columns encode sentences. If an entity is mentioned in a sentence, the corresponding entry in the grid indicates the grammatical role of the mention in the sentence (S: subject, O: object, X: others, and “–”: none).

Each pattern represents entity occurrences between sentences and the way that their grammatical roles in n adjacent sentences change. For instance, for two adjacent sentences ($n = 2$) there are 16 possible patterns. These patterns are shown in Table 3.2.

S S	S O	S X	S –	O S	O O	O X	O –	X S	X O	X X	X –	– S	– O	– X	– –

Table 3.2: Transition patterns that are defined in the entity grid model. These patterns represent all possible entity occurrences in two adjacent sentences. Symbols S (subject), O (object), and X (others) show the grammatical role of an entity in a sentence. Symbol "–" encodes that an entity is not mentioned in a sentence.

Each pattern that is shown in Table 3.2 represents one possible way in which an entity may occur in two adjacent sentences. For example, pattern "S O" encodes that an entity appears in two adjacent sentences, and its grammatical role is changing from subject to object across sentences. As another example, consider pattern "S –". It indicates that an entity is referred to by a mention in the subject position of a sentence, and that the entity has no mention in the immediately following sentence.

3.1.2.1 Coherence Representation: Probabilities of Transitions

The entity grid model revolves around the assumption that coherent texts reveal certain regularities over the frequency of transitions or patterns (Barzilay and Lapata, 2005, 2008). The frequency of patterns can be used as an indicator of the preference of coherent texts in using or avoiding certain transitions. However, in order to prevent the model to be biased towards the text length, the probability of each pattern, rather than its raw frequency, is computed. More formally, given the entity grid representation of a text, the probability of a transition pattern occurring in the grid is computed as follows:

$$p(t) = \frac{n(t)}{n(t^{*})}, \tag{3.1}$$

where t is a transition, $n(t)$ indicates the number of times that transition t occurs in the entity grid, and denominator $n(t^{*})$ depicts the number of occurrences of all patterns whose length is as same as the length of t in the grid. For instance, consider the grid in Table 3.1; the probability of pattern "O O" is $.01$, which is computed as a ratio of its frequency, i.e. 1, divided by the total number of patterns of length two, i.e., 140. Therefore, the coherence of a text can be represented by the distribution of patterns in the text. The entity grid model captures frequencies of entity transitions in the entity grid representation of a text with a

SS	SO	SX	S–	OS	OO	OX	O–	XS	XO	XX	X–	–S	–O	–X	––
.00	.00	.00	.04	.00	.01	.01	.04	.00	.00	.00	.12	.04	.04	.11	.60

Table 3.3: Probabilities of the entity transition patterns, which are introduced in Table 3.2, in the entity grid shown in Table 3.1.

vector, which represents the coherence of the text. This vector can be interpreted as a feature vector for the coherence of the text, where each feature is the probability of a pattern in the grid representation of the text. Table 3.3 shows the feature vector representation of the grid presented in Table 3.1 using all transitions of length two.

Centering theory and its extensions define and rank transistions for center shifts through a text. The key advantage of the entity grid model is that it does not define any preference over transitions. It just computes probabilities of patterns in entity grids and defines them as features. The ranking or any other interplay among these features are learned by a machine learning model such as support vector machines.

In summary, given a dataset consisting of texts with different degrees of coherence, the entity grid model captures the coherence of each text in the dataset with a vector of transition probabilities. These vectors are supplied to machine learning models to distinguish texts concerning their coherence. Machine learning models automatically learn how coherence patterns should interact with each other to accomplish the final evaluation task.

3.1.2.2 Extensions of the Entity Grid Model

There are several extensions to the entity grid model in the literature. They mostly extend the entity grid model in two ways. Some employ different approaches for entity identification, and some others use various linguistic information about entities to fill entries of grids.

Filippova and Strube (2007) extend the entity grid approach by grouping all entities that are semantically related. They demonstrate that by grouping related entities, the performance of the entity grid model improves, especially when syntactic information is not involved. They use WikiRelate (Strube and Ponzetto, 2006) to compute relatedness between entities, $SemRel(e_i, e_j) > t$, where t is a threshold. Different values of t result in different grid densities. For small values, a grid is dense since many entities are grouped into one.

Elsner and Charniak (2008) employ the information status, i.e. new or the first mention vs. given or subsequent mentions, of entities, rather than grammatical roles. They run a maximum-entropy classifier to assign each noun phrase (i.e. mention) a label $L_{np} \in \{new, old\}$.

The coherence score of a text is then estimated by the product of probabilities over the information status of each mention. They show that adding such a classifier, which distinguishes discourse-new entities from discourse-old ones, improves the performance of the entity grid model, which uses grammatical role information. Another finding of their work is that incorporating pronouns in the entity definition phase of the model enhances the entity grid representation, and consequently, the performance of the coherence model. Indeed, pronoun resolution systems, as they are highly precise but specific coreference resolution systems, can be used to acquire more meaningful references to entities.

Elsner and Charniak (2011b) extend the entity grid model by distinguishing between important and unimportant entities. The motivation of their work is that the standard entity grid model uses no information about the entity itself in transitions; the probability of a transition is the same regardless of the entity that is under discussion. In order to involve information about entities, they associate each entity with some features, e.g., Is_Named_entity, Has_Singular_Mention, Has_Proper_Mention, and the like. They show that by distinguishing salient entities from other ones, the discriminative performance of the entity grid model improves.

Lin et al. (2011) use the grid representation, i.e. a two-dimensional matrix, but instead of modeling entity transitions, they model discourse relation transitions between sentences. The grid is filled in by discourse relations, which connect a term in a sentence with other sentences. Then, similar to the entity grid model the probabilities of transitions are used to represent the coherence of a text. In a follow-up paper, Feng et al. (2014) train the same model but use features derived from deep discourse structures (as presented in Penn Discourse Treebank (Prasad et al., 2008)) annotated with Rhetorical Structure Theory (RST) relations (Mann and Thompson, 1988). Early RST-based models include Marcu (1997) and Mellish and Dale (1998), which focus on coherent text generation rather than coherence evaluation.

Tien Nguyen and Joty (2017) propose a deep learning model to learn patterns in the entity grid representation of text. Their model first transforms grammatical roles in an entity grid into vector representations and then supplies them to a convolution operation to model entity transitions in a distributed space. The max-pooled features from the convoluted features are used for coherence scoring. This model limits relations between sentences to entities that are shared by sentences, which makes its performance dependent on the performance of other tools like coreference resolution systems and syntactic parsers. In a later work, Joty et al. (2018) extend their neural entity grid model by lexicalizing its entity transitions such that each entry of the entity grid contains two vectors, one representing its corresponding lexicon and one representing the grammatical role of the entity in the corresponding sentence.

Li and Hovy (2014) model sentences as vectors derived from recurrent neural networks (Goldberg, 2017) and train a feed-forward neural network that takes an input window of sentence vectors and assigns a probability which represents the coherence of the sentences in the window. Text coherence is evaluated by sliding the window over sentences and aggregating their coherence probabilities. Similarly, Li and Jurafsky (2017) study the same model at a larger scale and use a sequence-to-sequence approach in which the model is trained to generate the next sentence given the current sentence and vice versa. Our approach differs from these methods because it takes distant relations between words in a text into account as it is built on graph representations of texts.

To conclude this part, we point out the advantages and disadvantages of the entity grid model. The main benefit of the entity grid model is that it learns the properties of coherent texts, which are represented by patterns of entity distributions, from a corpus of texts without recourse to manual annotations or a predefined knowledge base. However, the main limitation of the entity grid model is that it only takes into account relations between adjacent sentences, while in many cases adjacent sentences do not have any entities in common. For example, in an investigation on texts in the CoNLL 2012 dataset (Pradhan et al., 2012), it is shown that 42.34% of adjacent sentences do not share any common entities (Zhang et al., 2015). Moreover, non-adjacent sentences can be related to each other as well. The entity grid model does not model such relations mainly because its grid representation cannot capture long-distance relations. It is worth noting that increasing the sequence length of grammatical transitions does not lead to incorporating long-distance relations between sentences. In practice, the length of sequences has never been fixed to a value higher than two. The reason is that enlarging the length of sequences increases the number of transitions, many of which do not frequently occur in texts. As a result, many transitions have zero probability in feature vector representations of the coherence of texts. This problem is known as "sparsity" in statistical machine learning. We discuss more about this problem in Chapter 5.

3.1.3 The Entity Graph Model

The entity graph model (Guinaudeau and Strube, 2013) represents entity-based relations between sentences in a text with a graph[2]. Graphs are capable of spanning the entire text and capture connections between any two sentences in a text. Moreover, an advantage of formulating a problem, like coherence modeling, with graphs is that standard algorithms in graph theory can be employed to solve the problem. It is sufficient to encode a problem with graphs

[2] We formally define the graph concepts that are required for the research in this thesis in Chapter 4.

and then choose a proper solution from graph theory to solve the problem. Here, we review how Guinaudeau and Strube (2013) use a graph to represent the distribution of entities through a text, and how they use such a representation to formulate, and then solve the task of coherence modeling using connectivity measures in graph theory.

3.1.3.1 Text Representation: Entity Graphs

The entity grid representations of texts are mostly sparse, which means many entries in grids are "–". This happens because each sentence in a text contains a few entities out of all entities mentioned in the text. Graphs can deal with this sparsity issue in grid representations. Guinaudeau and Strube (2013) propose to recast the entity grid representation of a text by a graph representation. We refer to this representation as the entity graph because it captures the distribution of entities through a text via a graph. Figure 3.1 depicts the entity graph representation of the text in Example (7), where the graph is constructed based on the entity grid shown in Table 3.1.

The idea is that the entity grid representation can be taken as the incidence matrix[3] of a bipartite graph, which consists of two disjoint sets of nodes. Node sets in the entity graph representation correspond to rows and columns in the entity grid representation. One set consists of nodes associated with entities, and the other set consists of nodes associated with sentences. Edges in the entity graph encode entries in the entity grid such that if a sentence contains a mention of an entity, then an edge connects the associated node with the sentence and the associated node with the entity in the entity graph. Therefore edges in the entity graph are equivalent to entries in the entity grid that are not equal to "–". The value of other entries in the entity grid are encoded as edge weights in the entity graph. More concretely, the grammatical role of an entity in a sentence is encoded in the entity graph by the weight of the edge that connects the entity node to the sentence node. Given the linguistic intuition that entities with important grammatical roles are prominent entities in each sentence, three numbers $3 > 2 > 1$ are used to model subject (S), object (O), and any other grammatical roles (X), which are employed by the entity grid model.

[3] The incidence matrix, or the adjacency matrix, of a graph is a two dimensional matrix A with binary elements. An entry is 1 if there is an edge between the nodes corresponding to the row and the column of the entry; otherwise, the value of the entry is 0.

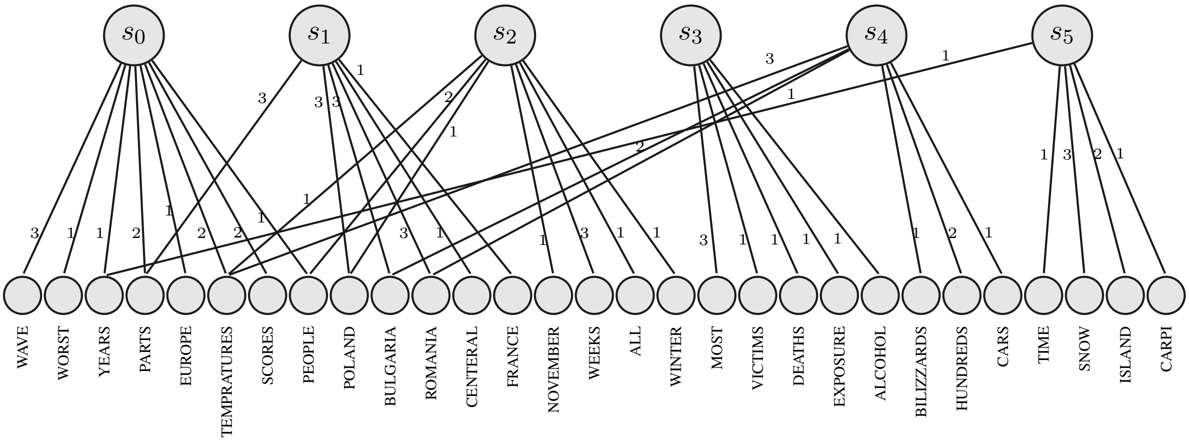


Figure 3.1: The entity graph representation of the text from Example (7). The graph is obtained from the entity grid representation shown in Table 3.1. The top nodes represent columns in the grid or sentences in the text. The bottom nodes capture rows in the grid or entities in the text. Edges encode the entries in the grid. Weights of edges represent the value of each entry in the grid: 3:S, 2:O, 1:X, and 0:–. The weight of 0 is equivalent to no edge in a graph, so they are not drawn in the graph.

3.1.3.2 Coherence Measurement: The Average Outdegree of Projection Graphs

Projection graphs. Local coherence is about the connectivity among sentences in a text. However, nodes in the entity graphs consist of two disjoint node sets, one of which represents sentences. The other set captures entities. Guinaudeau and Strube (2013) propose to transform an entity graph to a graph whose nodes capture only sentences and whose edges encode entity-based relations among sentences. Such a graph, which is obtained from the entity graph as a bipartite graph, is called a "one-mode projection graph" (or a projection graph for the sake of brevity) in graph theory (Newman, 2010). Edges in projection graphs are weighted in different ways in order to retain specific information about relations between sentence nodes and entity nodes in entity graphs. Moreover, edges in projection graphs are directed to encode the order of sentences in texts.

Guinaudeau and Strube (2013) apply three kinds of projections, namely P_U, P_W and P_{Acc}. Figure 3.2 shows these graphs obtained from the entity graph presented in Figure 3.1. These projection graphs differ in the weighting scheme that they use:

- In P_U, weights are binary, i.e., 0 or 1. The weight of an edge between two nodes in this type of projection graphs is equal to 1 if the corresponding sentence nodes are connected to at least one entity node in the entity graph. This projection graph merely captures which sentences are linked to each other in a text.

- In P_W, an edge is weighted according to the number of the entity nodes that are connected with both sentence nodes in the entity graph. In other words, the weight of an edge between two nodes in this type of projection graphs represents the number of shared entities by the corresponding sentences. This projection graph not only models that sentences in a text are connected, but also captures how strongly they are connected. It takes the number of common entities between a pair of sentences as the strength of the relation between sentences.

- In P_{Acc}, grammatical information is accounted for by integrating the edge weights in the entity graph. In this case, the weight of the edge between nodes s_i and s_k is equal to

  $$W_{ik} = \sum_{e \in E_{ik}} w(e, s_i) \cdot w(e, s_k), \tag{3.2}$$

  where E_{ik} is the set of the entity nodes that are connected to both s_i and s_k in the entity graph. This type of the projection graph incorporates grammatical information about entities shared by sentences in order to measure the strength of the relation between sentences.

Distances between sentences can be integrated into the weighting schemes of edges in projection graphs to decrease the importance of links between non-adjacent sentences (Guinaudeau and Strube, 2013). In this case, edge weights in projection graphs are divided by the difference between sentence IDs.

The average outdegree as a coherence feature. Given a projection graph representation of a text the coherence of the text is measured based on the connectivity of nodes in the projection graph. Guinaudeau and Strube (2013) define a coherence metric given the assumption that projection graphs of coherent texts contain more edges than projection graphs of incoherent ones. They propose to use a centrality metric (Newman, 2010) from graph theory to measure to what extent nodes in a graph are connected with each other. Let $outDegree(s)$ be the sum of the weights associated to edges that leave node s in projection graph P, then the centrality metric of the projection graph is computed by the average outdegree of all nodes (N) in the graph:

$$AvgOutDeg(P) = \frac{1}{N} \sum_{i=0}^{N-1} outDegree(s_i). \tag{3.3}$$

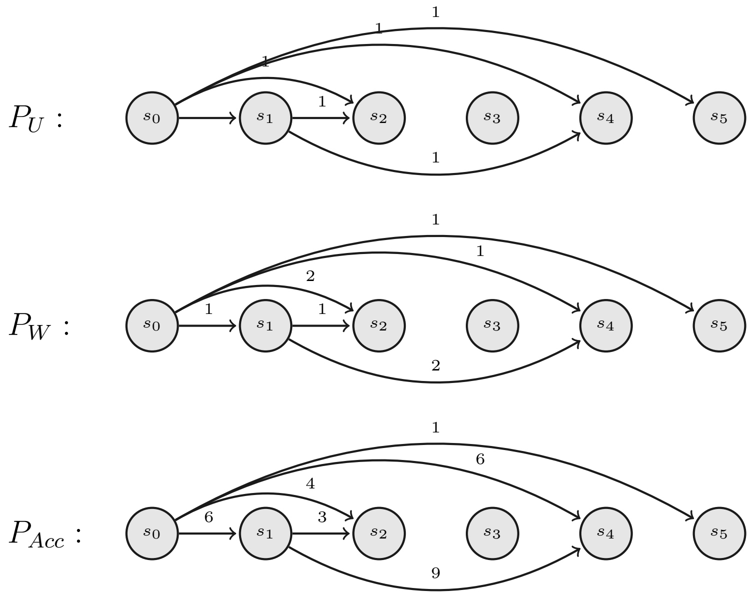


Figure 3.2: Three types of projection graphs that are employed by the entity graph model. P_U shows only which sentence nodes are connected. P_W takes the number of shared entities as the weight of edges. P_{Acc} involves grammatical roles of entities shared by sentences.

Table 3.4 shows the AvgOutDeg for different projection graphs presented in Figure 3.2 with and without incorporating the distance information.

P	$AvgOutDeg(P)$
P_U	$\frac{1}{6}\left((1+1+1+1)+(1+1)+(0)+(0)+(0)+(0)\right))=1.00$
P_W	$\frac{1}{6}\left((1+2+1+1)+(1+2)+(0)+(0)+(0)+(0)\right))=1.33$
P_{Acc}	$\frac{1}{6}\left((6+4+6+1)+(3+9)+(0)+(0)+(0)+(0)\right))=4.83$
P_U, $Dist$	$\frac{1}{6}\left((1+0.50+0.25+0.20)+(1+0.33)+(0)+(0)+(0)+(0)\right))=0.55$
P_W, $Dist$	$\frac{1}{6}\left((1+1+0.25+0.20)+(1+0.66)+(0)+(0)+(0)+(0)\right))=0.69$
P_{Acc}, $Dist$	$\frac{1}{6}\left((6+2+1.5+0.2)+(3+3)+(0)+(0)+(0)+(0)\right))=2.61$

Table 3.4: The average outdegree of nodes in projection graphs presented in Figure 3.2 . Dist. shows when distance is integrated in edge weights.

In order to rank a pair of texts with respect to their coherence, Guinaudeau and Strube (2013) represent both texts with the same type of projection graphs, and then use the average outdegree of their projection graphs to compare texts. It is worth to mention that the described entity graph model is an unsupervised model as average outdegrees are directly employed for comparing texts with respect to their coherence. However, the average outdegree captures no information about the connectivity style of nodes in a projection graph. For example, consider the two projection graphs that are shown in Figure 3.3. These two graphs have the same average outdegree, i.e. $\frac{5}{6}$, but graph (b) is disconnected because node s_2 is connected to none of its previous nodes. Consequently, its corresponding text is less coherent than the text associated with (a) (Karamanis et al., 2009). The outdegree does not capture such information.

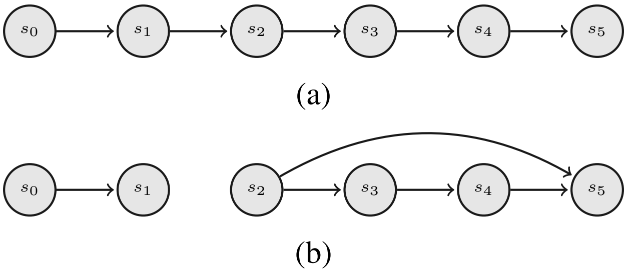


Figure 3.3: (a): A projection graph with the total outdegree of five where all nodes are in one component; (b): A projection graph with the total outdegree of five where nodes are in two components.

Moreover, the three types of projection graphs behave differently for different tasks examined by Guinaudeau and Strube (2013). So it is not clear which type of projection graphs is more useful for a downstream task.

3.1.3.3 Extensions of the Entity Graph Model

The entity graph model is extended from two different perspectives: the method that is used to represent texts with graphs and the graph metric that is employed to measure coherence.

Dias and Pardo (2015) propose to fill in the grid based on the RST relations between sentences in a text. An entry in the entity grid is 1 if an entity is part of a sentence that participates in an RST relation. Based on such a grid representation, they define a bipartite graph similar to the entity graph and then construct its projection graph to model relations among sentences. They use the average outdegree metric of projection graphs to measure the coherence of a text. Their model outperforms the entity graph model. However, similar to RST-based extensions of the entity grid model, obtaining RST relations is subjective, and human annotations or discourse parsers are not available for many languages.

Petersen et al. (2015) use several graph metrics, rather than the average outdegree, to approximate different aspects of the text flow that can indicate coherence. These metrics are designed to capture more information about the connectivity style of nodes in projection graphs. For example, they leverage the mean of the PageRank scores (Newman, 2010) of nodes in a projection graph for distinguishing a star-graph, in which all nodes are connected to one node, and no other edges occur, and a path graph, in which all nodes occur in a chain. Some other assessed metrics include a clustering coefficient, which measures to what extent neighbors of a node are connected among themselves; and betweenness, which is the fraction of shortest paths that contain a node. Although their results are better than the original entity graph which uses the average outdegree, the difference is not substantial enough to consider these metrics. We admit that the average outdegree is quite straightforward and efficient to compute.

Zhang et al. (2015) use semantic relations between entities to identify not only the mentions that refer to the same entity but also the mentions that refer to entities which are semantically related. They capture such semantic relations by leveraging WordNet (Baccianella et al., 2010) as a knowledge base. By incorporating such relations, the performance of the entity graph model improves. They also challenge the average outdegree metric and propose to combine this metric with another score which is named "reachability". The reachability score is the sum of the weights of edges in the shortest path that starts from the first sentence node and ends at the current sentence node. The intuition behind the reachability score is that this score reflects the tightness between a sentence in a text and its previous context in the text. In this way, they overcome some weaknesses of the average outdegree but not all of them.

We propose[4] an extension of the entity graph model by taking the entity and sentence importance into account (Mesgar and Strube, 2014). We reflect the connectivity structure of an entity graph into its edge weights by applying a normalization method to the weights. The normalization method reduces the differences in performance of three types of projection graphs.

3.2 Lexical Approaches to Local Coherence

Local coherence is an essential factor in text comprehension. It is about the extent to which sentences in a text are linked together. Halliday and Hasan (1976) emphasize the role of lexical cohesion in connecting sentences in a text to each other. They consider several linguistic devices – repetition, synonymy, hyponymy, and meronymy – which contribute to the "continuity of lexical meaning" observed in a coherent text. In this section, we mainly survey the computational models that use lexical relations for modeling local coherence. However, since these models are based on the lexico-semantic relations between words in a text, we first

[4] We just briefly explain this model here because it does not focus on coherence patterns which are the core of the research presented in this thesis.

discuss different resources that are used in the literature to recognize such relations.

3.2.1 Lexical Resources

Computational models in natural language processing that are built upon lexical-relations, including coherence models, crucially rely on the existence of resources that encode information about semantic relations between words in a language. Such resources are typically acquired via two main approaches: the knowledge-based approach (or top-down) where humans manually solicit such information, and the corpus-based approach (or bottom-up) where information is automatically learned from corpora. Although the latter has gained ground during the last decades due to the availability of large amounts of texts and increased computing capacities, the former remains fundamental because it allows us to collect reliable, fine-grained, and explicit information.

Knowledge-based resources. One of the fundamental lexical knowledge resources for English is the Princeton WordNet (Fellbaum, 1998). WordNet aims to represent real-world concepts and their relations similar to what humans perceive about them. WordNet covers about 20 million instances of concepts and relations extracted from raw texts. Nouns, verbs, adjectives, and adverbs are each organized into networks of synonym sets, which is called “synsets”. Each synset represents one lexical concept and a variety of its relations. The WordNet-based similarity measures have been shown to correlate with human similarity judgments reliably. WordNet has been used in a variety of applications, ranging from malapropism detection to word sense disambiguation (Budanitsky and Hirst, 2006). The Princeton WordNet for English inspired the creation of lexical knowledge bases in other languages such as German, which is called GermaNet (Hamp and Feldweg, 1997). However, WordNet is not available for many languages because it requires a lot of human effort and knowledge for annotation.

YAGO (Hoffart et al., 2013) is another example of top-down knowledge resources. It consists of four million instances of concepts and relations where the instances are automatically extracted from online encyclopedias such as Wikipedia (Denoyer and Gallinari, 2006) and FreeBase (Bollacker et al., 2008). Relations then are edited by human experts. Generally speaking, manually defined knowledge bases, like WordNet, have better accuracy but lower coverage, while automatically extracted knowledge bases, like YAGO, are the opposite.

Although different coherence models have employed these knowledge bases, there are weaknesses with these resources. Zhang et al. (2015) describe two main issues in retrieving knowledge about words from such resources as follows:

- knowledge source: Which resource is the best for obtaining this knowledge?
- knowledge selection: How do we pinpoint the most relevant entry in a knowledge base?

Knowledge resources cover semantic relations between particular sets of word categories. For example, WordNet is designed to provide complete coverage of common open-class English words. Therefore, it has little or no coverage of vocabulary from specialized domains and has minimal coverage of proper nouns. This issue may hinder its application to domain-specific contexts and tasks which require to deal with proper nouns. The issue related to knowledge selection refers to how we should retrieve knowledge instances. Should we use exact or partial matching of words? The chance of exact matching of words (especially entities) in a text with instances in a knowledge base is low. In contrast, partial matching between arguments and entities usually increases coverage but at the risk of introducing some noise. Methods developed for word sense disambiguation may solve the above problems but they may bring difficulties on their own.

Finally, regardless of which knowledge resource is employed, a similarity metric is required to quantify if two words are semantically related or not. For these knowledge resources, a simple way to compute the semantic relations between words is to view the knowledge base as a graph. The semantic relatedness can be measured based on graph properties such as the path length between the words (Budanitsky and Hirst, 2006). The shorter the path between two word nodes, the more similar the words are.

Corpus-based resources. The bottom-up knowledge resources are obtained based on the co-occurrence of words in texts. In these resources, words are taken as similar if they frequently occur in a similar context. The main idea of such resources is to represent semantic properties of words by vectors in a multi-dimensional space. Such vectors are obtained by observing the distributional patterns of word co-occurrences with their nearby words in large bodies of texts. The similarity between word vectors in vector space quantifies the semantic relatedness between words in language space.

Different approaches to learning such vector representations exist. One of the early techniques is "Latent Semantic Analysis" (LSA) proposed by Landauer and Dumais (1997). This method constructs a matrix, namely a co-occurrence matrix, containing word counts per text from a large number of texts. It then uses a mathematical method called "Singular Value Decomposition" (SVD) (Furnas et al., 1988) to reduce the text dimensionality in the co-occurrence matrix while preserving the similarity structure among words.

