Cognitive Structures of Content for Controlled Summarization

Ronald Cardenas Acosta

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2024

Abstract

In the current information age, where over 1 Petabyte of data is created every day on the web, demand continues to rise for effective technological tools to aid end-users in consuming information in a timely way. Automatic summarization is the task of consuming a text document –or collection of documents– and presenting the user with a shorter text, the *summary*, that retains the gist of the information consumed. In general, a good summary should present content bits that are relevant –be informative–, non-redundant -be non-repetitive–, organized in a sensical way –be coherent–, and read as a unified thematic whole –be cohesive.

The particular information needs of each user prompted many variations of the summarization task. Among them, extractive summarization consists of extracting spans of text -usually sentences- from the input document(s), concatenating them, and presenting them as the final summary. Traditionally, extractive systems focus their attention on presenting highly informative content, regardless of whether content bits are repeated or presented in an incoherent, non-cohesive manner. How to balance these properties remains an understudied problem, even though the understanding of the trade-offs between them could enable a system to produce text with relevant content that is also more readable to humans.

This thesis argues that extractive summaries can be presented in a non-redundant, cohesive way, and still be informative. We investigate the interaction between these summary properties and develop models that balance their trade-off during document understanding and during summary production. At the core of these models, an algorithm –inspired by psycholinguistic models of memory– simulates how humans keep track of relevant content in short-term memory, and how cohesion and non-redundancy constraints are applied among content bits in memory.

The results are encouraging. When modeling trade-off during document understanding in an unsupervised scenario, we find that our models are able to detect relevant content, reduce redundancy, and significantly improve cohesion in summaries, especially when the input document exhibits high redundancy. Furthermore, we show that this balance can be controlled through specific, interpretable hyper-parameters. In a similar reinforcement learning scenario, we find that informativeness and cohesion can influence each other positively.

Finally, when modeling trade-off during summary extraction, our models are able to better enforce cohesive ties between semantically similar text spans in neighboring sentences. Our approach produces summaries that are perceived by humans as more cohesive and as informative as summaries only built for informativeness. Catering to the need to process extremely long and redundant input, we design this system to be capable of consuming sequences of text of arbitrary length and test it on scenarios with single, long documents, and multi-documents.

Acknowledgements

I would like to express my deep gratitute to my supervisor, Shay B. Cohen, for his guidance and continuous support throughout this journey, and to my second supervised, Matthias Galle, for the endless and fruitful discussions. To both, for their incredible patience and encouragement during hard times, their wealth of knowledge, and their admirable attitute towards science and life. Their example has and will continue nurturing me as a scientist and as a person. I extend my gratitute to Mark Steedman for his valuable feedback during annual reviews, and to this thesis examiners, Ivan Titov and Arman Cohan, whose insight and comments improved this thesis significantly.

I was fortunate to share paths with many talented colleagues at Shay's group: Zheng Zhao, Marcio Fonseca, Nickl Maveli, Yftah Ziser, Shun Shao, Yifu Qiu. Their brilliance was a constant source of inspiration. I am also grateful to colleagues and friends at the CDT in NLP programme: Tom, Laurie, Nikita, Dan, Parag, Nina, Henry, Faheem, Nicole, Rimvydas, Jie, Emelie, and ohers in posterior cohorts. Additionally, from the School of Informatics: Nikos, Amir, Mohammad, Peter, Ondrej, Lorena, Laura, Sharon, Hao, and many more. I will miss our chats during lunch dearly. Finally, I am extremely grateful to my officemates, Georgia, Rohit, Jason, Lucy, and Irene, for contributing to an amazing office environment.

I would like to extend my thanks to my colleagues at NAVER Labs, German Kruszewski, Jos Rozen, Marc Dymetman, Laurent Besacier, Nadia Chirkova, Agnes Sandor, Nikolao Lagos, and Vadim Tschernezki, for the always stimulating formal and informal discussions. To my collaborators BingSheng Yao, Dakuo Wang, and Yufang Hou, for their guidance and insights.

I am grateful to those close friends who helped me maintain a stable lifestyle and enjoy life after a long day of work: Zihao (a man of thousand talents), Grace, Lewis, Kyohei, Arturo, Traiko, Alvile, Saule, Nikos, Amir, and Yolanda. Also, to the folks at the HCA deparment, for reminding me just how vast (and ancient) human knowledge is.

Special thanks to my parents for their unwavering support and for engraving into me the meaning of commitment thourgh their sacrifice and courage. And to Beatrice, for her dedication, patience, and love. By your side, it is always sunny and warm, even in the darkest corners of this wide planet.