The other method is named “Latent Dirichlet Allocation” (LDA) proposed by Blei et al. (2003). LDA is a generative statistical model which allows sets of words to be explained by a set of latent topics. Its intuition is that words of texts with similar topics are semantically related. It is worth noting that in this approach, topics are neither semantically nor epistemologically defined. Topics are defined on the basis of automatic detection of the likelihood of term co-occurrence. This method is identical to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a sparse Dirichlet prior. The intuition of sparse Dirichlet priors is that texts cover a small set of topics and those topics are frequently expressed by means of a small set of words. In practice, LDA yields better disambiguation of words and more precise assignments of texts to topics in comparison to LSA.

Finally, recent methods for representing words in a distributional space utilize deep neural networks rather than co-occurrence matrices. In contrast to the above methods (LSA and LDA) which are unsupervised, the neural network models applied for obtaining word vectors are trained in a supervised manner. This property is an advantage for these methods because as the vocabulary in a language grows, new vectors for new vocabulary can be trained and appended to such knowledge resources. The word vectors that are generated by these models are named “word embeddings”. Well-known approaches for obtaining word embedding are “word2vec” (Mikolov et al., 2013) and “GloVe” (Pennington et al., 2014). The word2vec approach focuses on learning the embeddings of a word given its local usage context, where a window of words surrounding the word defines its context. The length of the window is a configurable parameter of the model. Large windows tend to produce more topical similarities, and smaller windows give more functional and syntactic similarities (Goldberg, 2017). The GloVe approach, rather than using a small window to define local context, constructs an explicit word-context or word co-occurrence matrix using statistics across the whole text.

Indeed, distributional representations of words, in general, are beneficial if they are trained on sufficiently large and balanced corpora; otherwise, there is a risk of finding words whose similarity only makes sense in the examined corpus (Lin, 1998b). See Budanitsky and Hirst (2006) where they highlight several problems that arise from the imbalance and sparseness of corpora for such methods. In resources that are provided bottom-up, the cosine of the angle of a pair of word embeddings (or the inner product between the normalizations of the two embeddings) measures the similarity of the two words. In other words, the absolute value of the cosine function over two word embeddings indicates the strength of the semantic relation between the corresponding words. Absolute cosine values near $+1$ represent semantically related words, while values near 0 represent semantically unrelated words.

3.2.2 Lexical Cohesion Models

In this section, we review local coherence models that are built upon lexical cohesion (Halliday and Hasan, 1976), i.e., lexical relations between words. We categorize these approaches into three high-level research trends: models that are based on lexical chains, models that are based on sentence similarities, and models that are based on word distributions.

The first trend of research includes methods that use lexical relations to build lexical chains. A lexical chain is a sequence of semantically related words spanning a topical text unit (Morris and Hirst, 1991). Lexical chaining has a long history in local coherence modeling for different applications (Morris and Hirst, 1991; Feng et al., 2009; Wong and Kit, 2012; Ben et al., 2013; Flor et al., 2013). Morris and Hirst (1991) induce semantic relations from Roget's Thesaurus as a knowledge resource. The thesaurus provides an account of the vocabulary of English, grouped into hierarchical categories. Their central intuition is that coherent texts have a high concentration of dense chains. Therefore, the distribution of lexical chains is a surface indicator of the structure of coherent texts. Galley et al. (2003) construct lexical chains for topic segmentation, which is tightly related to coherence. This model does not need any knowledge resource because it builds lexical chains based merely on word repetitions. In contrast, Stokes et al. (2004) employ WordNet to extract lexical chains from texts. Weak[5] relations between words in lexical chains in a text are used as an indicator of topic shifts. Barzilay and Elhadad (1997) propose a lexical chaining algorithm which uses WordNet, Thesaurus, and Part of Speech (POS) tags to extract lexical relations. They do not use this model directly for coherence modeling, but they utilize it for generating coherent summaries. Somasundaran et al. (2014) use lexical chaining for measuring the coherence of essays written by non-native English students. To do so, they employ Lin's thesaurus (Lin, 1998a) to identify semantically similar words in essays. The main lesson learned from this work is that features related to lexical chains measure the coherence of essays. In order to capture different aspects of a lexical chain, they employ several features such as the number of chains in a text, the length of a chain, and the number of chains with more than one word.

The second trend of research related to lexical cohesion includes papers that consider lexical relations between words of sentences in order to compute the similarity between sentences. The central insight of these approaches, in general, is that sentences of coherent texts are similar because they contain semantically related words. Generally, these models first aggregate word vectors corresponding to words in sentences for representing each sentence via a vector,

[5] The difference between weak and strong relations is identified using a threshold on the employed similarity function.

and then compute the similarity between sentence vectors to measure the similarity between two sentences. Finally, the average of all similarities between adjacent sentences in a text measures the coherence of the text:

$$coh(T) = \frac{\sum_{i=0}^{N-2} sim(s_i, s_{i+1})}{N-1}, \tag{3.4}$$

where $sim(s_i, s_{i+1})$ is a measure of similarity between two adjacent sentences, and N is the number of sentences. This idea is operationalized in different ways. For example, Foltz et al. (1998) employ LSA to represent each word in a sentence by a vector and then use the weighted average of word vectors to obtain sentence vectors. The weighting scheme in their approach is inspired by information retrieval techniques, most notably TF-IDF, where TF stands for the "Term Frequency" and IDF for the "Inverse Document Frequency". The similarity between two adjacent sentences is computed by applying the cosine function to sentence vectors. Higgins and Burstein (2007) apply a similar strategy for essay coherence, but they use a Random Indexing (RI) model to represent sentences. Lapata and Barzilay (2005) compute the similarity between two adjacent sentences by counting the number of exact repetitions between nouns in sentences. Yannakoudakis and Briscoe (2012) represent each sentence by a vector of lemmas and the POS tags of words in sentences, and then the average of the cosine similarities between adjacent sentences encode how coherent a text is. Hearst (1994, 1997) computes the cosine similarity between adjacent windows of words, rather than adjacent sentences.

The third trend of research related to lexical cohesion uses probabilistic models to assign a coherence score to a text. Lapata (2003) proposes to compute the coherence probability between two adjacent sentences based on their lexical relations. This probability for a given pair of sentences is a conditional probability of words in a sentence given all of the words in its immediately preceding sentence. The coherence score of a text is the product of coherence probabilities between adjacent sentences. Although this model does not use any external knowledge resources, it computes its own co-occurrence matrix representation of a text, which captures the distribution of words across adjacent sentences. Moreover, this model learns the order of word pairs in different types of texts, e.g., whether "CAR" precedes "TIRE" more in coherent texts or in incoherent ones. Li and Hovy (2014) propose to represent words in sentences by pre-trained word embeddings. Then a recurrent (Schuster and Paliwal, 1997) or a recursive neural network is used to represent each sentence based on its word embeddings. A window with length three is sliding through a text, and a coherence probability is computed for every three adjacent sentences. The final coherence score of the text is obtained by multiplying these probabilities.

3.3 Applications and Evaluations of Coherence Models

Coherence plays a crucial role in different natural language processing applications such as automatic text summarization (Celikyilmaz and Hakkani-Tür, 2011; Lin et al., 2012; Feng and Hirst, 2012), automatic essay scoring (Miltsakaki and Kukich, 2004; Higgins et al., 2004; Burstein et al., 2010), readability assessment (Pitler and Nenkova, 2008; Wang et al., 2013), and so forth. Each of these downstream tasks can be employed to evaluate a coherence model. In the research presented in this thesis, we focus on readability assessment and document summarization for evaluating our coherence model. Therefore, we review the research related to these two tasks.

3.3.1 Readability Assessment

Readability is a property of a text, which describes how easily the text can be read and understood. Dale and Chall (1949) define readability as

"The sum total (including all the interactions) of all those elements within a given piece of printed material that affect the success a group of readers have with it. The success is the extent to which they understand it, read it at optimal speed, and find it interesting".

Assessing the degree of readability of a text has been a field of research for many decades. Early readability metrics (Flesch, 1948; Kincaid et al., 1975) have been established as a function of shallow features of a text, such as the number of syllables per word and the number of words per sentence. These traditional readability metrics are still used in many settings and domains, mainly because they are very easy and efficient to compute.

Later research has investigated the use of statistical language models (uni-gram in particular) to capture the distribution of vocabulary between two readability grade levels (Si and Callan, 2001; Collins-Thompson and Callan, 2004). This research trend is followed by an investigation on the effect of syntactic features (Schwarm and Ostendorf, 2005; Heilman et al., 2007; Petersen and Ostendorf, 2009) in the assessment of text readability. While language model features alone outperform syntactic features in classifying texts according to their reading grade levels, the combination of these two sets of features performed the best.

However, these features are not sufficient to encode the readability of a text because they never go beyond the level of words and sentences. In order to accurately model the difficulty of a text for its readers, besides the surface features some discourse level features are required. Indeed, well-written texts are more than unrelated sequences of sentences. Discourse-level factors (e.g. coherence) of a text play a critical role in the overall understanding of the text

(Pitler and Nenkova, 2008). In a high-quality text, sentences relate semantically to one another so that they become less ambiguous. Moreover, the relationships among sentences of easy-to-read texts are easy to recognize and interpret for readers. In a well-written text, information smoothly flows sentence by sentence as the text progresses. So, coherence is an essential factor of text quality. Beigman and Flor (2013) use the lexical relations between words of a text to model the quality of the text. The core intuition of their model is that a text segmentation algorithm, which uses information about patterns of word co-occurrences, can detect topic shifts in a text. Eisenstein and Barzilay (2008) state that coherent texts contain some proportions of more highly associated word pairs (those in sentences within the same topical unit) and of less highly associated pairs (those in sentences from different topical units). They illustrate that the distribution of patterns of semantically related words correlate with the writing quality.

Pitler and Nenkova (2008) compare the performance of several feature sets such as the features proposed by the entity grid model to model local coherence and the frequency of the RST relations in a text for the readability assessment task. They employ two approaches. The first approach recasts readability assessment as a rating task (Pitler and Nenkova, 2008; Kate et al., 2010). The requirement for this task is a dataset whose texts accompany ratings assigned by human judges. Several human annotators judge each text for its readability by assigning the text a readability rating on an n-point scale, where n is a design choice. The average of these ratings is then the final readability rating of the examined text. Given such a dataset, a statistical correlation coefficient metric, e.g. the Pearson correlation coefficient, between values of a feature and the average human ratings of texts in a corpus is computed to measure which feature is more correlated with human-provided ratings. The second approach is to distinguish difficult-to-read texts from easy-to-read ones. This approach treats the readability assessment task as a pairwise classification task (Pitler and Nenkova, 2008; Guinaudeau and Strube, 2013; Barzilay and Lapata, 2008): Given a pair of texts, which one is easier to read? In this approach, all related features related to coherence and readability are taken into account to increase the predictive power of the classifier. However, each feature class is also separately used to classify texts. For example, Pitler and Nenkova (2008) show that entity transition features introduced by the entity grid model for coherence modeling are the best category of features to classify texts with respect to their readability.

3.3.2 Automatic Text Summarization

Coherence is a fundamental factor for automatic text generation systems as the output text is supposed to be readable. An example of such systems is an automatic text summarization system. The input to a summarizer is a text (or several texts in the case of multi-document summarization), and the task of the system is to produce a shorter text which contains the gist of the information presented in the input text(s). This output text, of course, should be readable and understandable to be used by humans or other natural language processing applications.

The summarization task has several design choices: single-document vs. multi-document, and extractive vs. abstractive summarization[6] (Hahn and Mani, 2000). Single-document summarization systems take only one input text, whereas multi-document summarizers produce a summary from a cluster of texts. Extractive summarizers (Kupiec et al., 1995; Carbonell and Goldstein, 1998; Gillick et al., 2009) produce a summary by selecting a subset of sentences from an input text and concatenating them, while abstractive summarizers (Wang and Cardie, 2013; Alfonseca et al., 2013) involve the generation of sentences for the summary as well.

The summarization task, in all of its variations, and coherence modeling meet in two general research trends. The first trend, which is called summary coherence ranking (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013), employs a coherence model to discriminate between pairs of summaries generated by either humans or machines. This trend is (almost) similar to the readability assessment task; just examined texts are summaries. In this trend, the performance of a coherence model is assessed by comparing rankings induced by the model against rankings elicited by human judges. A coherence model that exhibits a high agreement with human judges accurately captures the coherence properties of the texts (Barzilay and Lapata, 2008). Although this approach can potentially be used for evaluating the quality of summaries produced by any summarization system, it is mainly used to compare the outputs of multi-document extractive summarization models.

The second trend combines coherence metrics with an automatic text summarization system, with the intention of producing coherent summaries. This approach is more beneficial for the real application of automatic text summarization. For example, some methods (Radev et al., 2004b; Nenkova and Vanderwende, 2005) use advantages of word repetitions, as a cohesive device, in their multi-document extractive summarizers. Some others involve assumptions about the text structure of the input document to a summarizer. Daumé III and Marcu (2002) hypothesize a hierarchical structure (e.g. sections and paragraphs), and Teufel

[6] There is another type of summarization which is called compressive summarization, where summaries are formed by compressed sentences not necessarily extracts (Knight and Marcu, 2000).

and Moens (2002) assume a flat structure for input texts. Teufel and Moens (2002) focus on summarizing scientific articles and use lexical cohesion clues in an article for dividing it into research goal (aim), the outline of the paper (textual), presentation of the paper's contribution (methods, results, and discussion), and presentation of other work (other). These topical segments are not necessarily consistent with the physical structure of an article and might be distributed evenly through the whole article. This strategy is especially fruitful if a summary should contain information about specific parts of a text rather than on the text as a whole. Barzilay and Elhadad (1997) employ lexical chains occurring in a text to divide the text into segments. The use of lexical chains allows topicality to be taken into account to heighten the quality of summaries. Celikyilmaz and Hakkani-Tür (2010, 2011) incorporate the idea of hierarchical topical coherence models for segmentation and sentence extraction. Clarke and Lapata (2010) retain entities which serve as centers of sentences (in the sense of centering theory) in summaries. McKeown et al. (1999) first initially cluster sentences and then choose the representative sentences of each cluster to be added in the summary. In their work, the clustering stage minimizes redundancy, and the representative sentence selection maximizes importance. However, this model does not deal with the problem of coherence within the task of sentence selection. Carbonell and Goldstein (1998) propose a global model through the use of the Maximum Marginal Relevance (MMR) criteria, where the model scores sentences by considering a weighted combination of importance plus redundancy with sentences already in the summary. Summaries are then created with an approximate greedy procedure that incrementally includes the sentence that maximizes this criterion. Greedy MMR style algorithms are still a popular baseline for summarization.

One of the popular approaches to optimizing three main factors for summarization – importance, variance, and coherence – is Integer Linear Programming (ILP). ILP techniques have been used to solve many intractable inference problems in natural language processing applications. Examples include applications to sentence compression (Clarke and Lapata, 2010; Filippova and Altun, 2013), coreference resolution (Denis and Baldridge, 2009), syntactic parsing (Klenner, 2007), as well as semantic role labeling (Punyakanok et al., 2004). In an ILP approach, an objective function and some constraints model the optimization problem. Solving arbitrary ILP problems is NP-hard. However, ILP approaches are well studied resulting in efficient branch-and-bound algorithms for finding an optimal solution. Similar to the other natural language processing applications, ILP has been popularly employed for automatic summarization (Nishikawa et al., 2010; Galanis et al., 2012; Marciniak and Strube, 2005; McDonald, 2007; Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012; Li et al., 2013; Hirao et al., 2013).

Parveen and Strube (2015) propose an automatic summarizer using ILP. The objective of the optimization in their ILP formulation is to select a subset of sentences from the input document so that the selected sentences produce a coherent summary. The summary should ideally be optimal regarding importance, non-redundancy, and coherence. This model is of our interest because it sets up the optimization problem on the entity graph representations of texts. Given an input text, the importance and non-redundancy of selected sentences for a summary are measured using the bipartite entity graph of an input text. They employ entities as units of information which relate sentences. Entities are also used in other summarization techniques to measure the importance and the diversity of the information presented in sentences. Sentences that contain prominent entities of a text are essential to be extracted. A summary whose sentences refer to few entities may contain redundant information. Parveen and Strube (2015) use the outdegree of each node in the projection graph representation of an input text in order to measure how important the sentence associated with the node is for the coherence of the summary. In Chapter 4, we employ this model to evaluate our coherence patterns by replacing their out degree feature with the frequency of our coherence patterns.

4 Graph-based Coherence Patterns

In this chapter, we motivate and devise a method for extracting coherence patterns. We first provide motivations for representing texts by graphs and for the need of graph-based patterns for coherence modeling (Section 4.1). We then present an algorithm for coherence pattern extraction based on subgraph mining algorithms in graph theory (Section 4.2). We finally show how extracted coherence patterns can be used to represent the coherence of a text (Section 4.3) and evaluate coherence patterns and the coherence model on assessing the readability of texts and on generating coherent summaries (Section 4.4). We conclude with a summary of main contributions presented in this chapter (Section 4.5).

4.1 Why Graph-based Patterns?

In Chapter 2, we introduced the formal definition of the coherence modeling problem as it is investigated in the research presented in this dissertation. We used the "set" notation, which is a collection of unordered elements, from mathematics to provide a general formulation. We first defined the relation set as a set whose members indicate which sentences in a text are connected to each other. Then the concept of coherence pattern is explained as a subset of the relation set (see Definition 5 in Chapter 2). Our definitions do not make any assumption about the structure of connections among sentences, in order to give coherence models some flexibility to define or learn such structures.

In Chapter 3, we discuss the entity grid (Barzilay and Lapata, 2005, 2008) and the entity graph (Guinaudeau and Strube, 2013) coherence models. The entity grid model represents the relation set in our definition via a matrix. This model makes the definition of coherence patterns more specific by considering sequences, instead of sets, of grammatical transitions. It predefines coherence patterns by all possible grammatical transitions of entities across two adjacent sentences. The entity graph model employs graphs to represent the relation set of a text. It does not extract any pattern, but it makes a strong assumption about the set of all relations among sentences in a text: The bigger the relation set among sentences is, the more coherent the text is.

However, the entity graph model, without using any pattern, outperforms the entity grid model in experiments performed by Guinaudeau and Strube (2013). They argue that the entity graph model achieves higher performance in comparison to the entity grid model because the graph representation captures the distribution of entities across adjacent and non-adjacent sentences. Graphs are preferred over grids for coherence modeling for two reasons:

- They can model long-distance connections between sentences,
- They do not encounter the sparsity problem in text representations.

The graph representation of the distribution of entities in a text is transformed into a one-mode projection graph among sentence nodes. The average outdegree of nodes in a projection graph measures the extent to which sentence nodes in the graph are connected to each other. Guinaudeau and Strube (2013) assume that the average outdegree metric of a projection graph quantifies the local coherence of its corresponding text. Some research papers challenge this assumption (see Chapter 3) by employing different graph-based metrics with the goal to capture more information about the connectivity of graphs. The average outdegree metric is insufficient to measure the connectivity style of relations among nodes in the graph. Therefore, it is not a good predictor of the perceived coherence of a text. The results of the experiments in this chapter support this claim as well.

This weakness of the entity graph model motivates us to introduce some graph-based features that capture the connectivity style of nodes, how nodes are connected, in projection graphs. Considering the linguistics of coherence (see Chapter 2) we hypothesize that coherent texts reveal similar connectivity patterns in their graph representations which make them distinguishable from graph representations of incoherent texts. The results of the experiments in this chapter support this hypothesis (Section 4.4).

In general, the term “pattern” refers to some elements which are repeated or which are potentially repeatable. From the machine learning perspective, “pattern” refers to regularities in data (Bishop, 2006). In graph theory, patterns are subgraphs that occur or can potentially occur in graphs (Newman, 2010). Graph-based patterns can be extracted using subgraph mining methods. These subgraphs are called motifs in graph theory. Inspired by linguistic work by Daneš (1974), in the research of this dissertation we refer to the subgraphs extracted from graph representations of texts as connectivity patterns for graphs and coherence patterns for texts.

4.2 Coherence Pattern Extraction

In Chapter 2, different linguistic theories to coherence patterns are discussed. The primary intuition of coherence patterns in the research presented in this dissertation is inspired by the theory proposed by Daneš (1974). What Daneš (1974) proposes is that the generalized structure of coherent texts may be described in terms of underlying patterns of transitions between presented information in texts. Following entity-based approaches, we take entities mentioned in a text as the pieces of information that make sentences connected. The entity graph representation, introduced in Chapter 3, is employed to model the distribution of entities across sentences of each text in a corpus. Then projection graphs of texts are obtained from their entity graph representations. Projection graphs model the structure of sentence relations, which are obtained based on coreferent mentions, in texts. We employ a subgraph mining algorithm to automatically extract all connectivity patterns occurring in projection graphs of texts in a corpus as coherence patterns. We show that our patterns are similar to the patterns introduced by Daneš (1974).

4.2.1 Background about Graphs

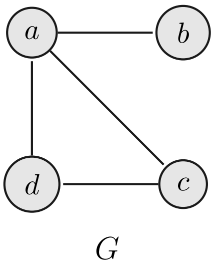
The main goal of the research presented in this dissertation is to represent texts with graphs and then use connectivity measures of graphs to quantify coherence. In order to explain our graph-based method, we need to define some necessary concepts from graph theory. We follow Newman (2010) to define these terms. We refer to them in the rest of the content of this dissertation.

Graph. A *graph* consists of a set of vertices, which are referred to as nodes, and a set of links, which are called edges. Following is a formal definition of a graph.

Definition 11. *A graph is a pair of two finite sets $G = (V, E)$ where V is a set of nodes and E is a set of edges whose elements are pairs of nodes.*

Figure 4.1 shows a graph with four nodes and four edges.

Directed graphs. If nodes in a graph are ordered or relations between nodes are not symmetric then pairs in the edge set of the graph should be interpreted as directed edges. A graph whose edges are directed is called a directed graph. Figure 4.2 shows the directed version of graph G depicted in Figure 4.1, where edge pairs in set E represent directed edges.



$$V = \{a, b, c, d\}$$

$$E = \{(a, b), (a, c), (c, d), (a, d)\}$$

$$V : Nodes, E : Edges$$

Figure 4.1: $G = (V, E)$ is a graph with node set V and edge set E.

Definition 12. *In a directed graph, an edge $e = (x, y)$ indicates a directed edge from node x towards node y, which are called the source node and the target node of edge e, respectively.*

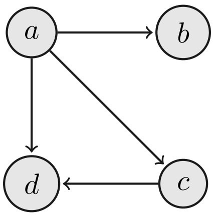


Figure 4.2: The directed representation of graph G in Figure 4.1.

Isomorphic. Two graphs may have an identical connectivity style but different appearances. Such graphs are called isomorphic graphs in graph theory. More formally, two graphs are isomorphic, if there is an isomorphism relation between the graphs.

Definition 13. *An isomorphism relation between graphs G_1 and G_2 is an association between node sets of these graphs:*

$$f : V(G_1) \rightarrow V(G_2), \tag{4.1}$$

such that any two nodes u and v of G_1 are adjacent if and only if $f(u)$ and $f(v)$ are adjacent in G_2.

In other words, two graphs G_1 and G_2 are isomorphic if they fulfill two conditions: (i) a one–to–one association exists between nodes of G_1 and nodes of G_2, (ii) two nodes of G_2 should be connected if and only if their associated nodes in G_1 are connected. Figure 4.3 illustrates two isomorphic graphs and an isomorphic relation between them.

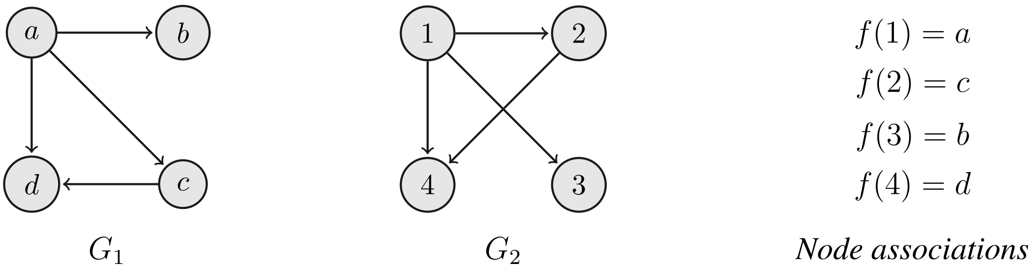


Figure 4.3: Two isomorphic graphs and a sample association between their nodes.

Subgraph. Graphs are a pair of two sets: a set of nodes and a set of edges. Since each of these sets has several subsets, a graph has several subgraphs as well.

Definition 14. *Graph G_2 is a subgraph of graph G_1 if G_2 is isomorphic to a graph whose nodes and edges are subsets of nodes and edges in G_1.*

For example consider graphs G_1 and G_2 in Figure 4.4. Graph G_2 is isomorphic with graph $G = (\{a, b, c\}, \{(a, b), (a, c)\})$ whose node and edge sets are subsets of the node and edge sets of G_1, respectively.

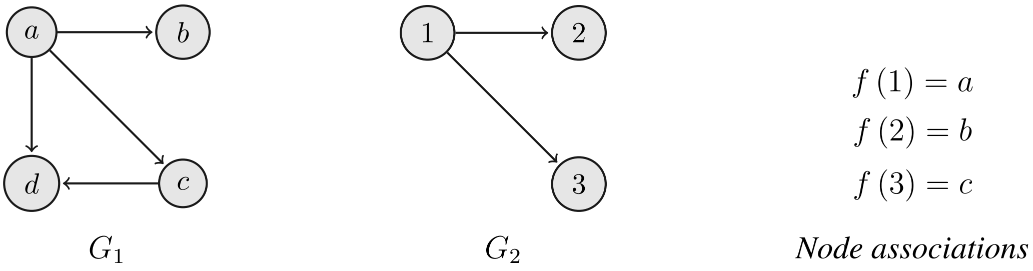


Figure 4.4: Graph G_2 is a subgraph of graph G_1.

K-node (sub)graph. The size of a (sub)graph is equal to the size of its node set. In Figure 4.4, graph G_2 is a 3-node subgraph of graph G_1.

Definition 15. *Graph $G_2 = (V_2, E_2)$ is a k-node subgraph of graph $G_1 = (V_1, E_1)$ if G_2 is a subgraph of G_1, and V_2 has k elements, $|V_2| = k$.*

Induced subgraph. An induced subgraph of a graph is a subgraph of the graph with an extra condition on its edges. Edges must connect any two subgraph nodes whose associated nodes in the main graph are connected.

Definition 16. *Graph $G_2 = (V_2, E_2)$ is an induced subgraph of graph $G_1 = (V_1, E_1)$ if $V_2 \subseteq V_1$ and $E_2 = \{(x, y) | x \in V_2, y \in V_2, (f(x), f(y)) \in E_1\}$.*

Figure 4.5 shows graph G_1 and two of its subgraphs G_2 and G_3. Graph G_2 is not an induced subgraph of graph G_1 because there is no edge between node 1 and node 3 in G_2 while their associated nodes, a and b, in graph G_1 are connected. In contrast, graph G_3 is an induced subgraph of G_1 because it contains all possible edges that exist in graph G_1.

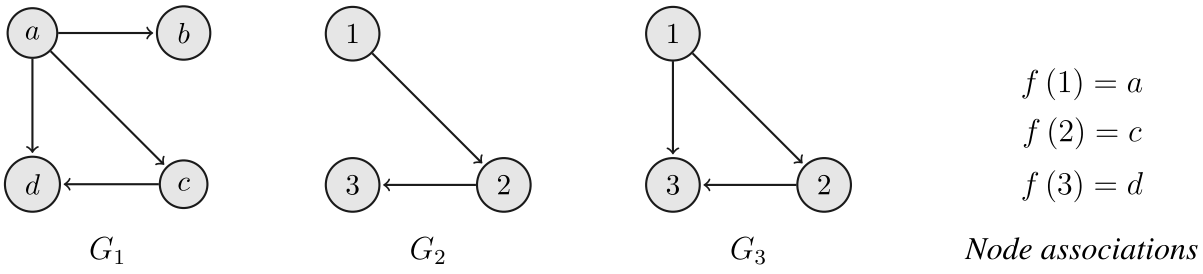


Figure 4.5: Both graph G_2 and G_3 are subgraphs of graph G_1. In contrast to G_2, graph G_3 is an induced subgraph of G_1.

It is worth mentioning that in this dissertation we mean induced subgraphs when we use the term "subgraph".

Graph signature. Given a list of graphs $\zeta = [G_1, G_2, \cdots, G_m]$, which are called basic graphs, a graph signature of graph G with respect to ζ is a vector of normalized frequencies of graphs in ζ in graph G:

$$\phi(G) = (f_1, f_2, f_3, \cdots, f_m), \tag{4.2}$$

where $\phi(G)$ denotes the graph signature and f_i is the frequency of graph G_i in graph G. The frequency of graph G_i in graph G is computed as follows:

$$f_i = \frac{count(G_i, G)}{\sum_{G_j \in \zeta} count(G_j, G)} \tag{4.3}$$

where $count(G_i, G)$ is a function which counts the number of occurrences of G_i in graph G. The reason of using normalized frequency instead of raw count is that normalized frequency cannot become biased to the number of nodes and edges in graph G. Normalized frequency of a subgraph can also be interpreted as the probability of the subgraph given graph G.

4.2.2 Coherence Pattern Mining

The entity graph representation of a text encodes the distribution of entities across sentences in a text. One-mode projection graphs model the connectivity between sentence nodes considering the entities shared by sentences. The main contribution of the research presented in this chapter is to introduce a set of graph-based patterns that encode the structure of connections (i.e. the connectivity style) in projection graphs. We use the frequencies of different subgraphs occurring in projection graphs to encode the connectivity style of projection graphs and ideally the coherence of texts.

Given a corpus of texts, we model connections among sentences in each text by its simple projection graph representation, i.e., P_U. Two sentence nodes are connected in such a projection graph if they share at least one entity node (see Chapter 3 for more details). The output is a set of graphs, each of which represents connections among sentences in a text in the corpus. We refer to this set as the *graph set*.

The connectivity of each graph in a graph set can be represented with its graph signature. The graph signature encodes the connectivity style of a graph into a vector. However, for representing the connectivity style of a graph with a graph signature, a list of basic graphs are required (see Section 4.2.1). We apply a subgraph mining algorithm to projection graphs in the graph set in order to obtain all basic graphs for computing graph signatures. These basic graphs, which are subgraphs of projection graphs, can be taken as patterns, each of which may have several occurrences in each projection graph. That is the reason that we refer to these basic graphs and their frequencies as coherence patterns and features, respectively. Figure 4.6 illustrates our approach for extracting coherence patterns.

The gSpan method. Coherence patterns are subgraphs that occur at least once in one of the projection graphs of texts in a corpus. Mining all subgraphs that occur in graphs of a graph set is computationally expensive and is proved to be an NP-complete problem (Althaus et al., 2004). Intuitively, a graph with $|E|$ edges, potentially has $\mathcal{O}\left(2^{|E|}\right)$ subgraphs. A graph with $|V|$ nodes at most has $\frac{(|V|-1)(|V|-2)}{2}$ edges which is in order of $\mathcal{O}\left(|V|^2\right)$. So the number of subgraphs in a graph is exponential to the squared number of nodes in the graph.

The goal of the research presented in this thesis is not to develop an algorithm for mining subgraphs. This problem has been extensively studied in computer science, and different algorithms and packages have also been developed for it. The gSpan algorithm (Yan and Han, 2002) is one of the efficient methods for mining subgraphs from graphs in a graph set. Here, we briefly describe its idea and method. Interested readers may find the exact algorithm of

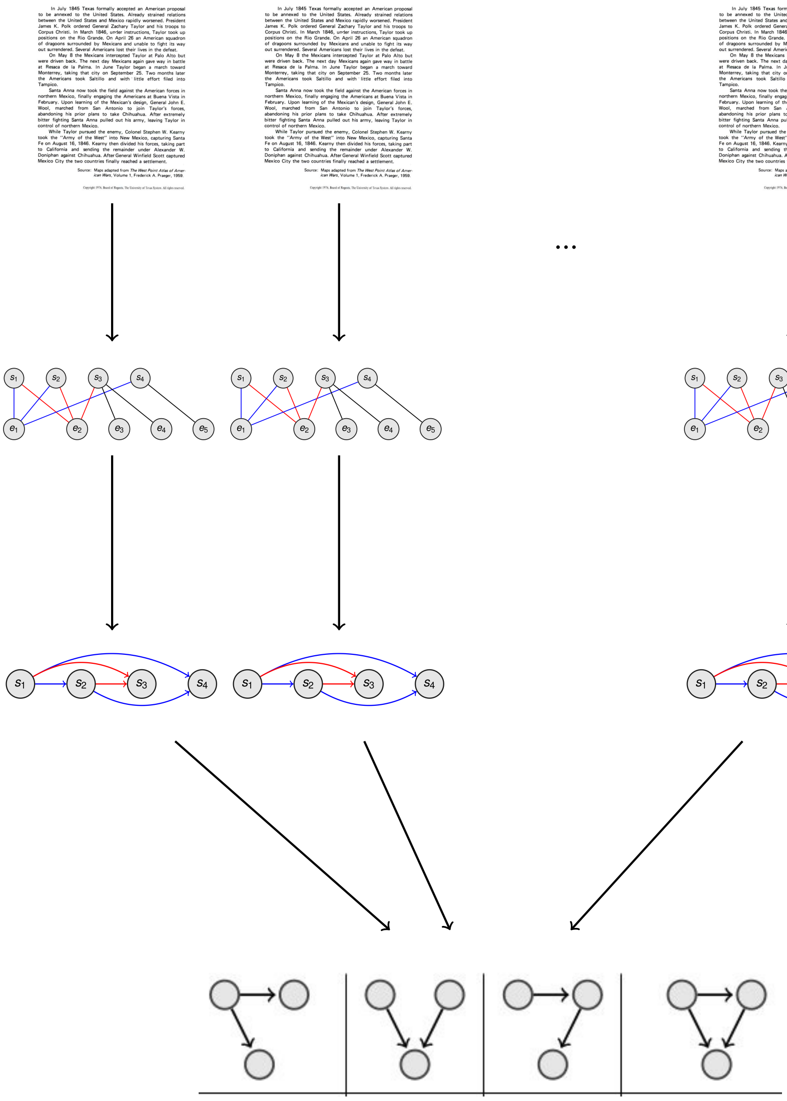


Figure 4.6: An illustration of the approach employed for extracting coherence patterns.

gSpan in Yan and Han (2002). The gSpan, which stands for graph-based Substructure pattern mining, algorithm is an approach for extracting all connectivity patterns (i.e. subgraphs) that occur in graphs of a graph set. It discovers all connected subgraphs without generating the candidates, so it is efficient in terms of computation time and memory usage. The gSpan method orders graphs in a graph set with respect to their structures. It then adapts a Depth-First-Search (DFS) strategy to extract connected subgraphs efficiently. It begins with subgraphs with only two nodes and expands them to larger subgraphs. We use gSpan to extract patterns from projection graphs. These patterns are basic graphs for graph signatures in our model.

4.3 Coherence Modeling

In this section, we explain how patterns, which are extracted by gSpan, can be used to measure the coherence of a text. Given a corpus of texts, we employ the entity graph to represent the entity distribution across sentences of each text in the corpus. We apply a one-mode projection on each entity graph to construct a projection graph (i.e. P_U) over the sentence nodes of the entity graph. The projection graph models the overall entity-based relationships among sentences.

The output of the subgraph mining method consists of subgraphs in different sizes. Involving subgraphs with different sizes for computing graph signatures results in subgraphs whose frequencies might be dependent on one another. Moreover, small subgraphs are likely to occur inside large subgraphs. This type of features is advised to be supplied separately to machine learning methods (Aggarwal, 2018). So, we extract all possible k-node subgraphs (i.e. coherence patterns) which occur in projection graph representations of texts in the corpus. The parameter k controls the size of subgraphs. Controlling the size of subgraphs helps to run the subgraph mining algorithm more efficiently in terms of the computational time. Besides, it controls the number of possible subgraphs that can be extracted: Large values of k yield many possible graphs with the size of k.

Assume that m coherence patterns, i.e. k-node subgraphs, are mined from all graph representations of texts in a corpus, the coherence of a text is encoded by a vector of frequencies of these patterns in the graph representation of the text. More formally, the coherence of text d is represented by a vector as follows:

$$Coh(T) \approx < f_0, f_1, f_2, ..., f_m > \tag{4.4}$$

where f_i represents the frequency of the ith pattern in the graph representation of text T.

This vector representation of coherence is identical with the graph signature concept in graph theory (see Section 4.2.1). As these vectors can be used to model the similarities and dissimilarities in connectivity structures of graphs, they can also model similarities and dissimilarities of connectivity structures of sentences in a text, which matches our definition of coherence vector in Chapter 2. From the machine learning viewpoint, these vectors can be viewed as feature vector representations of coherence. Each element of a vector is a feature which represents one aspect of the connectivity structure of sentences in a text. Consequently, these feature vectors can be supplied to a machine learning model in order to rank texts with respect to their coherence. A machine learning model during training learns to map a feature vector to a score, which can be interpreted as a coherence score, such that the score of a more coherent text is supposed to be higher than the score of a less coherent one. More formally, given two texts d and d' if text d is more coherent than text d' then a machine learning model learns parametric function β such that

$$\beta(d_v) > \beta(d'_v), \tag{4.5}$$

where d_v and d'_v are the feature vectors representing the coherence property of these texts.

4.4 Experiments

In this section, we perform a set of experiments to assess coherence patterns and features discussed in this chapter. We first investigate how patterns and their frequencies behave on encoding coherence in comparison to average outdegree. We then examine how the size of patterns affects the predictive power of our features for distinguishing graph representations of coherent texts against incoherent ones.

To this end, we limit the size of patterns by fixing parameter k in the model, which equals to the number of nodes in subgraphs (see Section 4.3), resulting in two sets of patterns. One set consists of subgraphs with three nodes, 3-node patterns, and the other set contains subgraphs with four nodes, which are referred to as 4-node patterns. There are no criteria on the number of edges in subgraphs.

We begin with extracting 3-node subgraphs in order to examine the soundness of mined coherence patterns. 3-node subgraphs are the smallest meaningful patterns that can model the connectivity style of sentence nodes in projection graphs. There is only one 2-node subgraph, $G = (V = \{a, b\}, E = \{(a, b)\})$, whose frequency in any projection graph is equal to the number of edges in the graph. Its interpretation is identical with the interpretation of the average outdegree applied by the entity graph model. 3-node subgraphs are too small and therefore are very likely to occur in most projection graphs in a graph set. Moreover, in order to analyze impacts of the size of patterns, we extract 4-node subgraphs to have more informative representations of the connectivity style of graphs.

As we discussed in Chapter 2, coherence is better to be evaluated in an extrinsic fashion because annotating texts for coherence is quite expensive and controlling all conditions for annotators is very difficult (Karamanis et al., 2004). In the research presented in this thesis, we focus on two extrinsic evaluation tasks: readability assessment and automatic text summarization. The former one is used to evaluate the predictive power of different proposed coherence features by computing the correlation between the values of the coherence features and the readability ratings assigned by human judges. We also check how well coherence features can rank texts with respect to their readability levels. The summarization task is a specific instance of text generation, where a subset of informative sentences in a text should be extracted and presented in a proper order to generate a coherent and readable summary. In this experiment, we evaluate the capabilities of our coherence patterns in producing coherent summaries. We show that by integrating our coherence patterns into a graph-based text summarizer (Parveen and Strube, 2015), the performance of the summarizer improves in terms of ROUGE metrics and human qualitative evaluations.

4.4.1 Readability Assessment

Readability assessment is about how well a text is understandable for its readers. Possible applications of readability assessment are automatic text summarization and simplification systems. Measuring readability can also be used in question answering and knowledge extraction systems to prune texts with low readability (Kate et al., 2010).

Readability assessment is a challenging task since various factors influence the processing time of a text for its readers. Accordingly, different features related to syntactic and semantic properties of texts have been used to assess readability. These features include shallow features (Flesch, 1948; Kincaid et al., 1975), language modeling features (Si and Callan, 2001; Collins-Thompson and Callan, 2004), syntactic features (Schwarm and Ostendorf, 2005) and the text flow or coherence (Barzilay and Lapata, 2008; Pitler and Nenkova, 2008). In the research of this dissertation, the readability assessment task mainly is utilized to evaluate the impact of features that represent coherence in quantifying the difficulty of texts. In a coherent text, each sentence has some connections with other sentences. Although these local connections somewhat make texts easy-to-read, none of the entity transition features introduced in the entity grid model (Barzilay and Lapata, 2008) significantly correlate with readability ratings assigned by human judges (Pitler and Nenkova, 2008). It is shown that the entity graph representations of the entity distribution across sentences provide more informative representation than the entity grid model (see Chapter 3). Here, we first investigate if the average outdegree metric proposed by the entity graph model strongly correlates with readability ratings. We also use this experiment to evaluate and interpret our coherence patterns.

It is worth to emphasize that the readability of a text is, of course, beyond the coherence of the text. That is why we do not use our features to predict the exact readability ratings associated with texts. However, since coherence is one of the crucial factors for readability assessment, easy-to-read texts appear to be coherent as well. We expect that the values of our features show a considerable correlation with human-provided readability scores associated with texts. In another experiment, we also check how well we can rank texts with respect to their readability if only coherence has been taken into account.

4.4.1.1 Data

We utilize the dataset collected by Pitler and Nenkova (2008), which consists of thirty articles randomly selected from the Wall Street Journal (WSJ) corpus. These articles are intended for an educated adult audience implying that they are well-written and free of grammatical errors. The articles were rated by three human judges on a scale from 1 to 5, where a higher rating indicates an easier-to-read article. Each of the judges had unlimited time to read the articles and assign the ratings. Human judges received the following questions (Pitler and Nenkova, 2008):

- How well-written is the article?
- How well does the article fit together?
- How easy is it to understand?
- How interesting is the article?

Pitler and Nenkova (2008) state that since in most cases judges gave the same rating to all questions, they only consider the given rates for the first question ("How well-written is the article?"). Then the average of ratings for this question is defined as the final rating of a text. That is the reason that they use these scores for coherence evaluation (Pitler and Nenkova, 2008). Moreover, as articles in this dataset are written for the Wall Street Journal and are aimed at the adult audience, their quality relates to the discourse features such as coherence rather than surface features, such as syntactic issues. The text presented in Example (8) is one[1] of the articles in this dataset with the final human readability score of 3.7 out of five.

(8) The Associated Press's earthquake coverage drew attention to a phenomenon that deserves some thought by public officials and other policy makers. Private relief agencies, such as the Salvation Army and Red Cross, mobilized almost instantly to help people, while the Washington bureaucracy "took hours getting into gear." One news show we saw yesterday even displayed 25 federal officials meeting around a table. We recall that the mayor of Charleston

[1] The ID of the article is WSJ-1818.

> complained bitterly about the federal bureaucracy's response to Hurricane Hugo. The sense grows that modern public bureaucracies simply don't perform their assigned functions well.

We exclude one article, which is a poem, from the dataset to make the texts consistent in style. Two articles are not available in Penn Treebank II, released by LDC. The final list of articles that are used from this dataset in the experiments presented in this dissertation is as follows: *WSJ_0068, WSJ_0177, WSJ_0232, WSJ_0311, WSJ_0402, WSJ_0494, WSJ_0613, WSJ_0663, WSJ_0717, WSJ_0744, WSJ_1011, WSJ_1027, WSJ_1043, WSJ_1281, WSJ_1324, WSJ_1472, WSJ_1520, WSJ_1724, WSJ_1746, WSJ_1773, WSJ_1784, WSJ_1818, WSJ_1906, WSJ_2121, WSJ_2238, WSJ_2336,* and *WSJ_2339.*

4.4.1.2 Feature Analysis

The goal of this experiment is to investigate the correlation of different features for representing coherence with readability ratings associated with the texts. To this end, we represent each text in the dataset by its entity graph, where entities are obtained heuristically by string matching among all nouns in an article. We apply a one-mode projection to obtain projection graphs, more specifically P_U. This projection graph captures the entity-based connectivity between sentences of articles. Given projection graphs of every article in the dataset, we compute the Pearson correlation between the values of a feature and readability ratings associated with articles. We use the Pearson correlation because feature values and ratings are meaningful by themselves. Furthermore, Pitler and Nenkova (2008) employ the Pearson correlation coefficient to evaluate entity transition features extracted from the entity grid representation (Barzilay and Lapata, 2008) of texts for the readability assessment task. Additionally, we use the Spearman correlation, as another correlation assessment method, to check how well rankings of texts based on the examined coherence features correlate with rankings based on human readbility ratings.

Pearson correlation and Spearman correlation. The Pearson correlation coefficient is a measure of the linear correlation between the values of two variables. In our experiments, variables are a coherence feature and readability ratings, which are assigned to texts by human judges. The Pearson correlation coefficient ranges between -1 and $+1$. Its high absolute value shows a strong correlation between the input variables. The sign of the Pearson correlation coefficient indicates the direction of the relationship between examined variables. In extreme cases, $+1$ shows a total positive linear correlation, 0 is no linear correlation, and -1 is a total negative linear correlation.

The Spearman correlation coefficient measures the ranking correlation between two variables. In our study, this metric assesses how well the relationship between the values of a coherence feature and the readability ratings can be described by a monotonic function. While Pearson's correlation assesses "linear" relationships, Spearman's correlation assesses "monotonic relationships" (whether linear or not). Intuitively, a perfect Spearman correlation coefficent of $+1$ or -1 occurs where there is a perfect monotonic relationship between the values of a coherence feature and readability ratings. The Spearman correlation coefficient equals $+1$ where the rankings are identical. It is high where the values of a feature and readability ratings have a similar rankings. It is low where they have dissimilar rankings, and is -1 where the rankings are fully opposed.

We use the notation ρ to refer to the correlation coefficient of Pearson and Spearman correlations. These correlations also measure how statistically significant examined variables are correlated. We refer to this measure as p_value, and we consider correlations with p_value < 0.05 statistically significant.

Experimental settings. In order to be compatible with the entity grid features that are evaluated by Pitler and Nenkova (2008), we use the gold parse trees in the Penn Treebank II (Marcus et al., 1994) to extract all nouns in an article as mentions. All nouns with identical stems are taken to be coreferent to the same entity. The Stemmer class from Stanford CoreNLP[2] is employed in this regard.

The subgraph mining and primary subgraph counting parts are performed by the Java implementation of the gSpan algorithm which is also publicly available[3]. This package counts all subgraphs, but it does not take care whether a subgraph is induced or not. Since we are interested in only induced subgraphs we employ SageMath[4] for counting induced subgraphs in each graph. For computing the Pearson and Spearman correlation coefficents and p_value, we use the implementation of these correlations provided by the Scipy[5] package for Python[6].

Mined coherence patterns. Figure 4.7 shows 3-node patterns, which are extracted from projection graph representations of texts in the examined readability dataset. There is an order

[2] V3.2.0, 2013-06-19

[3] `http://www.cs.ucsb.edu/~xyan/software/gSpan.htm`

[4] `http://sagemath.org/download-linux.html`

[5] Version: 1.1.0

[6] Version: 3.7.1

between nodes $s_t < s_u < s_v$, where "<" is the sign of preceding: Sentence s_t appears before both s_u and s_v in the input text, and sentence s_u precedes s_v.

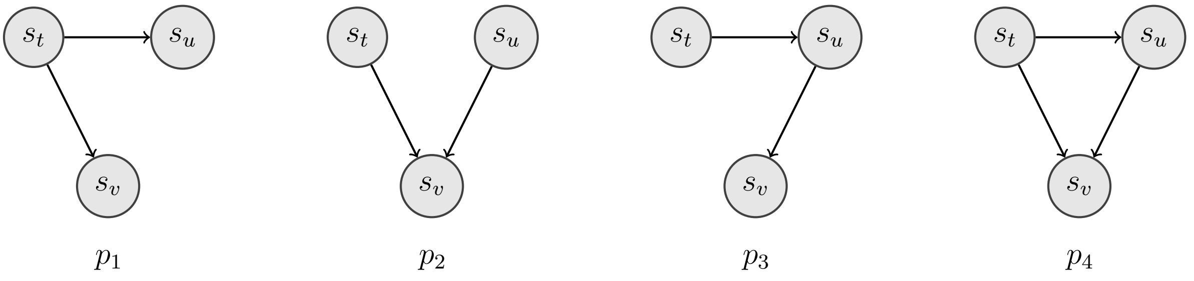


Figure 4.7: Extracted 3-node patterns from the readability dataset. Nodes are in order $s_t < s_u < s_v$ to show the order of sentences in a text.

We interpret these patterns as follows:

- $\boldsymbol{p_1}$**:** A sentence is connected with two of its subsequent sentences, while those sentences are not connected to each other. More precisely, at least two entities are mentioned in one sentence, i.e. $\boldsymbol{s_t}$, and the subsequent ones, i.e. $\boldsymbol{s_u}$ and $\boldsymbol{s_v}$, are about those entities. This non-linear pattern is similar to pattern 4 proposed by Daneš (1974) depicted in Table 2.1.

- $\boldsymbol{p_2}$**:** The connection between two sentences is made by a subsequent sentence of those sentences. This patterns indicates that entities in $\boldsymbol{s_t}$ and $\boldsymbol{s_u}$ are connected to each other in $\boldsymbol{s_v}$.

- $\boldsymbol{p_3}$**:** Each sentence tends to refer to an entity in its immediately preceding sentence. The absence of a connection between $\boldsymbol{s_t}$ and $\boldsymbol{s_v}$ indicates that the entity that connects $\boldsymbol{s_t}$ and $\boldsymbol{s_u}$ is different from the entity that connects $\boldsymbol{s_u}$ and $\boldsymbol{s_v}$. This pattern roughly reminds us of the center shift in centering theory. This pattern is also similar to the linear pattern proposed by Daneš (1974), i.e. pattern 1 presented in Table 2.1.

- $\boldsymbol{p_4}$**:** This pattern encodes three sentences that all are connected with each other. Entities that connect sentences are not necessarily unique. An essential property of this pattern is that it has the maximum number of edges showing many repetitions of entities among sentences. This pattern is roughly similar to two coherence patterns that are defined by Daneš (1974), i.e. pattern 2 and pattern 3 illustrated in Table 2.1.

Figure 4.8 shows all subgraphs with four nodes that are extracted from the projection graphs corresponding to texts in the readability dataset. These 4-node patterns have more capacity than 3-node patterns for capturing the connectivity structure of projection graphs because 4-node patterns contain more nodes and therefore more possible edges. So they are expected to distinguish texts better than 3-node patterns.

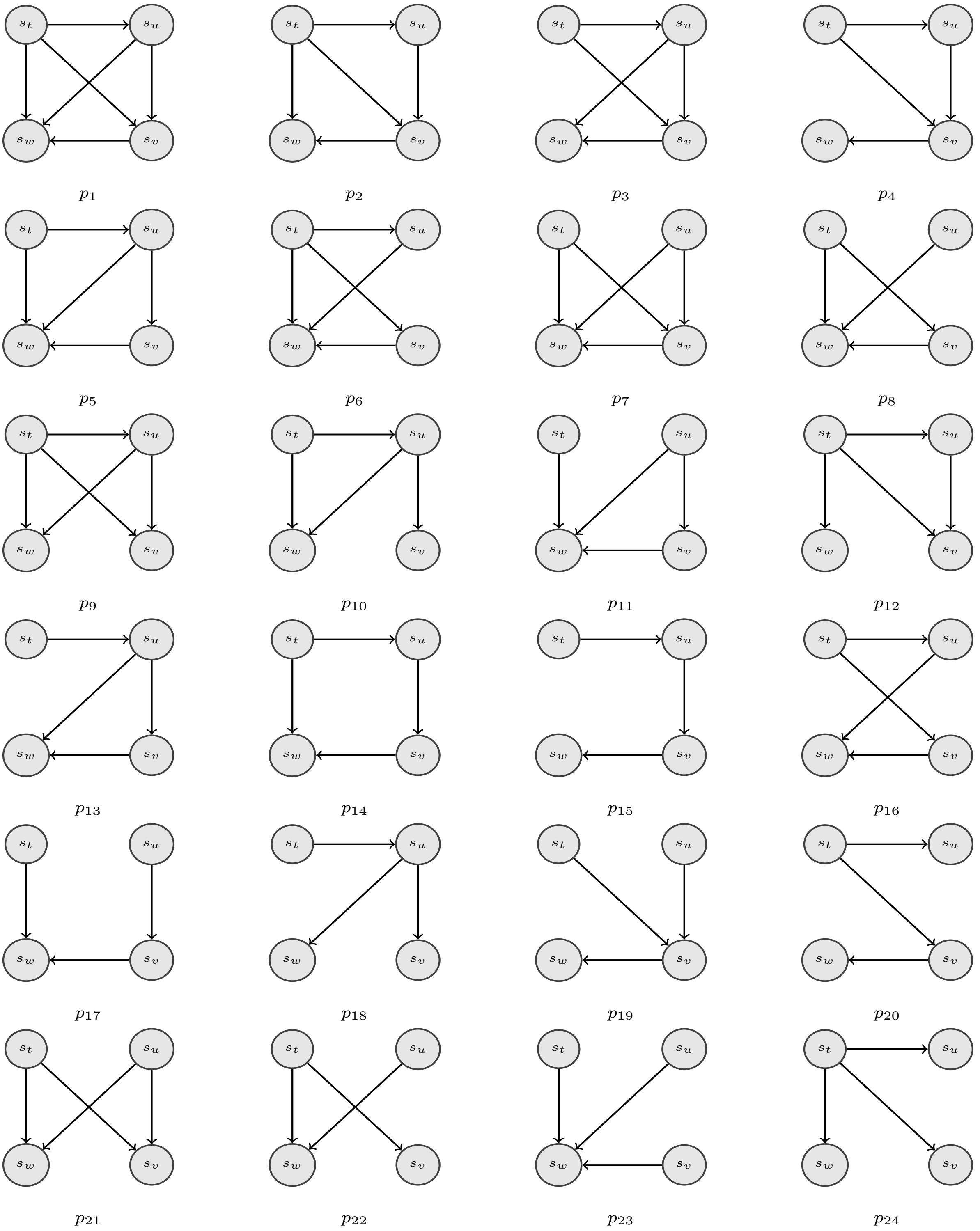
Compared features. We evaluate the following features: average outdegree, frequencies of 3-node patterns, and frequencies of 4-node patterns.

Results. Given the extracted coherence patterns as basic graphs, we compute the graph signature representation of each projection graph corresponding with each text in the dataset. The elements of the graph signature of a projection graph are frequencies of the extracted patterns in the projection graph. These elements together represent the connectivity style of the graph. From the texture perspective, the frequency of each subgraph encodes how frequently a coherence pattern occurs in a text.

We begin with the evaluation of the average outdegree feature as proposed by the entity graph model. Table 4.1 shows the results of computing the Pearson and Spearman correlation between the average outdegree of three projection graphs defined in the entity graph model (Guinaudeau and Strube, 2013). Each row in Table 4.1 presents the type of the projection graph that is used (see Chapter 3) to represent each text in the dataset. The columns with header ρ contain the correlation coefficients of the respective correlations, and columns with header p_value are what we use to identify significant correlations. The results show that the average outdegree feature of none of the projection graphs is significantly correlated with ratings assigned by humans with respect to both Pearson and Spearman correlations, confirming our argument at the beginning of this section: The average outdegree metric is insufficient to capture the connectivity style of a graph, and therefore the coherence of a text.

	Pearson		**Spearman**	
Projection type	ρ	**P_value**	ρ	**P_value**
P_U	-0.013	0.949	-0.016	0.938
P_W	0.151	0.452	0.133	0.508
P_{Acc}	0.150	0.455	0.159	0.427

Table 4.1: The Pearson and Spearman correlation coefficients and their p_value between the average outdegree feature of different projection graphs and readability ratings assigned by human judges.



p_1 p_2 p_3 p_4

p_5 p_6 p_7 p_8

p_9 p_{10} p_{11} p_{12}

p_{13} p_{14} p_{15} p_{16}

p_{17} p_{18} p_{19} p_{20}

p_{21} p_{22} p_{23} p_{24}

Figure 4.8: Extracted 4-node patterns, where order $s_t < s_u < s_v < s_w$ shows sentence order.

Table 4.2 shows the Pearson and Spearman correlation coefficients and their corresponding p_value between the frequencies of 3-node coherence patterns (see Figure 4.7) and readability ratings assigned by human judges. The results show that the frequencies of pattern p_1 and pattern p_4 are positively and the frequencies of pattern p_2 and pattern p_3 are negatively correlated with readability ratings. Among them, the frequency of pattern p_3 significantly correlates with human readability ratings with respect to both examined correlations. This pattern is similar to the shift in the center among sentences in a text. Since we are computing the frequency of this pattern in each text, this result shows that texts with many shifts in the center are perceived challenging to read.

	Pearson		**Spearman**	
3-node patterns	ρ	**P_value**	ρ	**P_value**
p_1	0.310	0.116	$\mathbf{0.396}$	$\mathbf{0.041}$
p_2	-0.325	0.098	-0.335	0.087
p_3	$\mathbf{-0.384}$	$\mathbf{0.048}$	$\mathbf{-0.419}$	$\mathbf{0.030}$
p_4	0.108	0.592	0.091	0.653

Table 4.2: The Pearson and Spearman correlation coefficients and their p_value between the frequency of 3-node patterns (see Figure 4.7) and readability ratings assigned by human judges.

We notice that the frequency of pattern p_1 is positively correlated with readability ratings. This positive correlation, which is singificant only with respect to the Spearman correlation, shows that this pattern occurs many times in easy to read texts. This pattern captures a sentence that introduces some information, which is limited in our model to entities, then subsequent sentences are about that information. These results are compatible with the structure of paragraphs in news articles. Good writers usually initiate topics, ideas or claims and then provide clear elaboration and reasons. Also, in English-speaking schools of essay writing and debating, there is the tendency to state the central claim of a text or a paragraph in the very first sentence followed by supporting arguments (Peldszus and Stede, 2015).

The frequency of pattern p_2 negatively correlates with human ratings, showing that this pattern has not been observed in coherent texts as frequently as in incoherent ones. This explains that it is difficult for readers to process sentences that become connected via their following sentences. Interestingly, the structural difference between pattern p_2 and pattern p_1 can be interpreted as the difference in the order of sentences. As shown in Figure 4.7, if node s_v preceded node s_t in pattern p_2 then this pattern would look like pattern p_1, which is a property of coherent texts. This means that these patterns can capture the order of information presented in sentences as well.

Finally, the frequency of one of our coherence patterns, unlike the average outdegree, is significantly correlated with human readability ratings with respect to both the Pearson and Spearman correlations. The correlation coefficients, ρ, have more distance to zero in comparison to those of the average outdegree feature. It indicates that the frequency of coherence patterns is more correlated with readability scores assigned by human judges compared to the average outdegree feature. Coherence patterns capture the connectivity style among sentences in a text as small units and beyond individual sentence connectivities. It is one of the essential differences between our coherence patterns and average outdegree.

Table 4.3 shows the Pearson and the Spearman correlation coefficients and their corresponding p_value between the frequencies of 4-node patterns and readability ratings assigned by human judges. Although patterns have the same number of nodes, they may contain a different number of edges. The second column presents the number of edges in each pattern.

Among all patterns that have positive correlation coefficients with readability ratings, p_{12} has the highest coefficient with respect to both examined correlations. This pattern has 4 edges. In contrast, among patterns with 4 edges the most negative correlation coefficient is for pattern p_{11}, with respect to both Pearson and Spearman correlations. A comparison of these two patterns shows that they have the same number of edges but different styles of connectivity. This confirms our intuitions: (i) The connectivity structure of projection graphs that represent coherent texts are similar to each other and different from those that represent incoherent texts; and (ii) The frequency of subgraphs can encode the connectivity structure of projection graphs and consequently coherence. Moreover, these two patterns, i.e. p_{11} and p_{12}, roughly remind us of the *ambiguity node* phenomenon introduced by Stoddard (1991) [p. 29]: *"[...] in some cases, there may be more than one logical, possible node for a given cohesive element in a text, in which case, a reader may see the resulting ambiguity but not be able to decide between the choices".* In pattern p_{11} a reader may need to make a decision about the center in s_w, whereas in p_{12} the center of s_w is the same as the center of s_t. This phenomenon can also be observed in all positively correlated patterns. It can be interpreted such that if readers have to return to one point in the text, they prefer to return to a sentence which is the core of the preceding sentences.

With reference to the results related to 3-node patterns, presented in Table 4.2, and the results of 4-node patterns, shown in Table 4.3, two observations are noticeable. First, the correlation coefficients of the 4-node patterns that are significantly correlated with readability

		Pearson		**Spearman**	
4-node patterns	**Number of edges**	ρ	**P_value**	ρ	**P_value**
p_1	6	0.103	0.609	−0.018	0.927
p_2	5	−0.212	0.288	−0.286	0.149
p_3	5	−0.176	0.380	**−0.392**	**0.043**
p_4	4	−0.257	0.196	**−0.463**	**0.015**
p_5	5	−0.140	0.486	−0.238	0.231
p_6	5	0.200	0.317	0.170	0.397
p_7	5	**−0.402**	**0.038**	−0.329	0.094
p_8	4	−0.317	0.107	−0.363	0.063
p_9	5	0.153	0.446	0.033	0.871
p_{10}	4	−0.238	0.232	−0.309	0.116
p_{11}	4	**−0.509**	**0.007**	**−0.509**	**0.007**
p_{12}	4	**0.449**	**0.019**	0.354	0.070
p_{13}	4	−0.045	0.824	−0.183	0.361
p_{14}	4	−0.033	0.870	−0.132	0.511
p_{15}	3	−0.358	0.067	**−0.450**	**0.019**
p_{16}	4	−0.068	0.736	−0.239	0.230
p_{17}	3	−0.308	0.118	**−0.440**	**0.022**
p_{18}	3	**−0.546**	**0.003**	**−0.439**	**0.022**
p_{19}	3	**−0.601**	**0.001**	**−0.439**	**0.022**
p_{20}	3	0.094	0.641	−0.103	0.610
p_{21}	4	0.068	0.736	0.000	0.998
p_{22}	3	−0.374	0.055	−0.311	0.114
p_{23}	3	−0.314	0.111	−0.298	0.130
p_{24}	3	0.100	0.620	−0.057	0.776

Table 4.3: The number of edges in 4-node patterns, and the Pearson correlation coefficient and their corresponding p_value between the frequency of patterns and human-provided readability ratings.

ratings are stronger, which means their absolute value is higher, than those of 3-node patterns. This confirms our intuition that large subgraphs capture more information about the connectivity style of projection graphs. So, they are more potent predictors of coherence than 3-node patterns. Second, pattern p_{12} from 4-node patterns is a combination of pattern p_1 and pattern p_4 from 3-node patterns with positive correlation coefficients, which could explain why pattern p_{12} demonstrates the strongest positive correlation coefficients. However, we should refrain from interpreting too much into these patterns.

Pitler and Nenkova (2008) show that none of the entity transition features proposed by the entity grid model strongly correlate with readability ratings of these texts. While these coherence features intend to capture entity-based coherence, they seem to be too weak to do so in isolation. In contrast to Pitler and Nenkova (2008), we are able to report a statistically significant correlations between some entity-based features and human readability ratings.

In the above experiments, we follow the experimental settings used by Pitler and Nenkova (2008) for reporting the significance test. However, one may, arguably, considers all k-node patterns in one statistical family and perform the Bonferroni correction to control the familywise error. In this case, the p_values reported in Table 4.2 should be less than 0.0125 and those reported in Table 4.3 should be less than 0.002 to demonstrate 95% confidence. Using the Bonferroni correction, none of the 3-node patterns shows a significant correlation but pattern p_{19} from 4-node patterns demonstrates a significant correlation with respect to the Pearson correlation. Overall, considering the ρ and p_value of all examined features, our patterns show stronger correlations with readability ratings assigned by human judges than the average out degree feature proposed by the entity graph model and also than the grammatical transition features proposed by the entity grid model. A side effect of statistical corrections is that they may increase the number of false negatives. In other words, the benefits of features might be undervalued because of corrections (Perneger, 1998; Nakagawa, 2004). Therefore, we further evaluate the benefits of our coherence patterns and features for ranking texts with respect to their readability (see Section 4.4.1.3).

We summarize our findings in this experiment as follows:

- The average outdegree metric proposed by Guinaudeau and Strube (2013) is not strongly correlated with human ratings assigned to texts in the examined dataset;
- 3-node patterns, mined from projection graphs, are roughly similar to patterns introduced by Daneš (1974);
- The connectivity style of a projection graph can be encoded by the frequency of extracted subgraphs or coherence patterns.

4.4.1.3 Readability Ranking

In this experiment, we use coherence models to rank texts with respect to their readability property. Given that easier to read texts are more coherent than difficult texts, a coherence model should ideally be able to rank texts with respect to their readability. We investigate how thoroughly rankings predicted by a coherence model match rankings that are based on the readability ratings assigned by humans. The readability ranking task may in principle be more natural than readability assessment because in most natural language processing applications the main concern is with the relative quality of texts rather than their absolute scores. Following Pitler and Nenkova (2008) we rank texts in a pairwise approach with respect to their readability ratings that are assigned by humans. We treat this task as a classification task: Given a pair of texts, which one is easier to read or more coherent?

Evaluation metric. The performance of a set of features that capture coherence for the readability ranking task is measured by the accuracy of rankings predicted by the classification model where it is supplied by the feature set. The accuracy is calculated as follows:

$$Accuracy = \frac{\textit{the number of correct rankings}}{\textit{the number of pairs}}. \tag{4.6}$$

We also compute the F-measure in order to compare our features with baseline features. The F-measure is the average of the F1-scores which are computed for each class. Note that our problem is ranking which is treated as classification. There is no true class to be classified. So, in different turns, we take each class as the true class and compute the F1-measure for that class. Then we report the average F1-scores as the F-measure for this task. The F1-score for each class is the harmonic mean of precision (P) and recall R:

$$\textit{F1-score} = 2\frac{P \cdot R}{P + R}. \tag{4.7}$$

Precision is computed as follows:

$$P = \frac{\textit{the number of correct decisions}}{\textit{the number of decisions}}, \tag{4.8}$$

and recall is calculated as follows:

$$R = \frac{\textit{the number of correct decisions}}{\textit{the number of pairs in that class}}. \tag{4.9}$$

Since the dataset is not accompanied by any standard split of training and test sets, we used

10-fold cross-validation for comparing the quality of different coherence feature sets. The reported numbers are the average of all numbers obtained from all runs of 10-fold cross-validation.

Experimental settings. The settings of this experiment are as it is performed by Pitler and Nenkova (2008). Text pairs include texts, from the dataset introduced in Section 4.4.1.1, whose readability ratings differ by at least 0.5. This criterion is supposed to ensure that the difference in readability ratings of texts in a text pair is noticeable enough that a coherence model distinguishes their differences. If the first text in a pair has a higher readability score, a label $+1$ is assigned to the pair; otherwise, a label -1 is assigned. In total, the number of pairs obtained by this procedure is 209 pairs in which 105 pairs have label $+1$, and 104 pairs have label -1. We employ WEKA's linear support vector implementation (SMO) to classify the pairs. The SMO model is supplied with different sets of features representing the coherence of each text. The first set of features is grammatical transitions of entities, with a sequence length of two, proposed by the entity grid model (Barzilay and Lapata, 2005, 2008). For creating entity grids we use Brown Coherence Toolkit v1.0, which is set up in a docker by the author of this thesis[7] to be used quickly for future research. The input parse trees to this toolkit are gold parse trees from Penn Treebank II (Marcus et al., 1994). Entities are obtained by performing a string-match over head nouns of noun phrases. We use Student's t-test, which can detect significant differences between paired samples, to test statistical significance of our improvments.

Results. We compare our graph-based coherence features, which are the frequencies of k-node subgraphs in projection graphs, with the proposed features obtained from grammatical transitions of entities by the entity grid model. In order to compare different sets of coherence features, we employ the same machine learning model, i.e. SVM, for training and testing on each feature set.

Table 4.4 summarizes the results of the readability ranking task on this dataset. Since the reported F-Measure and Accuracy have the same trend, we compare systems only based on Accuracy. Baseline features are the frequency of grammatical transitions of entities across sentences. This feature set is also evaluated[8] as coherence features on this dataset by Pitler and Nenkova (2008).

[7] The docker is available `https://github.com/MMesgar/text_to_entity_grid`

[8] The accuracy reported in their paper is 79.42%. Our reimplementation achieves higher accuracy because our dataset has three articles less. This also explains why the accuracy of the setting "None (Majority class)" is less than 50%.

Features	**Accuracy**	**F-measure**
None (Majority class)	47.85%	0.478
S&O features	71.77%	0.718
Baseline features	83.25%	0.833
3-node	79.43%	0.794
4-node	89.00%*	0.890*
3-node & 4-node	88.52%*	0.885*
Baseline features & 4-node	93.30%*	0.933*
S&O features & 4-node	95.70%*	0.957*

Table 4.4: The accuracy and F-measure of the SVM classifier with different sets of features which represent coherence. ∗ indicates statistically significant (p_value< .01) improvements with respect to the baseline features marked. S&O features are syntactic readability features presented by Schwarm and Ostendorf (2005) which lack coherence.

The accuracy of the classifier where it is supplied with 3-node subgraphs as coherence patterns is lower than where it is supplied with the entity grid's features. This might happen because of two reasons. The entity grid features represent grammatical role transitions of entities, which are more informative than the connections used in 3-node patterns. Connections in 3-node patterns only capture the existence of at least one shared entity between sentences. On the other hand, 3-node patterns are small and consequently more likely to occur in most projection graphs, so their frequencies cannot distinguish between coherent and incoherent texts effectively.

The feature set containing the frequency of 4-node patterns outperforms the baseline feature set by about 6 percentage points in terms of accuracy. This confirms our intuition that the entity-based connectivity structure among sentences, which is modeled by projection graphs, distinguishes coherent texts from incoherent texts. An advantage of 4-node patterns over the baseline features is that long-distance relations are also taken into account. However, our coherence patterns lack the grammatical information in the entity relations. Our other hypothesis was that since 4-node patterns are larger than 3-node patterns, they have more capacity in terms of nodes and edges, to model the connectivity style of projection graphs and therefore coherence. Comparing the accuracy of 4-node patterns against 3-node patterns confirms this hypothesis. 4-node patterns outperform 3-node patterns by 10 percentage point difference in accuracy. That is a substantial improvement.

We combine the frequency of 3-node and 4-node patterns into one feature set and supply it to the classifier. The obtained accuracy is superior to the one with only 3-node patterns but slightly worse (-1 percentage point) than the one with only 4-node patterns. An explanation for this is that 4-node patterns implicitly contain 3-node patterns in themselves. Combining these two sets of features does not provide useful information for modeling the connectivity style of projection graphs more than what exists in 4-node patterns.

The combination of baseline features, i.e. the frequency of grammatical transitions of entities over adjacent sentences, and 4-node subgraphs achieves the best accuracy. An interpretation for this is that although our coherence patterns capture the connectivity structure among sentences of a text, integrating linguistic information such as syntactic transitions of entities may improve the quality of a coherence model.

Finally, the combination of 4-node patterns with syntactic readability features (Schwarm and Ostendorf, 2005), which are presented by S&O in Table 4.4, demonstrates the highest performance for this task. This observation shows that our graph-based coherence features can effectively be combined with shallow features for readability to improve the quality of readability models.

4.4.2 Automatic Summarization

In this section, we evaluate our approach to coherence pattern mining in automatic text summarization, which is an instance of a text generation task. We employ the summarization system proposed by Parveen and Strube (2015) as it is developed on the entity graph model (see Chapter 3). This matters because the entity graph model is the framework of our coherence model as well. Parveen and Strube (2015) assume that the outdegree of a node in the projection graph of an input text measures how much its corresponding sentence contributes to the coherence of the summary. This assumption has three weaknesses:

- The summarizer becomes biased to extract sentences from the beginning of a text because these sentences potentially have high outdegrees. This is because edges in a projection graph are directed to capture the sentence order in a text. As the text progresses, the potential outdegrees of sentence nodes decrease. As an example, assume a text with n sentences, the outdegree of the first sentence in this text can be high as up to $n-1$, but the outdegree of the last sentence is zero. It is worth mentioning that in the entity graph coherence model, the outdegrees of nodes in a projection graph are averaged to measure the connectivity of the graph or the coherence of the entire text. Limiting this metric to each sentence node does not imply that the sentence is crucial for the coherence of the summary of the text.

- In the summarization model proposed by Parveen and Strube (2015) the outdegrees of sentence nodes are computed in the projection graph of the input text. The ideal aim of a summarization system is to extract sentences that make the summary coherent. So, the connectivity of sentences should be evaluated concerning only selected sentences for the summary, rather than all sentences in the input text.

- In the readability experiment in the previous section, we have shown that the average outdegree is not the best metric for encoding the connectivity style of projection graphs and therefore coherence.

We focus on the coherence aspect of the summarization system proposed by Parveen and Strube (2015) with two motivations: to use the automatic summarization task as another application for evaluating our graph-based coherence patterns; and to improve the performance of the examined summarization model by integrating our coherence patterns, rather than outdegree, into the summarization model. Our intuition is that human-generated summaries given in a dataset are expected to be coherent enough to be readable. So, if a summarization system extracts sentences of a text so that their connectivity style in the produced summary is similar to the connectivity style of sentences in human-generated summaries, then the produced summary is sufficiently well-connected and therefore coherent.

Summarization problem formulation. We use the notation H to denote a dataset which consists of summaries written by humans for a set of texts. We use a different dataset consisting of a set of document-summary pairs where the ith pair contains document d_i and its gold summary s_i written by human experts, where the dataset is represented by $D = \{(d_0, s_0), (d_1, s_1), ..., (d_{N-1}, s_{N-1})\}$. Parameter N is the number of document-summary pairs. We assume that each document has a title. In practice, in automatic summarization models when an input text does not have any title, the first sentence of the text is taken as the title. Gold summaries can be employed to train a summarization model, and can also be used to evaluate summaries produced by the model during the evaluation phase. We make a challenging assumption such that documents in H and D are disjoint (i.e. there is no intersection among summaries or documents of these two datasets) but come from the same genre.

4.4.2.1 Coherence Pattern Mining

In this section, we expertslain how we extract all coherence patterns from the human summaries that are collected in H. To this end, we represent all texts in this dataset by their entity graph representations. Then we apply the one-mode projection P_U to entity graphs for obtaining projection graphs. We refer to the set of projection graphs that represent texts in H by GS_H, which stands for the Graph Set of H. Afterward, we use the gSpan method to extract all possible subgraphs from graphs in GS_H. In Section 4.4.1, we have shown that frequencies of coherence patterns in a projection graph capture the connectivity style of the graph and correlate with readability ratings assigned by humans. Similarly, we take the subgraphs that frequently occur in GS_H as the connectivity styles that are desired by humans to connect sentences in summaries. In order to model this, we weight each coherence pattern based on its number of occurrences, i.e. count, in graphs in GS_H. The weight of a coherence pattern, $weight(p_u)$, is the sum of its counts in all graphs in GS_H divided by its maximum count:

$$weight(p_u) = \frac{\sum_{k=1}^{M} count(p_u, g_k)}{\max_{k=1}^{M} count(p_u, g_k)}, \tag{4.10}$$

where M is the number of graphs in GS_H, and g_k indicates the kth projection graph in GS_H. The nominator of the weight function is the sum of the number of occurrences of pattern p_u in graphs in GS_H. The denominator diminishes the weight of a coherence pattern if it occurs in a few graphs of GS_H. In an extreme case, if a pattern occurs only in one graph in the graph set then $max_{k=1}^{M} count(p_u, g_k)$ is equal to $\sum_{k=1}^{M} count(p_u, g_k)$, so the weight of the pattern becomes one. If a pattern occurs in many graphs in GS_H the denominator becomes smaller than the nominator; therefore the weight becomes greater than one. The weights of coherence patterns are not on the same scale. So we normalize the weights by

$$z = \frac{x - \mu}{\sigma}, \tag{4.11}$$

where μ and σ respectively are the mean and the standard deviation of all weights. Variable x is the weight of a pattern. Finally a sigmoid function

$$g(z) = \frac{1}{1 + exp(-z)}, \tag{4.12}$$

scales weights to a value between 0 and 1.

4.4.2.2 Summary Generation

In this section, we explain how our coherence patterns are integrated into the summarizer proposed by Parveen and Strube (2015). Assume that we want to produce a summary for document d from dataset D. Parveen and Strube (2015) develop an Integer Linear Programming (ILP) approach to extract the best possible subset of sentences with respect to importance, non-redundancy and coherence factors. They measure the contribution of each sentence in a document to the coherence of the output summary by the outdegree of the sentence node in the projection graph representation of the input text. To formulate the problem in ILP we represent all sentences in input document d by set $S = \{\hat{s}_0, \hat{s}_1, ..., \hat{s}_n\}$, where $\hat{s}_i$ is a boolean variable whose value represents if the ith sentence of document d is selected for the summary or not. Set $E = \{\hat{e}_1, \hat{e}_2, ..., \hat{e}_m\}$ is a set of boolean variables representing entities in a text. The value of variable $\hat{e}_i$ represents if its associated entity is mentioned in the selected sentences or not. Set $P = \{\hat{p}_1, \hat{p}_2, ..., \hat{p}_k\}$ is a set of boolean variables which are associated with coherence patterns. The True value of a variable in this set indicates that the pattern associated with the variable is a subgraph in the projection graph of the generated summary. We consider different weights, i.e. λ_I, λ_R, and λ_C, for the significant factors of a good summary, i.e., importance, non-redundancy, and pattern-based coherence. The objective function of ILP is as follows:

$$\max(\lambda_I f_I(S) + \lambda_R f_R(E) + \lambda_C f_C(P)), \tag{4.13}$$

where $f_I(S)$ is the function that measures the importance of the selected sentences, $f_R(E)$ measures the non-redundancy among the selected sentences with respect to the selected entities, and $f_C(P)$ measures the coherence of the selected sentences with respect to coherence patterns extracted from dataset H.

The importance function, $f_I(S)$, is calculated by considering the ranks of selected sentences for a summary:

$$f_I(S) = \sum_{i=1}^{n} Rank(s_i) \cdot \hat{s}_i, \tag{4.14}$$

where $Rank(s_i)$ represents the rank of sentence s_i compared to other sentences. Parameter n is the number of sentences in input document d.

The ranks of sentences are calculated by the Hyperlink-Induced Topic Search (HITS) algorithm. The HITS algorithm was developed by Kleinberg (1999) for ranking web pages considering the way they are connected. Kleinberg (1999) categorized web pages into two groups: Authorities, which are informative web pages; and Hubs, pages that link to informative web pages[9]. Here, authorities are sentences and hubs are entities. The entity graph representation of document d encodes the connections among sentences and entities in the document. Initial ranks for sentences are as follows:

$$Rank_{init}(s_i) = 1 + sim(s_i, title), \tag{4.15}$$

where $sim(s_i, title)$ is the cosine similarity between s_i and the title of document d. Initial ranks for all entities are set to 1s. After applying the HITS algorithm to the entity graph using the above initialization, the final ranks of sentences are taken as their importance.

The non-redundancy function in the objective function, $f_R(E)$, is measured as follows:

$$f_R(E) = \sum_{j=1}^{m} \hat{e}_j, \tag{4.16}$$

where m is the number of entities in the input document.

The summary contains non-redundant information if it includes only unique entities. The other interpretation of Equation 4.16 is that if a summary contains more entities, it is covering more details of the document.

The coherence function in the objective function, $f_C(P)$, measures the coherence of the summary that is obtained by concatenation of selected sentences in the order that they appear in the input document. This function uses coherence patterns and their weights, which are extracted from dataset H, as follows:

$$f_C(P) = \sum_{u=1}^{U} weight(p_u) \cdot \hat{p}_u, \tag{4.17}$$

where $\hat{p}_u$ is the binary variable associated with coherence pattern p_u, and $weight(p_u)$ is the weight of this pattern, which is basically the frequency of pattern p_u in graph set GS_H (see Equation 4.10), and U is the number of patterns extracted from GS_H. The value of binary variable $\hat{p}_u$ is one if pattern p_u is a subgraph of the projection graph representation of sentences selected from the input document. Computing the value of $\hat{p}_u$ is challenging because the list of selected sentences at different optimization states is not explicit, so building the entity and projection graphs only over the selected sentences at optimization states is impossible. However, since the projection graph of selected sentences at different states of ILP is a subgraph

[9] The idea behind Hubs and Authorities derived from an insight into the creation of web pages when the Internet was originally forming; that is, specific web pages, known as hubs, served as large directories that were not authoritative in the information that they held. However, they were used as compilations of a broad catalog of information that pointed users to other authoritative pages.

of the projection graph of the input document, we can define some constraints for our ILP to check if a coherence pattern occurs in a subgraph of the projection graph of the input document such that the subgraph consists of nodes associated with selected sentences or not. In the following, we explain the details of the constraints that are used in our ILP formulation.

Constraints. Here we define all constraints over variables of our model to complete our ILP formulation of the summarization task. The first constraint limits the length of the summary:

$$\sum_{i=1}^{n} l_i \cdot \hat{s}_i \leq l_{max} \tag{4.18}$$

where l_{max} is the maximal permitted length of the summary and l_i is the length of the sentence associated with binary variable $\hat{s}_i$. If l_{max} is defined based on the number of words in a summary then l_i is the number of words in the corresponding sentence. If l_{max} is defined based on the number of sentences in a summary then it is sufficient to take each sentence as one unit, i.e. $l_i = 1$.

The constraint in Equation 4.19 ensures that if the ith sentence of document d is selected, i.e. $\hat{s}_i = 1$, all entities that are mentioned in the sentence, shown by E_i, are also selected.

$$(\sum_{e_j \in E_i} \hat{e}_j) \geq (|E_i| \cdot \hat{s}_i) \text{ for } i = 1, ..., n, \tag{4.19}$$

where $|E_i|$ is the number of entities in the sentence.

Similarly if an entity is selected to be mentioned in the summary then at least one sentence which contains a mention of the entity is selected as well:

$$(\sum_{s_i \in S_j} \hat{s}_i) \geq \hat{e}_j \text{ for } j = 1, ..., m, \tag{4.20}$$

where S_j represents the set of binary variables of sentences whose nodes in the entity graph representation of the document are connected to the entity node associated with $\hat{e}_j$.

In order to define constraints for involving coherence patterns in the optimization process, we adapt the graph matching algorithm proposed by Lerouge et al. (2015). This algorithm uses ILP to check if a pattern is a subgraph of another graph. However, we need to introduce more criteria to check if a pattern occurs in a subgraph of a projection graph, where the subgraph consists of only selected nodes.

To model the graph matching problem between projection graph $g = (V_g, E_g)$ and pattern $p_u = (V_{p_u}, E_{p_u})$, two kinds of mapping binary variables are used:

- For each pair of nodes $i \in V_p$ and $k \in V_G$, there is a binary variable $\hat{x}_{i,k}$, such that $\hat{x}_{i,k} = 1$ if nodes i and k are matched together, 0 otherwise.
- For each pair of edges $(i, j) \in E_p$ and $(k, l) \in E_G$, there is a binary variable $\hat{y}_{ij,kl}$ such that $\hat{y}_{ij,kl} = 1$ if edges (i, j) and (k, l) are matched together, 0 otherwise.

Figure 4.9 illustrates these matching variables.

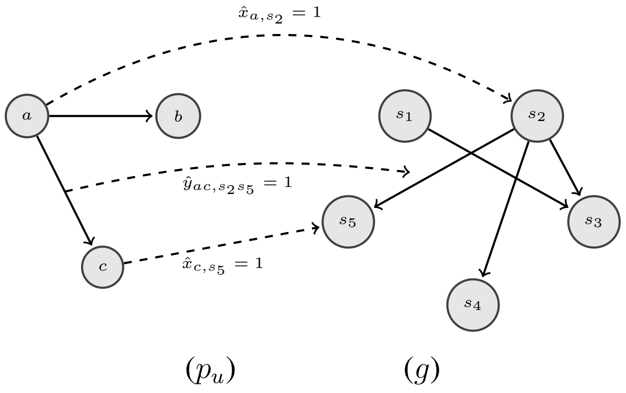


Figure 4.9: An illustration of matching variables for overlaying graph g with coherence pattern p_u.

Given the above variables, we need to define some constraints in order to check if pattern p_u is an induced subgraph of the selected nodes in projection graph g. To do so, we explain constraints which are used to check if a pattern is an induced subgraph of a projection graph or not.

- Every node of the pattern matches at most one unique node of the graph:

$$\sum_{k \in V_g} \hat{x}_{i,k} \leq 1 \quad \forall i \in V_{p_u}, \tag{4.21}$$

- Every edge of the pattern matches at most one unique edge of the graph:

$$\sum_{kl \in E_g} \hat{y}_{ij,kl} \leq 1 \quad \forall (i, j) \in E_{p_u}, \tag{4.22}$$

- Every node of the graph matches at most one node of the pattern:

$$\sum_{i \in V_{p_u}} \hat{x}_{i,k} \leq 1 \quad \forall k \in V_g, \tag{4.23}$$

- A node of pattern p_u matches a node of graph g if an edge originating from the node of pattern p_u matches an edge originating from the node of g:

$$\sum_{kl \in E_g} \hat{y}_{ij,kl} = \hat{x}_{i,k} \; \forall k \in V_g, \; \forall ij \in E_{p_u}, \tag{4.24}$$

- A node of pattern p_u matches a node of graph g if an edge targeting the node of pattern p_u matches an edge targeting the node of g:

$$\sum_{kl \in E_g} \hat{y}_{ij,kl} = \hat{x}_{j,l} \; \forall l \in V_g, \; \forall (i,j) \in E_{p_u}, \tag{4.25}$$

- Following constraint ensures that the model extracts induced patterns. Pattern p_u is an induced subgraph of graph g if p_u contains all possible edges that are present in g. So

$$\sum_{i \in V_{p_u}} \hat{x}_{i,k} + \sum_{j \in V_{p_u}} \hat{x}_{j,l} - \sum_{(i,j) \in E_{p_u}} \hat{y}_{ij,kl} \leq 1 \quad \forall (k,l) \in E_g. \tag{4.26}$$

The above constraints check whether the pattern is an induced subgraph of the projection graph. But we must also check if the pattern occurs in a subgraph of the projection graph such that the subgraph contains only selected sentence nodes for producing the summary. In simple words, all associated sentences to the pattern nodes must be selected for the summary. So we define some more constraints in this regard:

- If sentences s_k and s_l are selected for the summary then the edge between them must be selected $(\hat{z}_{kl} = 1)$ as well.

$$s_k \cdot \hat{s}_l = \hat{z}_{kl} \quad \forall k, l \in V_g \tag{4.27}$$

- Pattern p_u is present in the summary ($\hat{p}_u = 1$) if and only if one of its instances in the projection graph is included in the summary, i.e., some of the selected sentence nodes must be present in an instance of pattern p_u. Let $|V_{p_u}|$ be the number of nodes and $|E_{p_u}|$ be the number of edges in pattern p_u then this constraint can be formulated as follows:

$$\sum_{i \in V_{p_u}} \sum_{k \in V_g} \hat{s}_k \cdot \hat{x}_{i,k} + \sum_{ij \in e_{p_u}} \sum_{kl \in E_g} \hat{z}_{kl} \cdot \hat{y}_{ij,kl} = \hat{p}_u(|V_{p_u}| + |E_{p_u}|) \tag{4.28}$$

- If a sentence node is selected then it must match a node of at least one of the patterns:

$$\sum_{p_u \in P} \sum_{i \in V_{p_u}} \hat{x}_{i,k} \geq \hat{s}_k \quad \forall k \in V_g \tag{4.29}$$

Now we can set up our experiments for evaluating our approach to coherence patterns on the summarization task.

4.4.2.3 Data

We evaluate our model on two datasets: *PLOS Medicine* and *DUC 2002*. The PLOS Medicine dataset consists of 50 scientific articles. We are motivated to evaluate our model on scientific articles because of the growth in the number of scientific publications in different research fields. A summarizer assists researchers to have an informative and coherent gist of long scientific articles. Moreover, summarizing a scientific article is challenging because a scientific article tends to be long and presents important information in various sections of the article, unlike the distribution of information in a news article (Teufel and Moens, 2002). The reason that we selected the PLOS Medicine dataset is that articles in this dataset are accompanied by summaries written by editors of the month. Editors' summaries have a broader perspective than abstracts of articles. We use scientific articles and their corresponding editor's summaries as dataset D in our formulation for the summarization task (see Section 4.4.2). Abstracts of scientific articles can be taken as summaries of articles as well. We collect abstracts of 700 scientific articles from the PubMed[10] corpus, which is in the bio-medicine field, to mine coherence patterns and compute their weights. This dataset of abstracts is dataset H in our formulation for the summarization task. The articles of this dataset do not overlap with articles in the PLOS Medicine dataset.

We also evaluate our model on the DUC 2002 dataset that has been annotated for the Document Understanding Conference 2002. It contains 567 news articles for summarization. Every article in this dataset is associated with at least two gold summaries written by humans. This is dataset D in our formulation, and we use this dataset for the evaluation purposes. We use human summaries in the DUC 2005 dataset, which has 300 articles, to mine coherence patterns and then calculate the weights of patterns. This is dataset H in our formulation.

Texts in DUC 2002 are shorter than those in *PLOS Medicine* (25 vs. 154 sentence average lengths). In scientific articles clarity is paramount, so their authors endeavor to state things explicitly and avoid ambiguity. Scientific authors repeat terminology to be explicit. In contrast, in literature, word repetition is not only uncommon, but it is usually a sign of bad writing.

[10] `http://www.ncbi.nlm.nih.gov/pmc/tools/ftp/`

4.4.2.4 Experimental Settings

In the preprocessing phase, we extract texts from scientific articles by removing all figures, tables, references and non-alphabetical characters. We use the Stanford parser (Klein and Manning, 2003) to parse sentences of articles in all datasets. We represent each text by its entity graph that is obtained based on its entity grid representation. We employ the Brown coherence toolkit (Elsner and Charniak, 2011b) to build entity grids from parse trees. We use gSpan (Yan and Han, 2002) to extract all coherence patterns from the projection graphs in GS_H. We use coherence patterns with three and four nodes to which we refer as 3-node and 4-node patterns respectively. The optimization is formulated as Mixed Integer Programming (MIP) that deals with quadratic constraints, like the constraint in Equation 4.27. We use Gurobi (Gurobi Optimization, Inc., 2014) to solve the MIP optimization problem. The values of weights of the importance, non-redundancy and coherence functions are fine-tuned on the development sets of the corresponding datasets. The best values for the *PLOS Medicine* development set are $\lambda_I = 0.4$, $\lambda_R = 0.3$, and $\lambda_c = 0.3$. Weights for the DUC 2002 development set are $\lambda_I = 0.5$, $\lambda_R = 0.2$ and $\lambda_c = 0.3$. Once a summary is produced, all pronouns in the summary are substituted with their antecedents using the pronoun resolution system provided by Martschat (2013). We limit the length of summaries to 5 sentences, where we compare our system with the state-of-the-art systems on PLOS Medicine. However, since the word length limit of a summary is more reasonable than the sentence length limit of a summary, in addition we compare the examined summarization models where the length of a summary is restricted to the average length of editor's summaries in the dataset (750 words). We use Student's t-test to test statistical significance of our improvments.

4.4.2.5 Compared Summarization Systems

We compare our summarization system, which is enriched with coherence patterns, with the following summarization systems:

- *Random*: This model selects sentences randomly from the input document;
- *Lead*: This model takes the top n$\%$ of the sentences of the input document[11];
- *Maximal Marginal Relevance (MMR)* (Carbonell and Goldstein, 1998): This model uses a trade-off between relevance and redundancy to rank sentences. To do this, the model

[11] In our experiment it extracts five first sentences of articles

defines a linear ranking function as:

$$MMR(s_i) = \arg\max_{s_i}\{\gamma sim(s_i, title) + (1-\gamma)\max_{s_j} sim(s_i, s_j))\} \tag{4.30}$$

where γ is the trade-off factor. If γ equals to one then the sentences in the input document are ranked merely based on their similarity with the title, which means ranking based on relevance. If γ equals to zero, sentences are ranked based on the similarity among themselves, which can be interpreted as redundancy.

- *Text-Rank* (Mihalcea and Tarau, 2004): This graph-based model allows for the ranking of sentences that are recursively computed based on the information drawn from the entire text. It represents the input document by a graph whose nodes represent sentences and edges indicate the existence of content overlaps between sentences. Edges are weighted by the number of content words that overlap between sentences. The ranking of sentences is measured by computing the importance of their corresponding nodes in the graph. The node importance in the graph is computed based on the global information which is recursively drawn from the entire graph. The importance score of each node in each recursion is updated concerning the importance score of its neighbors in the graph.

- *EntOD* (Parveen and Strube, 2015): This model uses ILP in order to optimize the summary based on importance, non-redundancy, and coherence. The importance and non-redundancy components are identical to these components in the summarization system that is explained here. The only difference between the EntOD system and our summarization system is in the coherence component. In EntOD, the input document is represented by the entity graph to encode entity-based relations among sentences. Then the outdegree of a node in the projection graph representation of the input document is taken as a measure of the contribution of the corresponding sentence to the coherence of the summary. That is why we refer to this model as EntOD.

- *TopicOD* (Parveen et al., 2015): This summarization system is the same as the EntOD system except in the way that texts are represented. TopicOD uses topical graphs, instead of entity graphs, to encode topical relations among sentences. Topical graphs are bipartite graphs consisting of two sets of nodes: sentences and topics. The outdegree of each sentence in weighted projection graphs is taken as the coherence measure of the sentence.

- *Mead* (Radev et al., 2004a): This model assigns a score to each sentence of the input document using three scores. The centroid score, which is a measure of the centrality of a sentence to the overall topic of the input document; the position score, which decreases linearly as the sentence gets farther from the beginning of the input document; and the overlap-with-first score, which is the inner product of the TF*IDF-weighted vector representations of a sentence and the first sentence, or the title of the input document if it has one. Mead discards sentences that are too similar to other sentences (based on the cosine similarity). Any sentence which is not discarded due to high similarity and which obtains a high score (within the specified compression rate) is included in the summary.

4.4.2.6 Results

We evaluate the summarization system, which is enriched by coherence patterns in two ways. First, we use ROUGE scores to compare our summarizer with other models. Second, we explicitly evaluate the coherence of summaries by human judgments.

ROUGE assessment. The ROUGE score (Lin, 2004) is a standard evaluation metric for automatic text summarization. It principally measures word overlaps between gold summaries (usually generated by humans) and summaries produced by a model. ROUGE-1, ROUGE-2, and ROUGE-SU4 are three versions of ROUGE that are popularly reported for comparing different summarization systems. ROUGE-1 and ROUGE-2 capture unigram and bigram overlap between a gold summary and a produced summary. These are meant to assess informativeness. ROUGE-SU4 captures skip-bigram plus unigram-based co-occurrence statistics. We refer interested readers to Graham (2015) for more explanations about evaluation metrics for the summarization task.

Table 4.5 reports ROUGE scores of different systems on the *PLOS Medicine* dataset where the length of the summaries is limited to five sentences. Our summarization system that uses three nodes outperforms other systems. It works better than EntOD and TopicOD systems showing that our coherence patterns are more informative than the average outdegree feature.

Table 4.6 shows the performance of different systems with 750 words limit for a summary where editor's summaries are taken as gold standard. We calculate different variations of ROUGE-2 and ROUGE-SU4. These variations demonstrate the effect of stop words and stemming in the ROUGE calculation. For the sake of brevity, we use the notation “SW” to refer to stop words and “SM” to refer to word stemming. “SW–” shows that stop words are not taken into account in ROUGE calculation; “SW+” is the opposite. “SM–” shows that the

Systems	**ROUGE-SU4**	**ROUGE-2**
Random	0.048	0.031
Lead	0.067	0.055
MMR	0.069	0.048
TextRank	0.068	0.048
Mead	0.084	0.068
TopicOD	0.129	0.095
EntOD	0.131	0.098
3-node	**0.135**	**0.103**

Table 4.5: ROUGE scores on PLOS Medicine, and five-sentence summaries.

Porter Stemmer is not applied to summaries in ROUGE calculation, “SM+” is its opposite.

Our model achieves the best performance in comparison to other examined systems with respect to all variations of ROUGE. When we integrate coherence patterns with three nodes into the summarizer, i.e. 3-node, the summarizer significantly outperforms EntOD that uses the outdegree of sentence nodes as the coherence feature. These results confirm our argument at the beginning of this section: The outdegree of nodes in the projection graph of the input document is not a powerful representative for the coherence of selected sentences for a summary. 3-node works better than EntOD because our coherence patterns capture the connectivity style among selected sentences from the input document for the summary, whereas the outdegree measures to what extent a sentence is connected to other sentences in the input document, rather than the summary. Moreover, the outdegree does not capture how sentences should be connected to have a coherent summary.

When we integrate 4-node coherence patterns into the summarizer, the summarizer works slightly better than when 3-node patterns are combined. This confirms that large subgraphs capture more information about the connectivity style of nodes in a projection graph and therefore the coherence of sentences. However, this improvement is not statistically significant. 4-node patterns are less likely than 3-node patterns to occur in a subgraph of the projection graph where the subgraph contains only selected nodes for the summary.

The summarizer that is enriched by our coherence patterns outperforms *Mead* as one of the strong summarization systems. Summaries produced by *Mead* on average contain fewer sentences than summaries produced by 3-node patterns (17.5 vs. 27.2 sentences per summary). This observation shows that *Mead* selects longer sentences in comparison to our 3-node pat-

PLOS Medicine	**SW–**	**SW–**	**SW+**	**SW+**	**SW–**	**SW–**	**SW+**	**SW+**
	SM+	**SM–**	**SM+**	**SM–**	**SM+**	**SM–**	**SM+**	**SM–**
	ROUGE-SU4 ($*p < .05$)				ROUGE-2 ($*p < .01$)			
Random	0.140	0.113	0.169	0.153	0.102	0.088	0.125	0.116
Lead	0.191	0.158	0.246	0.222	0.158	0.140	0.185	0.171
MMR	0.183	0.149	0.240	0.215	0.141	0.125	0.171	0.157
TextRank	0.148	0.104	0.161	0.159	0.115	0.084	0.126	0.118
Mead	0.197	0.165	0.246	0.222	0.156	0.139	0.186	0.172
TopicOD	0.195	0.161	0.231	0.206	0.157	0.140	0.169	0.165
EntOD	0.204	0.167	0.254	0.228	0.160	0.145	0.187	0.173
3-node	0.215^*	0.178^*	0.268^*	0.241^*	0.172^*	0.153	0.200^*	0.184^*
4-node	$\mathbf{0.218}^*$	$\mathbf{0.179}^*$	$\mathbf{0.270}^*$	$\mathbf{0.245}^*$	$\mathbf{0.175}^*$	$\mathbf{0.156}$	$\mathbf{0.201}^*$	$\mathbf{0.187}^*$

Table 4.6: ROUGE scores on PLOS Medicine, and 750-word summaries. In each column, numbers with an asterisk are significantly better than the underlined number in that column.

terns. Long sentences are more complicated, less readable, and may also contain more irrelevant entities than short sentences.

Table 4.7 shows the results on DUC 2002 of well-performing systems in the previous experiment. In addition to other models, we compare our model to *NN-SE* that utilizes a neural network hierarchical document encoder and an attention-based extractor to extract sentences from a document for a summary (Cheng and Lapata, 2016). We observe that the performance of the employed summarizer improves when it uses our 3-node coherence patterns to measure coherence rather than the outdegree feature (TopicOD and EntOD). This observation confirms our intuition that our patterns encode coherence better than the out degree feature used by Parveen and Strube (2015). The differences in ROUGE scores obtained by EntOD and 3-node summarizers are not statistically significant (Student t-test, p_value$<$ 0.05). The ROUGE scores of our summarization approach, i.e. 3-node in Table 4.7, on this dataset surpass the scores of other summarization systems. This shows that our system performs well even in a different domain and with considerably short input texts. Our model outperforms the *NN-SE* system because our model explicitly takes into account the connectivity of selected sentences in the sentence extraction phase. We only use 3-node patterns on this dataset because the summaries are supposed to be very short (100 words).

Systems	ROUGE-1	ROUGE-2	ROUGE-SU4
Lead	0.459	0.180	0.201
TextRank	0.470	0.195	0.217
DUC 2002 Best	0.480	0.228	–
Mead	0.445	0.200	0.210
NN-SE	0.474	0.230	–
TopicOD	0.481	0.243	0.242
EntOD	0.485	0.230	0.253
3-node	**0.490**	**0.247**	**0.258**

Table 4.7: ROUGE scores on DUC 2002, and 100-word summaries.

Coherence assessment. Here we exclusively assess the coherence aspect of summaries by asking human judges to rank summaries that are generated by different systems. To this end, we ask four human judges[12] to rank summaries of four different systems for ten different articles. The most coherent summary is assigned with rank 1, the second best is assigned with rank 2, the third best gets rank 3, and the worst obtains rank 4. The four summarization systems are *3-node*, *EntOD*, *Text-Rank*, and *Lead*.

The Kendall concordance coefficient (W) (Siegel and Castellan, 1988) over the rankings is calculated in order to measure the agreement between the human judges. We calculate Kendall's W for every scientific article, which is given to the human subjects. Then, we calculate the average of Kendall's W of scientific articles. The average Kendall's W is 0.6725, which indicates a high level of agreement between human subjects. Applying the χ^2 statistical test shows that W is statistically significant (p_value <.05) indicating that the rankings provided by the human judges are reliable and informative.

Table 4.8 shows the overall average rankings summaries produced by a system received by human judges. *Lead* obtains the best overall average rank because it extracts adjacent sentences from the beginning of the text. Hence, the summaries produced by this system are coherent as the author intends them to be. Our summarizer, which is enriched with 3-node coherence patterns, follows LEAD by outperforming EntOD and TextRank, showing that the integration of our coherence patterns into the summarization system yields texts that are more coherent in comparison to summaries that are produced by baseline summarizers such as EntOD that uses the outdegree.

[12] Human judges are one PostDoc, two Ph.D. students and one Master student in the NLP group at HITS.

System	Average human scores
TextRank	3.950
EntOD	2.325
3-node	**1.875**
Lead	1.625

Table 4.8: The average human scores evaluated on the PLOS Medicine dataset. Lower is better. The bold line is our system.

4.5 Summary

In this chapter, we challenged the average outdegree metric that is heuristically defined by the entity graph model. The primary intuition behind the usage of this metric is that texts whose projection graphs have higher average outdegrees than others are more coherent. We showed that the average outdegree of nodes in a projection graph is not sufficient to capture the connectivity style of nodes and consequently the perceived coherence of the corresponding text. Instead, we proposed novel coherence patterns that capture the entity-based connectivity style of sentences in texts. We employed projection graphs of texts in a corpus to encode the connectivity style of sentences. Then by applying a subgraph mining algorithm to all projection graphs of all texts, we mine all occurring subgraphs in these graphs as coherence patterns. We use the frequency of each coherence pattern in a projection graph as a feature of the connectivity style of nodes in the projection graph and consequently, a feature of the perceived coherence of the corresponding text.

We evaluated our coherence patterns in two applications: readability assessment and extractive single-document summarization. In the former, we observed that frequencies of some coherence patterns positively and some others negatively correlate with readability ratings, which are assigned to texts by human judges. Positively correlated patterns mostly depict the intuition that a sentence introduces some entities, and its subsequent sentences elaborate on each of them. Negatively correlated patterns roughly remind us of the linear chain pattern in linguistics where a sentence is located between two sentences with different topics to make the topic change smooth across sentences. This pattern is an indicator of a topic shift. Although topic shifts make a text appealing, too frequent occurrences of this pattern in a text disturb the readability of the text. Our experiments showed that 4-node patterns are more predictive than 3-node patterns in ranking texts with respect to their readability. We believe that this is mainly because large patterns have more capacity than small ones for encoding the connectivity style of nodes in projection graphs.

In the summarization task, we examined our coherence patterns by integrating them into a baseline summarization system that is developed on the entity graph representation of texts. This task was challenging because we had to model the existence of our coherence patterns in a summary by defining several novel constraints in linear programming. The results of our experiments on DUC 2002 as a benchmark dataset for summarization and PLOS Medicine as a corpus of scientific articles show that our coherence patterns significantly improve the performance of the baseline summarizer with respect to ROUGE and human evaluation.

The key message of this chapter is that in order to capture the connectivity style of sentences in a text, which is encoded via the entity graph, coherence patterns, which are obtained automatically by applying a subgraph mining algorithm, are more useful than average outdegree, which is designed heuristically. Coherence patterns capture coherence by taking each sentence in its connections with other sentences in the text. Our data-driven approach to coherence pattern mining enables our model to extract patterns from texts so that a machine learning model can learn relations among the patterns systematically.

We observe that 4-node patterns are better coherence patterns than 3-node patterns. However, more investigation is required to be performed on how the size of patterns influences the performance of the model. The entity graph does not include mentions that are semantically related. It is restricted to only noun overlap relation among nouns in sentences, while any lexical semantic relation between words in sentences can relate sentences. We follow these points in the following chapter of this thesis.

The main contributions of the research presented in this chapter are:

- assessing the average outdegree metric for coherence measurement,
- proposing subgraphs of projection graphs as coherence patterns and their frequencies as features which encode coherence,
- evaluating our coherence patterns in ranking texts with respect to their coherence property,
- showing how coherence patterns can be utilized in readability assessment as a text quality evaluation task, and the text summarization task as an instance of text generation systems.

5 Lexical Cohesion Graph

An essential type of sentence relations is the semantic relationships among words. This type of connectivity in linguistics is known as lexical cohesion. In this chapter, we devise a method for identifying and representing lexical cohesion in texts. We first motivate the usage of lexical relations in a text for coherence modeling (Section 5.1). We then explain a graph-based approach for modeling lexical relations among sentences in a text (Section 5.2). We employ a sampling method for extracting subgraphs from graph representations of texts (Section 5.3). We explain the sparsity problem related to the frequencies of subgraphs and then adapt a solution from statistical language modeling methods for this problem for graphs (Section 5.4). We assess the lexical cohesion graph representations of texts and the impact of the smoothing method on two datasets for the readability assessment task (Section 5.5). Finally, we provide a summary of the research presented in this chapter (Section 5.6).

5.1 Lexical Cohesion

Coherent texts are beyond arbitrary choices of words and sentences. In such texts, sentences are related to each other to smooth the flow of information in texts. As we discussed in Chapter 2, this is achieved through cohesive semantic relations that are expressed through the grammar and the vocabulary of a language (Halliday and Hasan, 1976). The former is referred to as grammatical local coherence and the latter as lexical cohesion. Here we focus on lexical cohesion, which comprises one of the semantic connections among words in a text (Hoey, 1991).

The basis of lexical cohesion is, in fact, extended to any pair of lexical items that stand next to each other in some lexico-semantic relation. Some examples of such relations are as follows[1]:

- Repetition, which happens when a word in a sentence is repeated in another sentence;

[1] We refer readers for more information to Chapter 2 and Chapter 3.

- Synonymy, which happens when a word in a sentence means exactly or nearly the same as a word in another sentence. For example, the verbs "buy" and "purchase" have a synonymy relationship with each other;
- Hyperonymy, which shows the relationship between a generic word (hypernym) in a sentence and a specific instance of it (hyponym) in another sentence. For example, there exist a hyperonymy relationship between the words "red", "blue" and "color";
- Meronymy, which happens when a word in a sentence is a constituent part of or a member of the concept that is mentioned by a word in another sentence. For example, "finger" and "hand" are in the meronymy relationship because a finger is part of a hand;
- Antonymy, which occurs when two words are semantically opposite to each other, e.g., the words "willing" and "reluctant" are an antonym of each other.
- Collocation, which happens when two words frequently occur in the same context together. As an example, the words "doctor" and "patient" are semantically related just because they are frequently used together.

An essential property of lexical relations is that lexical items should not necessarily have the same reference in order to relate two sentences (Halliday and Hasan, 1976). Consider the sample text[2] that is presented in Example (9).

(9) Why does the little **boy** wriggle all the time?
Girls don't.

The lexical items "boy" and "girls" do not refer to the same entity but they make these two sentences related because they are semantically related.

In order to recognize lexical semantic relations between words, a lexical knowledge resource is needed. One option is to use a lexical resource such as WordNet (Fellbaum, 1998) or Freebase (Bollacker et al., 2008). This option is expensive in terms of determining the best resource. WordNet lacks a broad coverage in particular with proper names, and Freebase is restricted to nominal concepts and entities. Besides, lexical resources similar to WordNet and Freebase are rarely available for other languages than English. If they are available, their coverage is not as broad as their versions for English. The other alternative for identifying lexical relations is to employ a set of pretrained word embeddings. Word embeddings which are trained on large corpora of texts can capture word relationships in a language. Embedding

[2] Taken from the *Cohesion in English* book written by Halliday and Hasan (1976).

representations of words give the model the capability to efficiently encode semantic relations among lexical items in a vector space. If words are semantically related in the text space, their embeddings are similar to each other in the vector space. Therefore, the semantic relations between words can simply be measured by the cosine function over the angle between the corresponding word vectors. It is worth to mention that such word vectors can be easily trained for any language if a large corpus of texts on the language is available (see more about word embeddings in Chapter 3).

In the research presented in this thesis, we use word embeddings to check whether there exists a relationship between two words or not. There is no straight way to determine the type of a relation between two words by means of the cosine function. However, this is something that we do not need for the purpose of the research presented in this thesis. Halliday and Hasan (1976) argue that for the texture purposes, it is only necessary to recognize lexical items that relate to each other.

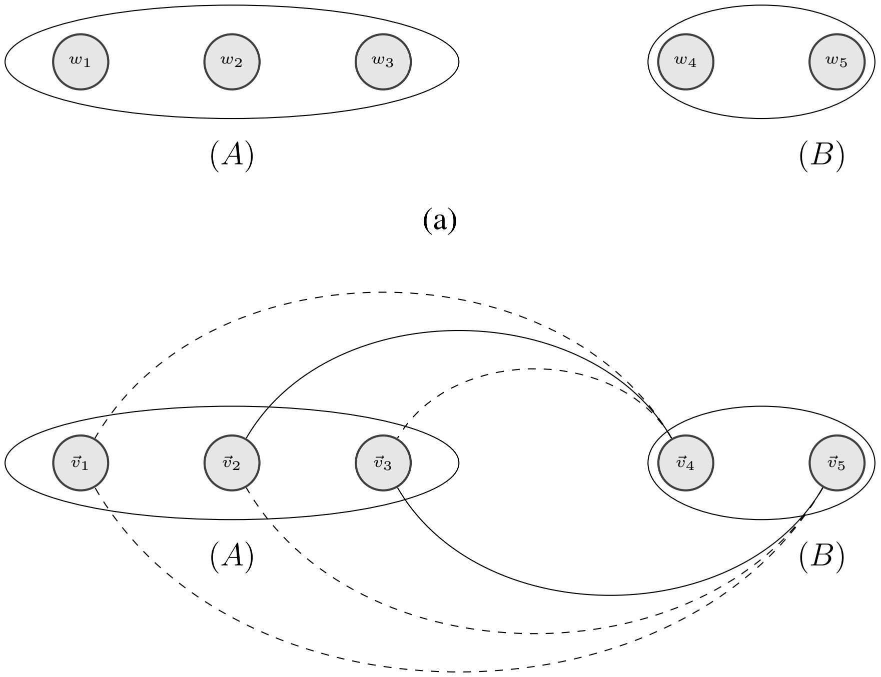
5.2 LexGraph

Since graph representations of entity-based relations across sentences in a text have been shown to be useful for modeling coherence (see Chapter 4), we focus on providing a graph representation of lexical semantic relations across sentences in a text. This graph is built on the existence of lexical relations among words in sentences. We refer to this graph representation of a text as lexical cohesion graph or *LexGraph*.

More formally, LexGraph $G =< V, E >$ comprises two sets: V is a set of nodes representing sentences in a text, and E is a set of directed edges. The direction of an edge between two nodes indicates the order of the sentences that are associated with the nodes. The edge itself represents the existence of two word vectors with high cosine similarity, which is a proxy of a lexical semantic relation between words and therefore their corresponding sentences.

Inspired by the lexical cohesion theory (Halliday and Hasan, 1976), two sentences are semantically connected if a pair of their words semantically relate to each other. Semantic relations between words are modeled by their corresponding pretrained word embeddings. Given word vector $\vec{v_i}$ for word w_i and word vector $\vec{v_j}$ for word w_j, the value of the cosine metric between these two word vectors, $\texttt{cos}(\vec{v_i}, \vec{v_j})$, measures the strength of the relation between word w_i and word w_j. The range of the cosine metric is in the interval $[-1, +1]$. One interpretation of the cosine metric is the normalized correlation coefficient of its inputs. This metric quantifies the relatedness between the words associated with two input word vectors (Manning and Schütze, 1999). The absolute value of the cosine metric, $|\texttt{cos}(\vec{v_i}, \vec{v_j})|$, encodes how strongly two words are related.

Figure 5.1 illustrates our approach to model the connection between two sentences as an edge in a LexGraph.



(a)

(b)

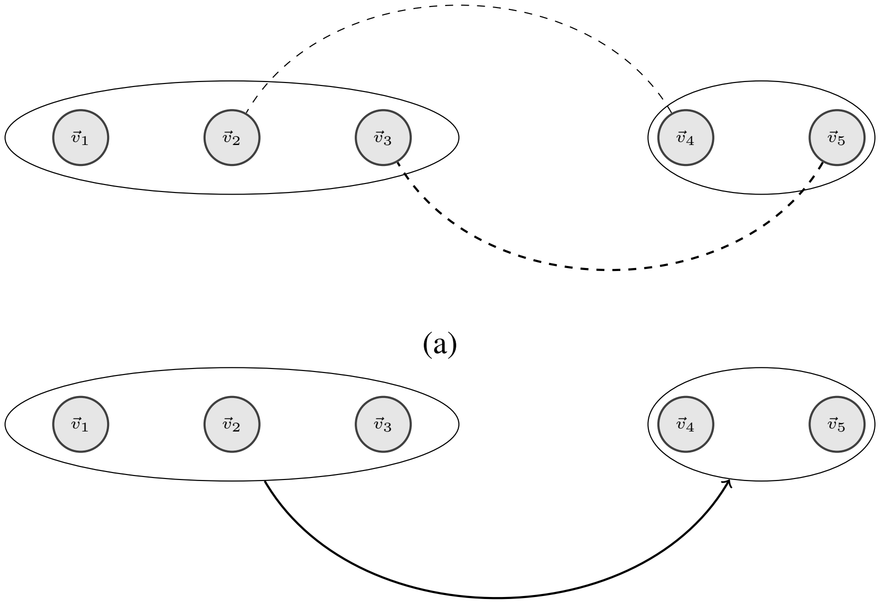
Figure 5.1: (a) Sentence A with three words $\{w_1, w_2, w_3\}$ precedes sentence B with two words $\{w_4, w_5\}$. (b) Words of sentences are replaced by their associated pre-trained word embeddings. Different type of connections represents various edge weights, i.e., the cosine metric between vector pairs. Solid edges have higher weights than dashed edges. The cosine metric of $\vec{v}_4$ and $\vec{v}_2$ is the highest among all edges that are connected to $\vec{v}_4$. The same interpretation holds for the connection between $\vec{v}_5$ and $\vec{v}_3$.

The connection between two sentences is defined based on the lexical relations among words in the sentences. Assume sentence A with three words $\{w_1, w_2, w_3\}$ precedes sentence B with two words $\{w_4, w_5\}$ in a text. Words in the sentences are mapped to their associated pre-trained word embeddings. Vectors $\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$ are word embeddings associated with words in sentence A and vectors $\{\vec{v}_4, \vec{v}_5\}$ represent words in sentence B. For each word in B, we compute its relatedness with any word in sentence A by computing the absolute value of the cosine function between the embeddings of words. We consider $|\texttt{cos}(\vec{v}_j, \vec{v}_i)|$ as the weight of the edge that connects vector $\vec{v}_j$ in B to vector $\vec{v}_i$ in A. Finally, the edge with the maximum weight is selected to capture the lexical relation between word w_j in sentence B with the most semantically related word in sentence A. More concretely, word w_j in sentence B is connected with word w_i^* in sentence A such that

$$\vec{v}_i^* = \arg\max_{\vec{v}_i \in A} \mathtt{cos}(\vec{v}_j, \vec{v}_i), \tag{5.1}$$

where $\vec{v}_i$ and $\vec{v}_j$ are vector representations of words w_i and w_j. We use the connection with the largest weight because the semantic relation of any two words of two sentences, from the texture prespective, is a linguistic device to connect the sentences (Halliday and Hasan, 1976).

Then from all connections among the vector pairs in A and B, the connection with the maximum weight is selected to connect the nodes associated with these two sentences in the LexGraph representation of the text (Figure 5.2).



(a)

(b)

Figure 5.2: (a) The word relation with the maximum weight (b) represents the connections between sentences.

We follow the above procedure for all sentence pairs in a text. It results in a graph in which weighted edges connect every two nodes. We prune all edges whose weights are below a fixed threshold. This threshold is a parameter of the model. From the texture perspective, it ensures that relations that are made based on strong lexical relations are taken into account, and weak relations are filtered out. From the computational point of view, this parameter makes the graphs sparse so that the model can distinguish the differences in the connectivity structures of graphs.

5.3 Coherence Pattern Mining

We employ an approach similar to the method presented in Chapter 4 to capture the connectivity style of their LexGraph, i.e. lexical cohesion graph, representation of a text. A lexical cohesion graph encodes the lexical relations among sentences in a text. Given such graph representations of texts in a corpus, we apply a subgraph mining algorithm to these graphs in order to extract all patterns occurring in the graph representations of texts. For the sake of consistency, we use the term k-node to refer to the size, the number of nodes, of subgraphs. The value of k is fixed for extracting subgraphs. We take subgraphs as patterns and use their frequencies in each LexGraph as features. The features capture the connectivity style of the graph and, consequently, the coherence of the corresponding text.

In the subgraph mining method presented in Chapter 4, we employed gSpan as a basic subgraph mining algorithm to extract 3-node and 4-node subgraphs from all graphs. However, the gSpan method does not count the induced subgraphs by itself. We needed to develop a function to recount the extracted patterns as induced subgraphs in graphs. Additionally, the gSpan method extracts only connected subgraphs, while subgraphs with several components might be useful for coherence modeling as well. Therefore, we introduce another approach to simultaneously extract and count induced subgraphs, which could be connected or disconnected, as coherence patterns.

5.3.1 Pattern Mining: Sampling

In order to extract k-node subgraphs and compute their frequencies, we resort to a sampling approach (Weissman et al., 2003; Shervashidze et al., 2009). Assume that we construct all possible k-node subgraphs in advance, and we save them in a lookup table, which is called *pattern-list*. We need to define this pattern-list only once, so the processing time of this step does not have any negative impact on the overall efficiency of the mining method. The idea is to sample k-node subgraphs from graphs, and if a sample hits one of the patterns in the list, then the count of the pattern in the graph increases by one. Ideally, if a sufficient number of sample subgraphs are drawn from a graph, then the empirical distribution is close to the actual distribution of patterns in the graph (Shervashidze et al., 2009). We follow Algorithm 5.1 to count subgraphs in lexical cohesion graphs. Function `Generate` performs the sampling task. It selects k random nodes and the edges that connect these nodes from the input graph. Function `GetID` compares its input subgraph with each pattern in the pattern-list and returns the index of the pattern that is induced and isomorphic with the input subgraph.

Algorithm 5.1. Pattern counting.

```
Input: A list of graphs $L$, a list of k-node patterns $P$, a pattern size $k$, a sampling threshold
    MAX, a function Generate, a function GetID
function SubgraphSampling($L, P, k,$ MAX, Generate, GetID)
    Set $N_l$ to the number of graphs in $L$
    Set $N_p$ to the number of patterns in $P$
    Set $C[0..N_l][0..N_p] = 0$
    Set $t = 0$
    while $t < N_l$ do
        Set $g = L[t]$
        Set $count = 0$
        while $count <$ MAX do
            Set $s =$ Generate$(g, k)$
            Set $p =$ GetID$(P, s)$
            Set $C[t][p] = C[t][p] + 1$
            Set $count = count + 1$
        Set $t = t + 1$
Output: $C$
```

The complexity of the method presented in Algorithm 5.1 is $\mathcal{O}(N^{*}\texttt{MAX})$ where N is the number of graphs (i.e. the number of texts in a corpus) and `MAX` is the maximum number of samples that should be drawn from each graph, which is shown by `MAX`. It is worth to note that because the counting procedure of patterns in a graph (i.e. the inner loop in the function) is independent of the counting procedure in other graphs, the Algorithm 5.1, in practice, is implemented in parallel over graphs.

5.4 Smoothing

With reference to the results of experiments in Chapter 4, by increasing the size (i.e. parameter k) of subgraphs, patterns capture more structural information about the connectivity of nodes. Specifically, 4-node subgraphs contain more nodes and edges, so they potentially capture more information about the connectivity of graphs than what 3-node subgraphs capture. Besides, by enlarging the subgraphs, the number of patterns and consequently the number of features increase as well. However, many large subgraphs do not occur in the graph representation of a text yielding a sparsity problem in the feature vector, which represents the connectivity structure of the graph. On the other hand, large subgraphs are unlikely to occur in all graphs. They have zero frequencies in most graphs and non-zero frequencies in a few graphs. On the contrary, small subgraphs frequently occur in many graphs (so they have non-zero frequencies in most graphs) but they are not as informative as large subgraphs about the connectivity style of graphs. In order to use the predictive power of large subgraphs, we need to overcome the sparsity problem in graphs.

The sparsity issue in subgraphs may lead to two problems. Machine learning methods may become biased to some features because they occur only in a few graphs. The other problem is that if a large pattern has not been seen in training data (i.e. it has zero frequency in all graphs during the training) then the pattern is not informative for graphs that contain the pattern during the test phase.

The sparsity issue happens in the statistical language models as well (Jurafsky and Martin, 2008). These models use *N-gram* (which is a contiguous sequence of N words from a text) features in probabilistic language models. Long N-gram features have zero frequencies in many texts. The solution proposed in language modeling is smoothing, which deals with the problem of zero counts in feature vectors. It introduces a pseudo frequency for N-gram features that are not seen during training but are plausible for prediction in the test phase. One of the well-known smoothing methods in language modeling is Kneser-Ney smoothing (Ney et al., 1994). In this technique, the probability of a long N-gram is computed based on its actual frequency and the frequencies of short N-grams that are part of the long N-gram.

We show that a smoothing technique can also solve the sparsity issue in graphs. We adopt the Kneser-Ney smoothing method. This approach provides a trade-off between the predictive power of large subgraphs and frequently occurring small subgraphs. It estimates the frequency of a large subgraph based on the frequencies of smaller subgraphs. It allows the model to estimate frequencies of patterns in a graph even where subgraphs are not present in the graph.

In order to use the Kneser-Ney smoothing, we need to extract not only all possible k-node subgraphs but all subgraphs whose sizes are less than k. The sampling approach for subgraph mining introduced in Section 5.3 efficiently fulfills this requirement.

Inspired by the Kneser-Ney smoothing method in language models, a vector representation of a graph can be smoothed such that the model computes estimated frequency values for unseen subgraphs that may be seen in the testing phase. Kneser-Ney smoothing uses discount factor α to discount the raw count of pattern p in graph g, which is denoted by $count(p, g)$.

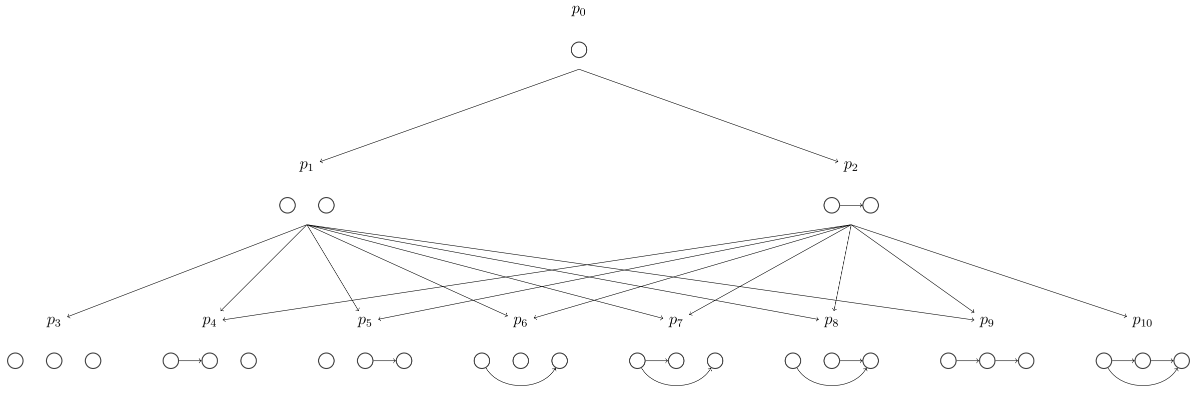


Figure 5.3: Hierarchical relations among patterns up to three nodes, where a connection from a pattern to another pattern shows that the former pattern is a subgraph of the latter one. The former pattern is taken as the parent and the latter one as the child in their relationship.

It then distributes the total discount to all pattern probabilities by means of a base probability P_b. The smoothed probability of pattern p in graph g is computed as follows:

$$KN(p,g) = \frac{\texttt{max}\{count(p,g) - \alpha, 0\}}{Z} + \frac{M \cdot \alpha}{Z} P_b(p), \tag{5.2}$$

where M is the number of times that the discount factor is applied. Variable Z is a normalization factor to ensure that the probability distribution sums to one. For a set of k-node patterns, which is represented by A, the value of Z is obtained as follows:

$$Z = \sum_{p \in A} count(p,g). \tag{5.3}$$

$P_b(p)$ in the Kneser-Ney formulation (Equation 5.2) is the base probability of pattern p among all k-node patterns. It is computed based on hierarchical relations among patterns. Figure 5.3 shows hierarchical relations between patterns with up to three nodes. Each level of this tree contains all patterns with a certain number of nodes. A k-node pattern p_i is connected to a (k+1)-node pattern p_j if pattern p_i is a subgraph of pattern p_j. We refer to such a relationship between two patterns as the parent-child relation, where pattern p_i is the parent of pattern p_j. We illustrate this relation by the direction of the edge between patterns in Figure 5.3. As an example, consider pattern p_1, pattern p_2, and pattern p_5 in Figure 5.3. Pattern p_1 and pattern p_2 are parents of pattern p_5 because they both are subgraphs of pattern p_5.

The weight of a connection from pattern p_i to pattern p_j is the frequency of pattern p_i as a subgraph in pattern p_j:

$$w_{ij} = \frac{count(p_i, p_j)}{\sum_{p_l \in A} count(p_i, p_l)}, \tag{5.4}$$

where A is all patterns with k-node and k equals the number of nodes in pattern p_j. In other words, weight w_{ij} is the normalized count of pattern p_i in pattern p_j with respect to the counts of other children of pattern p_i, i.e. all patterns which are connected to pattern p_i by outgoing edges from p_i. We use such weighted hierarchical relationships between patterns to compute base probabilities of patterns. The base probability of pattern p_j is the inner product of the Kneser-Ney probabilities of its parents considering the weights of their relations with p_j:

$$P_b(p_j) = P \cdot W, \tag{5.5}$$

where P is a vector of Kneser-Ney probabilities, i.e. $= KN(.,.)$ of all patterns that are parents of p_j, and W is the weight vector of relations between patterns.

This smoothing method traverses the tree recursively from large subgraphs to small subgraphs. We assume that the probability of the parent of pattern p_0 is one because its parent is a graph with no nodes, i.e., a subgraph of any graph. Because the weights of connections in the hierarchical relations among subgraphs are normalized in the interval of $[0, 1]$, the sum of the probabilities of all patterns with k-node is always equal to one. It is a necessary condition, which must hold to have a probability distribution among patterns.

Proof. Assume I and J are the sets of all k-node and (k+1)-node patterns, respectively; and set I has N patterns and set J has M patterns. Given the following assumption:

$$\sum_{i=1}^{N} p_b(p_i) = 1, \tag{5.6}$$

we prove that

$$\sum_{j=1}^{M} p_b(p_j) = 1. \tag{5.7}$$

We start to compute the sum of probabilities of all patterns in set J, which is $\sum_{j=1}^{M} p_b(p_j)$ in Equation 5.7. Based on the definition of the base probability, the value of $p_b(p_j)$ is computed with respect to the probabilities of its parents in I:

$$p_b(p_j) = \sum_{i=1}^{N} w_{ij} p_b(p_i), \tag{5.8}$$

where w_{ij} is the weight of the parent-child relation between pattern p_i and pattern p_j. Now we have:

$$\sum_{j=1}^{M} p_b(p_j) = \sum_{j=1}^{M} \sum_{i=1}^{N} w_{ij} p_b(p_i). \tag{5.9}$$

If we exchange the place of the summations in Equation 5.9 and rewrite the equation, we have:

$$\sum_{j=1}^{M} p_b(p_j) = \sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} p_b(p_i). \tag{5.10}$$

In Equation 5.10, $p_b(p_i)$ is independent of j (i.e. the index of the inner summation), so it can be moved out of the inner summation:

$$\sum_{j=1}^{M} p_b(p_j) = \sum_{i=1}^{N} p_b(p_i) \sum_{j=1}^{M} w_{ij}. \tag{5.11}$$

Finally, the sum over w_{ij} is equal to 1 because weights of relations among patterns are normalized (see Equation 5.4). If we replace this sum with 1, the result is as follows:

$$\sum_{j=1}^{M} p_b(p_j) = \sum_{i=1}^{N} p_b(p_i). \tag{5.12}$$

Based on our assumption that the sum of probabilities of patterns in set I is equal to 1, we have:

$$\sum_{j=1}^{M} p_b(p_j) = 1. \tag{5.13}$$

Therefore, the sum of the base probabilities of all (k+1)-node subgraphs is 1. This proof can recursively be applied through the different levels of patterns in the hierarchical tree. The recursion stops at the root of the tree, which is a pattern with one node. This pattern occurs in any graph and its probability is always one. So the assumption that $\sum_{i=1}^{N} p_b(p_i) = 1$ is a valid assumption. □

5.5 Experiments

We evaluate our LexGraph model through some experiments. In order to gain a better insight into the impact of the size of patterns on their predictive power, we experiment with different sizes of patterns. We investigate the problem of sparsity in the frequencies of subgraphs and then show that the smoothing approach proposed in this chapter of the thesis can overcome the sparsity problem.

We employ readability assessment as the running evaluation task in the research of this thesis. We approach readability assessment as the task of ranking texts with respect to their readability. The intuition is that more coherent texts are easier to read. So a coherence model ideally ranks texts similar to the rankings provided by humans.

5.5.1 Data

We run our experiments on two readability datasets. Texts in these datasets are annotated with readability information by human annotators. The first one is the dataset that is used in the experiments presented in Chapter 4. This dataset is provided by Pitler and Nenkova (2008) and we refer to this dataset as *P&N* in this chapter. It contains 27 news articles that are randomly selected from the Wall Street Journal corpus. The average number of sentences per text is about 10. Every article is associated with a human score between $[0.0, 5.0]$ indicating the readability ratings of that article. We create pairs of texts if the difference between the readability ratings of texts is higher than 0.5. If the first text in a pair has a higher score, we label the pair with $+1$, otherwise with -1. The number of text pairs in this dataset is 209.

The second readability dataset that is used for evaluating our coherence model is provided by De Clercq et al. (2014). We refer to this readability dataset as the *De Clercq* dataset in the experiments in this chapter. The *De Clercq* dataset consists of 105 articles from four different genres: administrative, journalistic, manuals, and miscellaneous. The average number of sentences of texts in this dataset is about 12. De Clercq et al. (2014) annotated texts in this dataset by asking human judges to compare two texts with respect to their readability. They use five labels:

- **LME:** left text is much easier,
- **LSE:** left text is somewhat easier,
- **ED:** both texts are equally difficult,
- **RSE:** right text is somewhat easier,
- **RME:** right text is much easier.

We map these labels to three class labels $\{-1, 0, +1\}$. Label $+1$ is assigned to text pairs in which left texts are easier to read (LME or LSE); 0 is employed for text pairs in which both texts are equally difficult to read (ED); and finally label -1 is associated with text pairs in which the right texts are easier to read (RSE or RME). The dataset in total contains 10907 text pairs, among which 3146 pairs have label $+1$, 3146 pairs have label -1, and the rest have label 0. Table 5.1 summarizes some properties of the different genres in this dataset.

Genre	Number of articles	Number of text pairs
Administrative	17	272
Journalistic	43	1806
Manuals	14	182
Miscellaneous	31	931
All	105	3191

Table 5.1: Some properties of the different genres in the *De Clercq* dataset.

5.5.2 Experimental Settings

We explain the settings that are considered in the experiments presented in this chapter.

Word embeddings. In order to reduce the effect of frequent words, stop words are eliminated by using the SMART English stop word list (Salton, 1971) in pre-processing. We use GloVe[3] (Pennington et al., 2014) as pre-trained word embeddings to measure semantic relatedness between words. Word embeddings in GloVe are trained on Common Crawl with 840B tokens and 2.2M vocabulary. The length of each word vector is 300. For handling out-of-vocabulary words, we assign a random vector to each word and memorize it for its next occurrence.

Graph processing and smoothing. In order to compare the text representations provided by LexGraph with the representations that are provided by the entity graph model, we first use the gSpan method (Yan and Han, 2002) to extract subgraphs that are occurring in graph representations of texts in a corpus. In this way, the only difference between models is the graph representations of texts. All other settings are identical.

[3] GloVe is publicly available at `https://nlp.stanford.edu/projects/glove/`.

We extend the entity grid representations of texts by incorporating the connections between pronouns and their antecedents. To this end, we apply the Stanford coreference resolution system (Lee et al., 2013) to resolve all pronouns. Involving the full coreference relations, however, decreases performance; hence we only use resolved pronouns. In the LexGraph representations of texts, edges whose weights are less than threshold 0.9 are filtered out. We selected this value to connect only sentences that are strongly related. In this way, we prevent noisy edges, which are not strong enough in terms of the cosine similarity, to be involved in graphs. Adding noisy edges to lexical cohesion graphs makes the discrimination between graphs difficult. A similar issue happens where full coreference relations are incorporated in the entity graph representation of texts (Guinaudeau and Strube, 2013).

We check the impact of the size of subgraphs on the performance of the model by evaluating patterns with a different number of nodes $k \in \{3, 4, 5, 6\}$. We need to count all possible k-node subgraphs because the probability should be distributed among all possible subgraphs. In the experiments in which we use smoothing, we compute the frequencies of coherence patterns by the sampling method that is explained in this chapter. For sampling, we draw $10,000$ samples from each graph. We compute the base probability for all subgraphs with $k \leq 6$. The best value for the discount factor, i.e. parameter α in Equation 5.2, is obtained greedily and iteratively. First, we initialize α with 0.001. In each iteration, we compute the performance on the test set[4]. Then we multiply the discount factor by 10. We iterate as long as the discount factor is less than 1000. We report the best performance.

Machine learning model and evaluation. The classification task is performed by the Support Vector Machine (SVM) implementation in WEKA, i.e. SMO, with the linear kernel function. We evaluate our model by 10-fold cross-validation, and use the Student t-test to test the significance of improvments. We ensure that text pairs in the training and test folds do not overlap.

Compared models. We compare our LexGraph model with the EGraph model. In the LexGraph model, each text is represented by a graph which is built upon lexico-semantic relationships between words in sentences. We employ a subgraph mining method to extract all possible subgraphs from graphs as coherence patterns. The frequencies of patterns in a graph are taken as features, which encode the connectivity structure of the graph and ideally the local coherence of the corresponding text. We compare this model with the same method on the entity graph representation, which is referred to as EGraph, of texts. It is worth noting

[4] We report the best possible performance on the test set.

that in this chapter the EGraph model uses the projection graph representations of texts and employs our coherence pattern mining approach to extract coherence features, rather than the average outdegree metric. We keep all the experimental settings identical for these systems to only investigate the impact of text representations on the performance of the coherence model. Besides, we compare these models with the EGrid model (Barzilay and Lapata, 2008) as a non-trivial baseline.

5.5.3 Results

We categorize the results of experiments on LexGraph in three groups. We first evaluate how LexGraph representations of texts perform in comparison to the entity graph representations. We then assess the influence of the smoothing approach on the performance of the LexGraph model. We finally evaluate the quality of the coherence patterns that are extracted from the LexGraph representations of texts and compare them with patterns that are extracted from the entity graph representations.

Evaluating LexGraph representations. We evaluate LexGraph representations of texts on the *P&N* dataset. Figure 5.4 reports the accuracies of different models on the *P&N* dataset. We can observe from the results that both the graph-based models, i.e. EGraph and LexGraph, rank texts with respect to their coherence superior to the EGrid model (Barzilay and Lapata, 2008) for $k > 3$. This observation confirms our initial intuition in the research in this thesis that graph-based models have more capacity to encode relations among sentences in a text because graphs capture long-distance relations, which is informative for coherence modeling.

The other observation of the results presented in Figure 5.4 is that the accuracies of both LexGraph and EGraph models increase by enlarging the size of patterns. This observation is compatible with the results reported in Chapter 4, and confirms the intuition that large subgraphs capture more information about the connectivity style of graph representations of texts. Large patterns can lead to features that are highly predictive for discriminating coherent texts from incoherent ones.

Given 3-node patterns, the LexGraph model does not beat the EGraph model. Considering 4-node patterns, the performance of LexGraph is close to the performance of the EGraph model. Finally, when 5-node patterns are utilized the LexGraph model significantly outperforms the entity graph model. These results can be explained such that the lexical graph representation of texts have more edges than the entity graph representations. Since the LexGraph is dense, 3-node subgraphs cannot encode the differences between the structure of graphs. How-

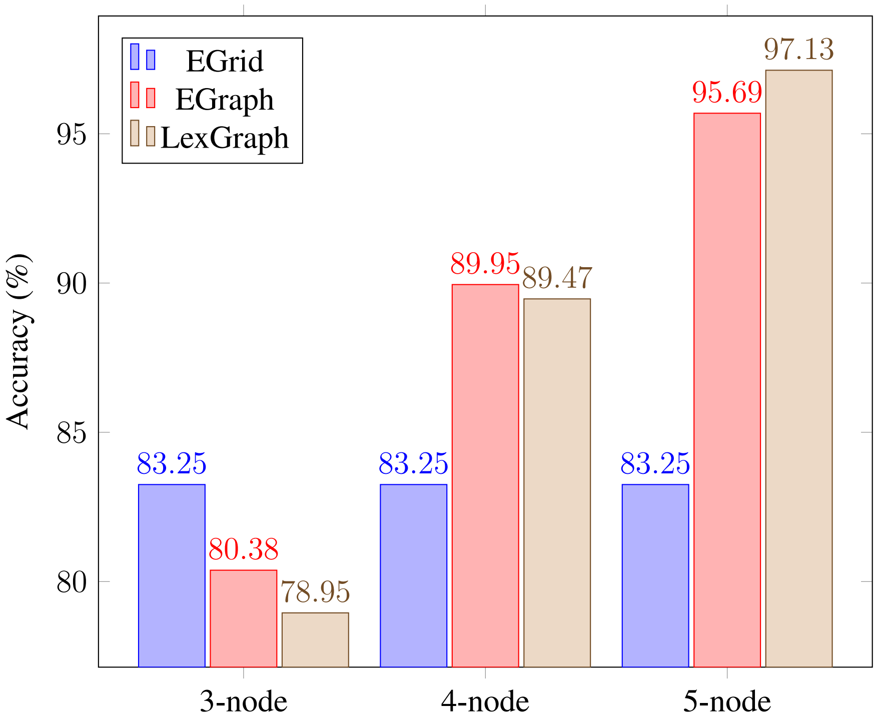


Figure 5.4: The accuracies of different systems on the *P&N* dataset.

ever, when sufficiently large patterns are employed we see that the LexGraph representation is more predictive than the entity graph.

The best result on the *P&N* dataset is about 97%. This performance leads us to evaluate our model on the *De Clercq* dataset for further investigation. The *De Clercq* dataset contains more articles than the *P&N* dataset. The other major difference between these two datasets is that the articles in the *De Clercq* dataset are from different genres, but those in the *P&N* dataset are only from one genre that is news. Different genres may follow different styles of connectivity, which makes this dataset challenging.

Figure 5.5 shows the performance of different models on the *De Clercq* dataset. The accuracies of both EGraph and LexGraph models are on par with the baseline system for 3-node patterns. Since these patterns frequently occur in all graphs, they are not able to distinguish texts. 4-node patterns lead both EGraph and LexGraph to work superior to the baseline. However, these patterns are not large enough to capture the connectivity style of dense graphs such as LexGraphs. Finally, 5-node patterns yield reasonable performance on the *De Clercq* dataset. In this case, the LexGraph model outperforms the EGraph model significantly. 5-node patterns are sufficiently large to encode the connectivity structure of nodes in LexGraphs, and consequently the local coherence of texts.

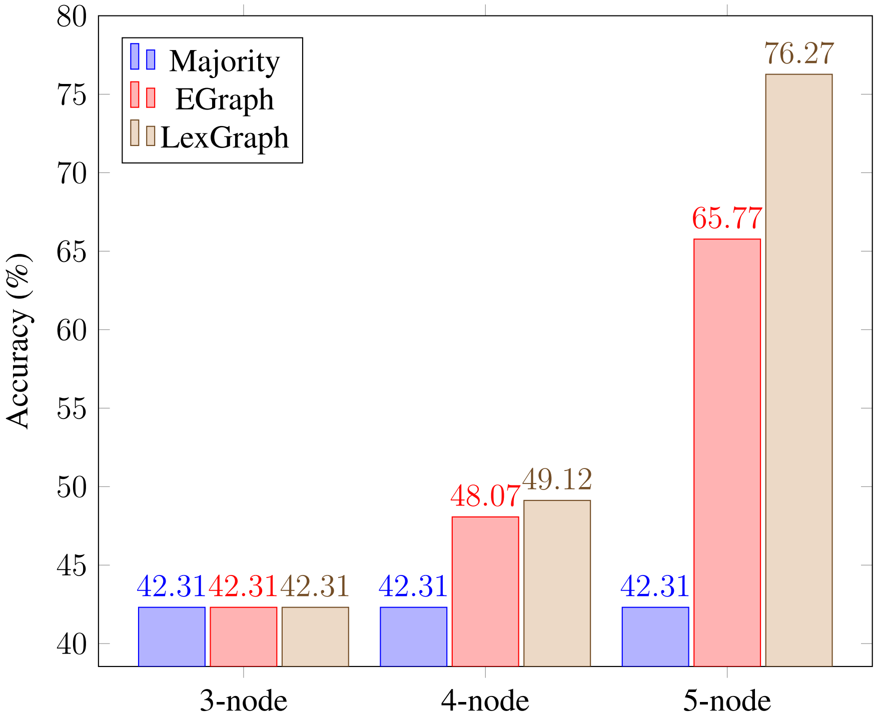


Figure 5.5: The accuracies of different systems on the *De Clercq* dataset.

When we compare the EGraph model and the LexGraph model on these two datasets, we observe that the general performance of these models on the *De Clercq* dataset is lower than the performance on the *P&N* dataset. It can be because the texts in the *De Clercq* dataset are from four different genres. These genres may have various styles of connectivity. In order to gain a deeper insight into this, we compute the accuracies of these systems on texts exclusively from each genre in the *De Clercq* dataset. We use 5-node patterns because these patterns can distinguish between texts better than other patterns for any of the two examined graph representations of texts, i.e., the entity graph and the LexGraph representations.

Figure 5.6 shows the performance of the EGraph model and the LexGraph model using 5-node patterns in different genres in the *De Clercq* dataset. The performance of the LexGraph model is higher than the EGraph model for all genres. This observation is compatible with the results reported in Figure 5.5. On the Administrative articles, the difference between the accuracies of the models is not substantial. In these texts, the exact repetition of words among sentences is persistent to ensure that texts are entirely unambiguous. Unlike the EGraph model, the LexGraph model achieves the best performance on Journalistic articles. The Journalistic texts use more variations of words to relate sentences, so the LexGraph representation is more

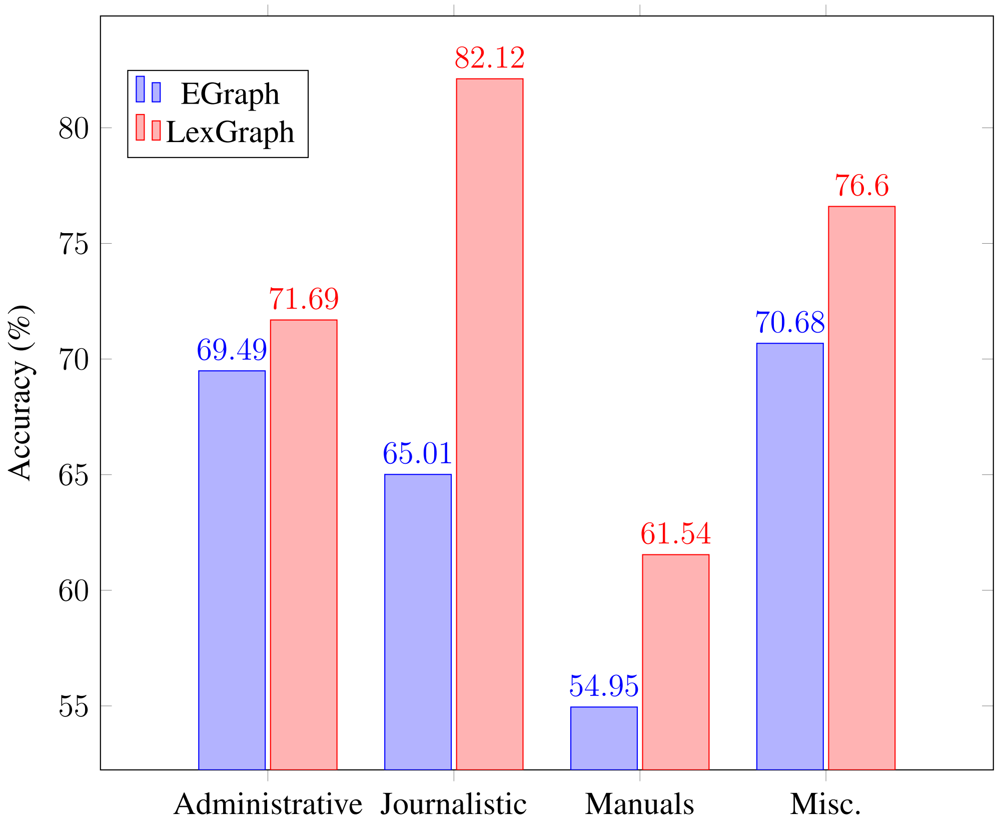


Figure 5.6: The accuracies of different systems on the various genres of articles collected in the *De Clercq* dataset.

predictive than EGraph. The lowest performance of both models is obtained for texts that are Manuals; however, the LexGraph model with a large margin outperforms the EGraph model on this genre as well.

Evaluating the smoothing approach. While large patterns are very informative for coherence modeling (especially for dense graphs such as LexGraphs), many large patterns have low or zero frequency in a graph. It yields a sparsity problem in the vector representations of graphs where large patterns are employed. On the other hand, large patterns occur in a few graphs in a graph set. So when large pattern-list are taken into account, each graph is represented by a high dimensional vector because there are many possible subgraphs, where most of the elements in the vector are zero. The problem with such vectors is that each graph representation roughly becomes unique and a machine learning model cannot learn from similarities and dissimilarities between vectors. We evaluate our solution that is Kneser-Ney smoothing.

Figure 5.7 shows the performance of the LexGraph model on the *P&N* dataset, where the smoothing method is employed to overcome the sparsity problem in frequencies of patterns in graph representations of texts. The performance of the model increases by enlarging the size of patterns up to five nodes. When 6-node patterns are employed, the performance of the model drops. This result can be because of the sparsity problem. When we use smoothing the performance of the model for any pattern size improves in comparison to the settings without smoothing. We observe that the drop in performance from 5-node patterns to 6-node patterns when smoothing is applied is less than the drop in their performance without smoothing. In other words, the performance of the system is more even with smoothing rather than when smoothing is not applied.

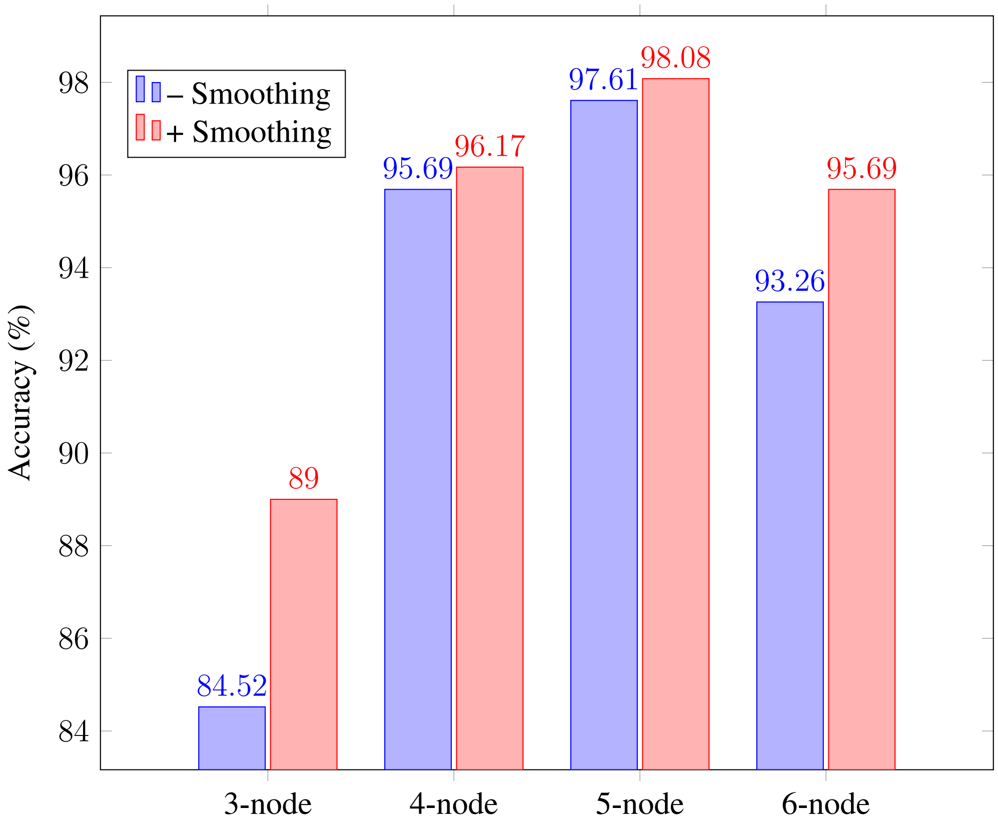


Figure 5.7: The effect of applying smoothing on the performance of the LexGraph on the *P&N* dataset.

Figure 5.8 depicts the results of this experiment on the *De Clercq* dataset. By applying Kneser-Ney smoothing, the results for all examined values of k improve on this dataset. Interestingly, on both datasets applying Kneser-Ney smoothing enhances the performance of the model with a large margin where 3-node subgraphs are employed. Smoothing makes the frequency distribution of subgraphs more even. It decreases the effect of frequency through

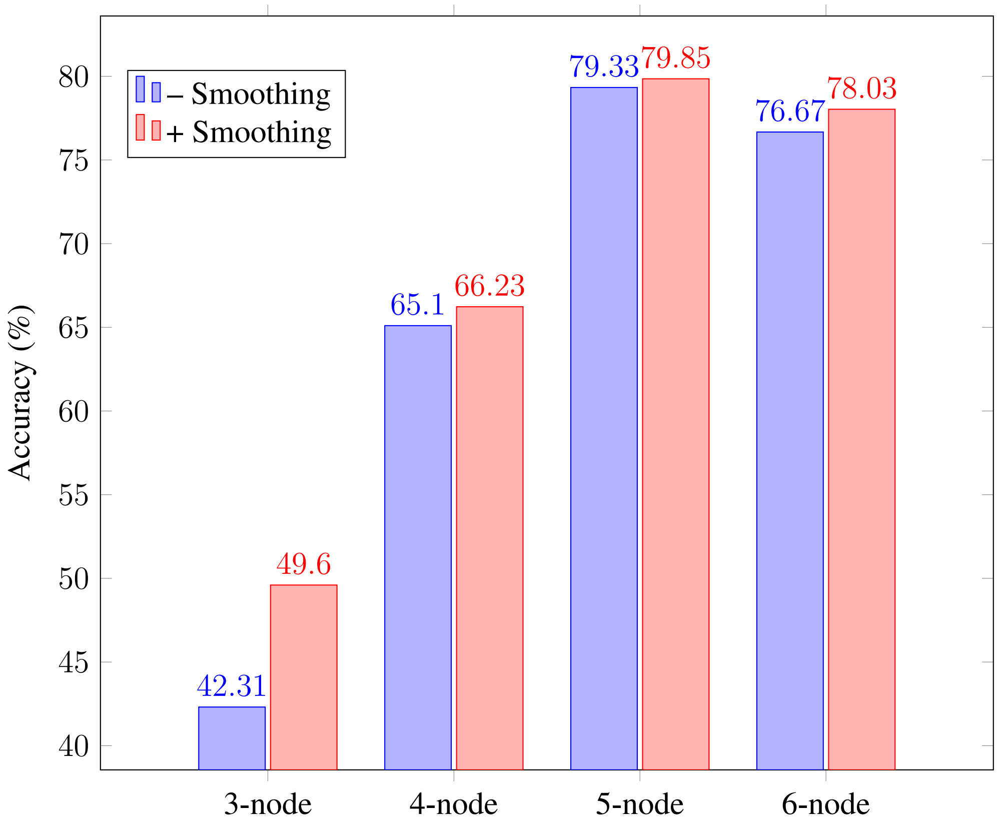


Figure 5.8: The effect of applying smoothing on the performance of the LexGraph on the *De Clercq* dataset.

all subgraphs by considering parent-child relations between subgraphs to relate similar subgraphs. That is the advantage of the Kneser-Ney method in comparison to the other smoothing methods like Laplace smoothing (Jurafsky and Martin, 2008).

In general, we observe the similar trend in results of the LexGraph model on the *De Clercq* dataset and on the *P&N* dataset. It shows that our coherence patterns plus the LexGraph representation of a text construct a model for local coherence. It is also worth to mention that none of the parameters (such as the maximum number samples drawn from a graph and the threshold for filtering edges of LexGraph) in this work is tuned on the datasets. One may get better performance by tuning the parameters.

Mined coherence patterns. In this experiment, we compute the Pearson correlation coefficient between the frequencies of few patterns in the LexGraphs of texts in the *P&N* dataset and the readability ratings that are assigned to texts by humans.

Table 5.2 (see page 111) shows the patterns whose frequencies are significantly (p_value <0.05) correlated with readability ratings assigned by humans.

Among all 3-node patterns, only the frequency of one pattern in the LexGraph representations of texts in this dataset is positively correlated with the readability ratings that are assigned by humans. Among the patterns with four nodes, frequencies of six patterns are significantly correlated with the readability ratings. The frequency of only one pattern is positively correlated, while the other five patterns are negatively correlated. Interestingly, both positively correlated 3-node and 4-node patterns have been determined as positively and significantly correlated with human readability ratings in experiments that are performed in the research presented in Chapter 4. It indicates that our coherence model is linguistically sound.

5.6 Summary

In the research presented in this chapter, we introduced a graph-based approach, named LexGraph, for representing the relations among sentences in a text based on the lexical relations among the words of sentences. We employ pre-trained word embeddings to identify lexical relations between words in sentences. This approach provides more informative graph representations of texts for coherence modeling in comparison to the entity graph representations of texts because the relations between sentences in LexGraph are not limited to coreference relation among entities. Relations in LexGraph are built upon the semantic relations between any word pair in sentences. In this chapter, we extracted and counted subgraphs by a sampling approach for modeling coherence patterns and features.

While the entity grid model works only on sequences of up to two adjacent sentences, we can model relationships of up to six non-adjacent sentences. We solve the sparsity problem of large subgraphs by adapting Kneser-Ney smoothing to graphs. Smoothing prevents LexGraph from losing performance with large subgraphs. It leads to superior performance on the Pitler and Nenkova (2008) dataset and to a first reasonable state-of-the-art on the De Clercq et al. (2014) dataset.

On the *P&N* dataset, we achieve the best results to date. Pitler and Nenkova (2008) reported 83.25% accuracy. In Chapter 4, by applying the idea of using frequency of subgraph as coherence features on the entity graph representation of texts, the model obtains 89.95% accuracy. In the research of this chapter, by providing the lexical cohesion graph over sentences in texts and applying a smoothing method to frequencies of 5-node subgraphs, we could report 98.08% accuracy. These results, however, indicate that this dataset may not be the best one to report performance on and evaluate our coherence model. We observed that smoothing improves the performance of our coherence model on the *De Clercq* dataset as well.

To conclude this chapter, the results of experiment presented in this chapter confirm our primary intuition that the capturing lexical relations among words in sentences via graphs is beneficial for coherence modeling. Applying the smoothing method on graphs of EGraph model increases the performance of this model. However, this improvement is not high as the improvement obtained by LexGraph. These results show that our smoothing method is useful for both graph-based models, and our new graph representation, i.e. LexGraph, is more informative than the entity graph for coherence modeling.

We summarize the contributions of this chapter as follows:

- proposing a new graph-based representation of lexico-semantic relations across sentences,
- adapting the Kneser-Ney smoothing approach in order to solve the sparsity problem in frequency of large coherence patterns,
- evaluating the model on two readability datasets: Pitler and Nenkova (2008) and De Clercq et al. (2014).

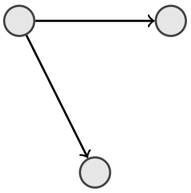
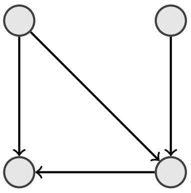
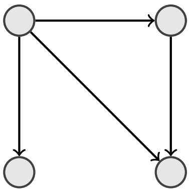
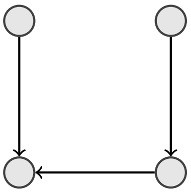
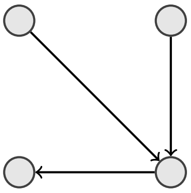
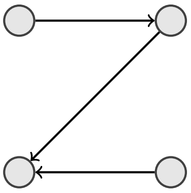
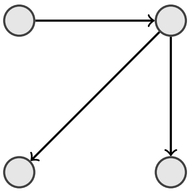
	Pattern	ρ	P_value
3-node		+0.43	0.024
4-node		-0.45	0.018
		+0.39	0.047
		-0.43	0.024
		-0.59	0.001
		-0.55	0.003
		-0.55	0.003

Table 5.2: The Pearson correlation coefficient between frequencies of 3-node and 4-node patterns that are significantly correlated with readability ratings that are assigned by human judges to texts in the *P&N* dataset.

6 Conclusions

The aim of the research presented in this dissertation was to provide an approach to local coherence modeling based on the connectivity structure of relations among sentences in a text. To this end, we defined two major research questions:

- **Do there exist nonlinear connectivity patterns in coherent texts such that the patterns capture long-distance relations among sentences?**
- **How can we model sentence relations in a text beyond coreference relations over entities by considering semantic relations between words in sentences?**

In this chapter, we revisit the research questions and summarize our contributions towards answering the questions (Section 6.1). Furthermore, we discuss some avenues for future work (Section 6.2).

6.1 Contributions

In this dissertation, we considered two main research questions. We now discuss how the research presented in this thesis contributes to answering these research questions.

A graph-based approach to coherence pattern mining. We employed the entity graph representations to encode entity coreference relations among sentences in a text. Graphs enable our model to involve connections between non-adjacent sentences as well as connections between adjacent sentences. We formalized the concept of connectivity patterns in linguistics (Daneš, 1974; Stoddard, 1991) by extracting all subgraphs that occur in graph representations of texts in a corpus. We referred to these subgraphs as coherence patterns. We represented the connectivity structure among nodes in a graph by a vector where each of its elements is the frequency of one of the extracted patterns in the graph. We observed some promising correlation between the frequencies of coherence patterns in texts and readability ratings assigned by humans. We used these vectors to train a machine learning method for ranking texts with respect to their coherence. Finally, we learned that by enlarging coherence patterns, i.e. increasing the number of nodes in subgraphs, the performance of our model improves.

In another experiment, we evaluated our coherence patterns on extractive text summarization. For doing so, we integrated subgraphs, which are extracted from a set of coherent summaries, into the process of sentence selection in a summarization system. We observed that summaries that are generated by considering our coherence patterns are more readable and coherent for human judges. We also noticed that our coherence patterns improve the performance of the summarization system with respect to the ROUGE metrics.

Developing an approach to coherence modeling based on lexical relations. In order to complete our approach, we proposed a new graph-based representation, which is called LexGraph, for sentence relations in texts rather than the projection graph representations used in the entity graph model. LexGraph representations are based on lexico-semantic relations between words in sentences. A pair of content words in two sentences makes a connection if there exists a lexico-semantic relation between them. We used pretrained word embeddings to find out if such relations exist between two words or not. Our representation connects more sentences in comparison to the entity graph representations. Therefore, the graphs provided by our model are denser than entity graphs. We extracted all subgraphs of lexical graphs as coherence patterns. First, we found out that some of these patterns are similar, in terms of structure, to patterns that are extracted from entity graphs and also the patterns presented by Daneš (1974). It shows that our LexGraph model is linguistically sound. We observed that frequencies of coherence patterns in lexical cohesion graphs are more predictive than those in entity graphs for the readability ranking task. We also noticed that large coherence patterns perform superior to the small ones on lexical cohesion graphs. However, there exists the risk of sparsity where coherence patterns become very large. We adapted Kneser-Ney smoothing for solving this problem, resulting in considerable improvements in the performance of our model.

6.2 Future Work

Based on the research presented in this dissertation, several possible ways for future work exist. Those can be in directions of either the coherence modeling method or the influence of the coherence model in other natural language processing applications. We discuss three possible extensions of the work presented in this thesis.

Using a machine learning method for coherence pattern mining. In the research of this dissertation, we used a graph-based approach, i.e. entity graph or lexical cohesion graph, to capture relations among sentences in a text. We then applied a subgraph mining method to graphs of texts in a corpus for obtaining coherence patterns. We introduced two methods for subgraph extractions. Our first method was an exhaustive search (Chapter 4), and our second method was sampling (Chapter 5). This process is independent of the machine learning method that is used to rank texts with respect to their coherence. Recent improvements in deep learning methods are promising to combine these two phases. Deep learning models (Goldberg, 2017) such as convolutional neural networks (CNNs) (Kim, 2014) can be employed to operate on graph representations of texts to extract coherence patterns that are especially beneficial for the ranking task.

Furthermore, in the proposed lexical cohesion graph representation, which is based on lexical relations among words in sentences, words are taken into account individually. In other words, sentences are taken as a bag of words while the structure within sentences (Louis and Nenkova, 2012) and the order of words in sentences provide some clues for coherence models. It has been shown that recurrent neural networks (RNNs) can overcome this weakness (Goldberg, 2017). These models sequentially take embeddings of each word in a sentence and at each word return a vector, which is called a state vector. State vectors contain information of their corresponding input word embeddings and information in embeddings of other words in a sentence as context. As we discussed above, a CNN can be used on the top of the RNN states to extract coherence patterns automatically.

Analysis of coherence patterns for other NLP applications. In the research presented in this thesis, we evaluated our model on two readability assessment datasets and two summarization datasets. Our coherence model improved the performance of these systems. However, coherence is a crucial factor in other NLP applications as well. An example is the essay scoring task (Dikli, 2006; Higgins et al., 2004; Miltsakaki and Kukich, 2004). This task is about assigning a score to a student essay so that the score reflects the quality of the essay. Of course, the essay quality depends on more circumstances, such as grammatical mistakes, word lists that are used in essays, the similarity between the content of essays and the topic given for the essay, and so forth. Therefore, in order to employ this task for evaluating a coherence model, one should integrate frequencies of coherence patterns as features for the coherence of an essay into a feature-based essay scorer. Burstein et al. (2010) applied a similar strategy using entity transition features that are introduced in the entity grid model (Barzilay and Lapata, 2005) to model coherence.

Further, a similar approach can be applied to essays which are written by non-native speakers. Texts that are written by people with an identical mother tongue may reveal certain regularities in their sentences connectivities. We are curious to see if the coherence patterns presented in this thesis can distinguish essays based on the mother tongue of their authors. An applicable dataset for this task is TOEFL11 (Blanchard et al., 2013).

Analyzing coherence patterns for other domains and languages. Finally, although we demonstrated the generality of our method across different English corpora, we leave open the question of extensions to other languages and domains, where the specific patterns we detected may not exist.

List of Figures

3.1 The entity graph representation of the text from Example (7). The graph is obtained from the entity grid representation shown in Table 3.1. The top nodes represent columns in the grid or sentences in the text. The bottom nodes capture rows in the grid or entities in the text. Edges encode the entries in the grid. Weights of edges represent the value of each entry in the grid: 3:S, 2:O, 1:X, and 0:–. The weight of 0 is equivalent to no edge in a graph, so they are not drawn in the graph. 31

3.2 Three types of projection graphs that are employed by the entity graph model. P_U shows only which sentence nodes are connected. P_W takes the number of shared entities as the weight of edges. P_{Acc} involves grammatical roles of entities shared by sentences. 33

3.3 (a): A projection graph with the total outdegree of five where all nodes are in one component; (b): A projection graph with the total outdegree of five where nodes are in two components. 34

4.1 $G = (V, E)$ is a graph with node set V and edge set E. 50

4.2 The directed representation of graph G in Figure 4.1. 50

4.3 Two isomorphic graphs and a sample association between their nodes. 51

4.4 Graph G_2 is a subgraph of graph G_1. 51

4.5 Both graph G_2 and G_3 are subgraphs of graph G_1. In contrast to G_2, graph G_3 is an induced subgraph of G_1. 52

4.6 An illustration of the approach employed for extracting coherence patterns. . 54

4.7 Extracted 3-node patterns from the readability dataset. Nodes are in order $s_t < s_u < s_v$ to show the order of sentences in a text. 61

4.8 Extracted 4-node patterns, where order $s_t < s_u < s_v < s_w$ shows sentence order. 63

4.9 An illustration of matching variables for overlaying graph g with coherence pattern p_u. 77

5.1 (a) Sentence A with three words $\{w_1, w_2, w_3\}$ precedes sentence B with two words $\{w_4, w_5\}$. (b) Words of sentences are replaced by their associated pre-trained word embeddings. Different type of connections represents various edge weights, i.e., the cosine metric between vector pairs. Solid edges have higher weights than dashed edges. The cosine metric of $\vec{v}_4$ and $\vec{v}_2$ is the highest among all edges that are connected to $\vec{v}_4$. The same interpretation holds for the connection between $\vec{v}_5$ and $\vec{v}_3$. 92

5.2 (a) The word relation with the maximum weight (b) represents the connections between sentences. 93

5.3 Hierarchical relations among patterns up to three nodes, where a connection from a pattern to another pattern shows that the former pattern is a subgraph of the latter one. The former pattern is taken as the parent and the latter one as the child in their relationship. 97

5.4 The accuracies of different systems on the *P&N* dataset. 104

5.5 The accuracies of different systems on the *De Clercq* dataset. 105

5.6 The accuracies of different systems on the various genres of articles collected in the *De Clercq* dataset. 106

5.7 The effect of applying smoothing on the performance of the LexGraph on the *P&N* dataset. 107

5.8 The effect of applying smoothing on the performance of the LexGraph on the *De Clercq* dataset. 108

List of Tables

2.1 Coherence patterns that are defined by (Daneš, 1974). A horizontal arrow indicates a transition in an utterance, while a vertical one indicates a contextual connection within utterances. 17

3.1 The entity grid representation of the text presented in Example (7). The rows represent entities, and the columns encode sentences. If an entity is mentioned in a sentence, the corresponding entry in the grid indicates the grammatical role of the mention in the sentence (S: subject, O: object, X: others, and "–": none). 25

3.2 Transition patterns that are defined in the entity grid model. These patterns represent all possible entity occurrences in two adjacent sentences. Symbols S (subject), O (object), and X (others) show the grammatical role of an entity in a sentence. Symbol "–" encodes that an entity is not mentioned in a sentence. 26

3.3 Probabilities of the entity transition patterns, which are introduced in Table 3.2, in the entity grid shown in Table 3.1. 27

3.4 The average outdegree of nodes in projection graphs presented in Figure 3.2 . Dist. shows when distance is integrated in edge weights. 33

4.1 The Pearson and Spearman correlation coefficients and their p_value between the average outdegree feature of different projection graphs and readability ratings assigned by human judges. 62

4.2 The Pearson and Spearman correlation coefficients and their p_value between the frequency of 3-node patterns (see Figure 4.7) and readability ratings assigned by human judges. 64

4.3 The number of edges in 4-node patterns, and the Pearson correlation coefficient and their corresponding p_value between the frequency of patterns and human-provided readability ratings. 66

4.4 The accuracy and F-measure of the SVM classifier with different sets of features which represent coherence. $*$ indicates statistically significant (p_value$<$ .01) improvements with respect to the baseline features marked. S&O features are syntactic readability features presented by (Schwarm and Ostendorf, 2005) which lack coherence. 70
4.5 ROUGE scores on PLOS Medicine, and five-sentence summaries. 83
4.6 ROUGE scores on PLOS Medicine, and 750-word summaries. In each column, numbers with an asterisk are significantly better than the underlined number in that column. 84
4.7 ROUGE scores on DUC 2002, and 100-word summaries. 85
4.8 The average human scores evaluated on the PLOS Medicine dataset. Lower is better. The bold line is our system. 86

5.1 Some properties of the different genres in the *De Clercq* dataset. 101
5.2 The Pearson correlation coefficient between frequencies of 3-node and 4-node patterns that are significantly correlated with readability ratings that are assigned by human judges to texts in the *P&N* dataset. 111

List of Algorithms

5.1 Pattern counting. 95

Bibliography

Charu C. Aggarwal. 2018. *Machine Learning for Text.* Springer International Publishing, Cham, Switzerland.

Enrique Alfonseca, Daniele Pighin, and Guillermo Garrido. 2013. HEADY: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 1243–1253.

Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. 2004. Computing locally coherent discourses. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics,* Barcelona, Spain, 21–26 July 2004, pages 400–407.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation,* La Valetta, Malta, 17–23 May 2010, pages 2200–2204.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent and Scalable Text Summarization,* Madrid, Spain, 11 July 1997, pages 10–17.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics,* Ann Arbor, Mich., 25–30 June 2005, pages 141–148.

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

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 2–7 May 2004, pages 113–120.

Beata Klebanov Beigman and Michael Flor. 2013. Word association profiles and their use for automated scoring of essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 1148–1158.

Guosheng Ben, Deyi Xiong, Zhiyang Teng, Yajuan Lü, and Qun Liu. 2013. Bilingual lexical cohesion trigger model for document-level machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 382–386.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pages 481–490.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, Heidelberg, Germany.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native English. Technical report, Educational Testing Service.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(1):993–1022.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM-SIGMOD International Conference on Management of Data,* Vancouver, B.C., Canada, 10–12 June 2008, pages 1247–1250.

Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics,* Stanford, Cal., 6–9 July 1987, pages 155–162.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Proceedings of Human Language Technologies 2010: The Conference of the North American Chapter of the Association for Computational Linguistics,* Los Angeles, Cal., 2–4 June 2010, pages 681–684.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval,* Melbourne, Australia, 24–28 August 1998, pages 335–336.

Asli Celikyilmaz and Dilek Hakkani-Tür. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pages 815–824.

Asli Celikyilmaz and Dilek Hakkani-Tür. 2011. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pages 491–499.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Berlin, Germany, 7–12 August 2016, pages 484–494.

James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

Kevyn Collins-Thompson and James P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 2–7 May 2004, pages 193–200.

Edgar Dale and Jeanne S. Chall. 1949. The concept of readability. *Elementary English*, 26 (1):19–26.

František Daneš. 1974. Functional sentence perspective and the organization of the text. In F. Daneš, editor, *Papers on Functional Sentence Perspective*, pages 106–128. Prague: Academia.

Hal Daumé III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, Penn., 7–12 July 2002, pages 449–456.

Orphée De Clercq, Véronique Hoste, Bart Desmet, Philip Van Oosten, Martine De Cock, and Lieve Macken. 2014. Using the crowd for readability prediction. *Natural Language Engineering*, 20(3):293–325.

Pascal Denis and Jason Baldridge. 2009. Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*, 42:87–96.

Ludovic Denoyer and Patrick Gallinari. 2006. The Wikipedia XML corpus. *ACM-SIGIR Forum*, 40(1):64–69.

Márcio De S. Dias and Thiago A.S. Pardo. 2015. Enriching entity grids and graphs with discourse relations: The impact in local coherence evaluation. In *Proceedings of Symposium in Information and Human Language Technology,* Natal, Brazil, 4–7 November 2015, pages 151–160.

Teun A. Van Dijk. 1977. *Text and Context.* Longman, London, U.K.

Seimire Dikli. 2006. An overview of automated scoring of essays. *Journal of Technology, Learning, and Assessment*, 5(1):5–35.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Hawaii, 25–27 October 2008, pages 334–343.

Micha Elsner and Eugene Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings ACL-HLT 2008 Conference Short Papers,* Columbus, Ohio, 15–20 June 2008, pages 41–44.

Micha Elsner and Eugene Charniak. 2010. The same-head heuristic for coreference. In *Proceedings of the ACL 2010 Conference Short Papers,* Uppsala, Sweden, 11–16 July 2010, pages 33–37.

Micha Elsner and Eugene Charniak. 2011a. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pages 1179–1189.

Micha Elsner and Eugene Charniak. 2011b. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Portland, Oreg., 19–24 June 2011, pages 125–129.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics,* Rochester, N.Y., 22–27 April 2007, pages 436–443.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics,* Athens, Greece, 30 March – 3 April 2009, pages 229–237.

Vanessa Wei Feng and Graeme Hirst. 2012. Extending the entity-based coherence model with multiple ranks. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics,* Avignon, France, 23–27 April 2012, pages 315–324.

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *Proceedings of the 25th International Conference on Computational Linguistics,* Dublin, Ireland, 23–29 August 2014, pages 940–949.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* Seattle, Wash., 18–21 October 2013, pages 1481–1491.

Katja Filippova and Michael Strube. 2007. Extending the entity-grid coherence model to semantically related entities. In *Proceedings of the 11th European Workshop on Natural Language Generation,* Schloss Dagstuhl, Germany, 17–20 June 2007, pages 139–142.

Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychlogy*, 32(3): 221–233.

Michael Flor, Beata Beigman Klebanov, and Kathleen M. Sheehan. 2013. Lexical tightness and text complexity. In *Proceedings of the 2nd Workshop of Natural Language Processing for Improving Textual Accessibility,* Atlanta, Georgia, 14 June 2013, pages 29–38.

Peter Foltz, Walter Kintsch, and Thomas Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2):285–307.

George W. Furnas, Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, Richard A. Harshman, Lynn A. Streeter, and Karen E. Lochbaum. 1988. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th*

Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Grenoble, France, 13–15 June 1988, pages 465–480.

Dimitrios Galanis, Gerasimos Lampouras, and Ion Androutsopoulos. 2012. Extractive multi-document summarization with integer linear programming and support vector regression. In *Proceedings of the 24th International Conference on Computational Linguistics,* Mumbai, India, 8–15 December 2012, pages 911–926.

Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,* Sapporo, Japan, 7–12 July 2003, pages 562–569.

Daniel Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. 2009. A global optimization framework for meeting summarization. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Taipei, Taiwan, 19–24 June 2009, pages 4769–4772.

Yoav Goldberg. 2017. *Neural Network Methods in Natural Language Processing*. Morgan and Claypool Publishers.

Yvette Graham. 2015. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* Lisbon, Portugal, 17–21 September 2015, pages 128–137.

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

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 93–103.

Gurobi Optimization, Inc. 2014. Gurobi optimizer reference manual.

Udo Hahn and Inderjeet Mani. 2000. The challenges of automatic text summarization. *IEEE Computer*, 33(11):29–36.

M. A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English.* London, U.K.: Longman.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a lexical-semantic net for German. In *Proceedings of the Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications at ACL/EACL-97,* Madrid, Spain, 12 July 1997, pages 9–15.

Laura Hasler, Constantin Orasan, and Ruslan Mitkov. 2003. Building better corpora for summarization. In *Proceedings of Corpus Linguistics,* Lancaster, UK, 28–31 March 2003, pages 309–319.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics,* Las Cruces, N.M., 27–30 June 1994, pages 9–16.

Marti A. Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Michael J. Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics,* Rochester, N.Y., 22–27 April 2007, pages 460–467.

Derrick Higgins and Jill Burstein. 2007. Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics (IWCS),* Tilburg, The Netherlands, 10–12 January 2007, pages 1–12.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Centile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 2–7 May 2004, pages 185–192.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* Seattle, Wash., 18–21 October 2013, pages 1515–1520.

Michael Hoey. 1991. *Patterns of Lexis in Text.* Oxford University Press, Oxford.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge based from Wikipedia. *Artificial Intelligence*, 194:28–61.

Eduard H. Hovy and Kathleen F. McCoy. 1989. Focusing your RST: A step toward generating coherent multisentential text. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society,* Ann Arbour, Mich., 16–19 August 1989, pages 667–674.

Aravind K. Joshi and Scott Weinstein. 1998. Formal systems for complexity and control of inference: A reprise and some hints. In M.A. Walker, A.K. Joshi, and E.F. Prince, editors, *Centering in Discourse*, pages 31–38. Oxford University Press, Oxford, U.K.

Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Melbourne, Australia, 15–20 July 2018, pages 558–568.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, N.J., 2nd. edition.

Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence for text structuring using a reliably annotated corpus. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics,* Barcelona, Spain, 21–26 July 2004, pages 392–393.

Nikiforos Karamanis, Chris Mellish, Massimo Poesio, and Jon Oberlander. 2009. Evaluating centering for information ordering using corpora. *Computational Linguistics*, 35(1):29–46.

Rohit Kate, Xiaoqiang Luo, Siddharth Patwardhan, Martin Franz, Radu Florian, Raymond Mooney, Salim Roukos, and Chris Welty. 2010. Learning to predict readability using diverse linguistic features. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 546–554.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 1746–1751.

J. Peter Kincaid, Robert P. Jr. Fishburne, Richard L. Rogers, and Brad S. Chisson. 1975. Derivation of new readability formulas (automated readability index, Fog count and Flesch reading ease formula) for navy enlisted personnel. Technical Report 8-75, Naval Technical Training Command, Naval Air Station Memphis-Millington, Tenn.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,* Sapporo, Japan, 7–12 July 2003, pages 423–430.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

Manfred Klenner. 2007. Shallow dependency labeling. In *Companion Volume to the Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Prague, Czech Republic, 23–30 June 2007, pages 201–204.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization – step one: Sentence compression. In *Proceedings of the 17th National Conference on Artificial Intelligence,* Austin, Tex., 30 July – 3 August 2000, pages 703–711.

Susumo Kuno. 1972. Functional sentence perspective: A case study from Japanese and English. *Linguistic Inquiry*, 3(3):269–320.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th ACM-SIGIR Conference,* Seattle, Wash., 9–13 July 1995, pages 68–73.

Alice Lai and Joel Tetreault. 2018. Discourse coherence in the wild: A dataset, evaluation and methods. In *Proceedings of the SIGdial 2018 Conference: The 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue,* Melbourne, Australia, 12–14 July 2018, pages 214–223.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,* Sapporo, Japan, 7–12 July 2003, pages 545–552.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence,* Edinburgh, Scotland, 30 July – 5 August 2005, pages 1085–1090.

Liisa Lautamatti. 1978. Observations on the development of the topic in simplified discourse. *AFinLAn vuosikirja*, 8(22):71–104.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Julien Lerouge, Pierre Le Bodic, Pierre Héroux, and Sébastien Adam. 2015. GEM++: A tool for solving substitution-tolerant subgraph isomorphism. In *International Worksop on Graph-based Representation in Pattern Recognition,* Phékin, China, 13–15 May 2015, pages 128–137.

Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ILP for extractive summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 1004–1013.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 2039–2048.

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* Copenhagen, Denmark, September 7—11 2017, pages 198–209.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Text Summarization Branches Out Workshop at ACL '04,* Barcelona, Spain, 25–26 July 2004, pages 74–81.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics,* Montréal, Québec, Canada, 10–14 August 1998, pages 768–774.

Dekang Lin. 1998b. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics,* Montréal, Québec, Canada, 10–14 August 1998, pages 768–774.

Zhiheng Lin, Chang Liu, Hwee Tou Ng, and Min-Yen Kan. 2012. Combining coherence models and machine translation evaluation metrics for summarization evaluation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Jeju Island, Korea, 8–14 July 2012, pages 1006–1014.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pages 997–1006.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 1157–1168.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory. Toward a functional theory of text organization. *Text*, 8(3):243–281.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Mass.

Tomacz Marciniak and Michael Strube. 2005. Beyond the pipeline: Discrete optimization in NLP. In *Proceedings of the 9th Conference on Computational Natural Language Learning,* Ann Arbor, Mich., USA, 29–30 June 2005, pages 136–145.

Daniel Marcu. 1997. From local to global coherence: A bottom-up approach to text planning. In *Proceedings of the 14th National Conference on Artificial Intelligence,* Providence, R.I., 27–31 July 1997, pages 629–635.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *Proceedings of ARPA Speech and Natural Language Workshop*.

Sebastian Martschat. 2013. Multigraph clustering for unsupervised coreference resolution. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Student Research Workshop,* Sofia, Bulgaria, 5–7 August 2013, pages 81–88.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the European Conference on Information Retrieval,* Rome, Italy, 2–5 April 2007, pages 557–564.

Kathleen R. McKeown, Judith L. Klavans, Vassileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the 16th National Conference on Artificial Intelligence,* Orlando, Flo., 18–22 July 1999, pages 453–460.

Chris Mellish and Robert Dale. 1998. Evaluation in the context of natural language generation. *Computer Speech and Language*, 12(1):349–373.

Mohsen Mesgar and Michael Strube. 2014. Normalized entity graph for computing local coherence. In *Proceedings of TextGraphs-9: Graph-based Methods for Natural Language Processing, Workshop at EMNLP 2014,* Doha, Qatar, 29 October 2014, pages 1–5.

Mohsen Mesgar and Michael Strube. 2015. Graph-based coherence modeling for assessing readability. In *Proceedings of STARSEM 2015: The Fourth Joint Conference on Lexical and Computational Semantics,* Denver, Col., 4–5 June 2015, pages 309–318.

Mohsen Mesgar and Michael Strube. 2016. Lexical coherence graph modeling using word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* San Diego, Cal., 12–17 June 2016, pages 1414–1423.

Mohsen Mesgar and Michael Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* Brussels, Belgium, 31 October – 4 November 2018. TO APPEAR.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing,* Barcelona, Spain, 25–26 July 2004, pages 404–411.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013),* Lake Tahoe, Nevada, 5–8 December 2013, pages 3111–3119.

Eleni Miltsakaki and Karen Kukich. 2000. The role of centering theory's rough-shifts in the teaching and evaluation of writing skills. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics,* Hong Kong, China, 1–8 August 2000, pages 408–415.

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

Marie-Francine Moens. 2008. Using patterns of thematic progression for building a table of contents of a text. *Natural Language Engineering*, 14(2):145–172.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Shinichi Nakagawa. 2004. A farewell to Bonferroni: the problems of low statistical power and publication bias. *Behavioral ecology*, 15(6):1044–1045.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2):103–233.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research.

Mark E.J. Newman. 2010. *Networks: An Introduction*. Oxford University Press, New York, N.Y.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38.

Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo, and Genichiro Kikui. 2010. Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 910–918.

Daraksha Parveen and Michael Strube. 2015. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence,* Buenos Aires, Argentina, 25–31 July 2015, pages 1298–1304.

Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* Lisbon, Portugal, 17–21 September 2015, pages 1949–1954.

Daraksha Parveen, Mohsen Mesgar, and Michael Strube. 2016. Generating coherent summaries of scientific articles using coherence patterns. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* Austin, Texas, 1–5 November 2016, pages 772–783.

Andreas Peldszus and Manfred Stede. 2015. Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* Lisbon, Portugal, 17–21 September 2015, pages 938–948.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 1532–1543.

Thomas V. Perneger. 1998. What's wrong with Bonferroni adjustments. *Bmj*, 316(7139): 1236–1238.

Casper Petersen, Christina Lioma, Jakob Grue Simonsen, and Birger Larsen. 2015. Entropy and graph based modelling of document coherence using discourse entities: An application to IR. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval,* Northampton, Massachusetts, 27–30 September 2015, pages 191–200.

Sarah E. Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer Speech and Language*, 23(1):89–106.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Hawaii, 25–27 October 2008, pages 186–195.

Massimo Poesio and Mijail A. Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation,* Lisbon, Portugal, 26–28 May 2004, pages 663–666.

Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. 2004. Centering: A parametric theory and its instantiations. *Computational Linguistics*, 30(3). 309-363.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 1–40.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation,* Marrakech, Morocco, 26 May – 1 June 2008.

Ellen F. Prince. 1981. Towards a taxonomy of given-new information. In P. Cole, editor, *Radical Pragmatics*, pages 223–255. Academic Press, New York, N.Y.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th International Conference on Computational Linguistics,* Geneva, Switzerland, 23–27 August 2004, pages 1346–1352.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celibi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. 2004a. MEAD – a platform for multidocument multilingual text summarization. In *Proceedings of the 4th International Conference on Language Resources and Evaluation,* Lisbon, Portugal, 26–28 May 2004.

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

Gerard Salton. 1971. *The SMART Retrieval System – Experiments in Automatic Document Processing*. Englewood Cliffs, N.J.: Prentice Hall.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions*, 11(45):2673–2681.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics,* Ann Arbor, Mich., 25–30 June 2005, pages 523–530.

Nino Shervashidze, S. V. N. Vishwanathan, Tobias H. Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS),* Clearwater Beach, Florida, 16–18 April 2009, pages 488–495.

Luo Si and Jamie Callan. 2001. A statistical model for scientific readability. In *Proceedings of the ACM 10th Conference on Information and Knowledge Management,* Atlanta, Georgia, 5–10 November 2001, pages 574–576.

Sidney Siegel and N. John Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York, 2nd edition.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of the 25th International Conference on Computational Linguistics,* Dublin, Ireland, 23–29 August 2014, pages 950–961.

Manfred Stede. 2012. *Discourse Processing*. Morgan & Claypool Publishers.

Sally Stoddard. 1991. *Text and Texture: Patterns of Cohesion*. Ablex, Norwood, N.J.

Nicola Stokes, Joe Carthy, and Alan F. Smeaton. 2004. SeLeCT: A lexical cohesion based news story segmentation system. *AI Communications*, 17(1):3–12.

Michael Strube. 1998. Never look back: An alternative to centering. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics,* Montréal, Québec, Canada, 10–14 August 1998, volume 2, pages 1251–1257.

Michael Strube and Udo Hahn. 1996. Functional centering. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics,* Santa Cruz, Cal., 24–27 June 1996, pages 270–277.

Michael Strube and Udo Hahn. 1999. Functional centering: Grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.

Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence,* Boston, Mass., 16–20 July 2006, pages 1419–1424.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Dat Tien Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Vancouver, Canada, 30 July – 4 August 2017, pages 1320–1330.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 1395–1405.

Xinhao Wang, Keelan Evanini, and Klaus Zechner. 2013. Coherence modeling for the automated assessment of spontaneous spoken responses. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Atlanta, Georgia, 9–14 June 2013, pages 814–819.

Bonnie Webber, Markus Egg, and Valia Kordoni. 2012. Discourse structure and language technology. *Natural Language Engineering*, 18(4):437–490.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J. Weinberger. 2003. Inequalities for the L1 deviation of the empirical distribution. Technical report, HPL–2003–97 (R.1), HP Laboratories, Palo Alto.

Billy T.M. Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 1060–1068.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 233–242.

Xifeng Yan and Jiawei Han. 2002. gSpan: Graph-based substructure pattern mining. In *Proceedings of the International Conference on Data Mining,* Maebashi City, Japan, 9–12 December 2002, pages 721–724.

Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in ESOL learner texts. In *Proceedings of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications,* Montréal, Québec, Canada, 3–8 June 2012, pages 33–43.

Muyu Zhang, Vanessa Wei Feng, Bing Qin, Graeme Hirst, Ting Liu, and Jingwen Huang. 2015. Encoding world knowledge in the evaluation of local coherence. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Denver, Col., 31 May – 5 June 2015, pages 1087–1096.