Lastly, I would like to extend special thanks to David Machuca Chocano, for bedazzling a young mind with the wonders of the languages of the world and the human hearts shaping them. For instilling discipline and courage to pursue my dreams, however far-fetched they seemed. This achievement is dedicated to him; may he rest in peace among the literature giants of yore.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Ronald Cardenas Acosta*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The contemporary era is witnessing an unprecedented surge in the generation and consumption of data globally, with a staggering 97 zettabytes produced in 2022 alone. This exponential trend is anticipated to persist, projecting a daily generation of 463 exabytes by 2025, of which 80% is expected to constitute unstructured data, as reported by Statista and the International Data Corporation (Taylor, 2023). This monumental influx of data has precipitated a state of *information overload* (Gross, 1964) in the digital era, wherein individuals tasked with decision-making find themselves flooded with excessive amounts of information, often resulting in suboptimal or uninformed decisions.

In response to this pervasive challenge, there is a compelling need for technologies dedicated to assisting both human users and software systems in comprehending vast quantities of data. In this context, *automatic text summarization* emerges as a critical machine technology. Developed with the specific aim of ingesting extensive textual information, automatic summarization systems are designed to present end-users with a shorter text, the *summary*, containing only the most important information and tailored to their distinct informational needs. Notably, the inception of automatic summarization was motivated by the imperative to alleviate information overload, particularly in domains where the synthesis of ever-increasing amounts of content demanded considerable human effort (Luhn, 1958). This was especially crucial in technical domains where expertise was requisite and qualified manpower was limited.

Despite the commendable intent behind automatic summarization, its implementation has encountered persistent challenges. These challenges include the escalating volume of information to be processed, the identification of relevant and informative content aligned with user needs and background knowledge, and the discernment and handling of redundant information within the input text. Furthermore, despite the remarkable advances in generative Artificial Intelligence in recent years, summarization systems continue to struggle with presenting text in a coherent and cohesive manner.

This chapter delves into the historical evolution of automatic summarization, exploring prior research efforts and methodologies. Then, we examine the role played by text properties, such as redundancy and cohesion, in the construction of summaries, and elaborate on why it is paramount to control these properties for enhancing the utility and usability of summarization systems. Finally, this chapter concludes by laying out the thesis statement in detail, as well as outlining of the entire thesis.

1.1 Overview of Automatic Text Summarization

Jones (1999) defines the task of automatic text summarization, in general terms, as

> *a reductive transformation of source text to summary text through content reduction by selection and/or generalisation on what is important in the source.*

Such a complex task requires deep understanding of the concepts and information in the source text, a criteria to select and condense only the content of most interest, and finally write down a fluent and coherent summary text. Formally, Jones (1999) sets up the summarization task as a process of three steps: (i) document understanding, (ii) content selection, and (iii) production of the summary based on selected content. The process is then open to design choices at each step, which encourages many variations of the summarization task. In general, task variations can be divided according to the following criteria.

- *Single vs Multi-document.* In single-document summarization, the system will consume one document, whereas in a multi-document, it will consume many documents linked by the same topic, e.g. news articles talking about the same event.

- *Generic vs Query-based.* The summarization task is inherently a subjective one, since different users will deem different content as relevant to them. As such, generic summarization aims to circumvent this subjectivity by producing a one-size-fits-all summary aimed at a broad audience, relying instead on the intrinsic summary-worthiness of content in the input. On the contrary, query-focused summarization tailors the selected content to match a query the user provides. Hence, this criteria is dealt mainly during step (ii) of the process.

- *Extractive vs Abstractive.* This criterion deals with the strategy followed to produce a text summary– step (iii) of the process. On the one hand, extractive systems retrieve text spans -usually sentences- from the input, concatenate them, and present the joint text as the summary. On the other hand, abstractive systems generate the summary from scratch, conditioned on the input, oftentimes by paraphrasing and generating novel tokens.

In this thesis, we develop extractive summarization systems that produce generic summaries of long, single documents, taking special care of summary qualities like informativeness, redundancy, and cohesion. Before discussing the specific role and benefits of each of these qualities, we first provide a brief account of how summarization has been tackled in previous work.

Early Research. The earliest work in summarization –applied almost exclusively to English texts– was extractive , and modeled the summary worthiness of sentences through the frequency of their composing words. Luhn (1958) hypothesized that a word of high "resolving power" should be in the middle range of word frequencies in a document, designed to summarize highly technical text. Later, Edmundson (1969) proposed modeling the importance of sentences as the linear combination of four features: frequency of its words, position of sentence in the document, words appearing in article title or section headings, and frequency of specific words in a curated list. Following work extended this feature set to include sentence length and presence of upper-case words (Kupiec et al., 1995), inverse document frequency of words, and occurrence of named-entities (Aone et al., 1997).

With the development of more robust machine learning techniques, summarization was posed as a classification task to decide whether a sentence should be selected or not, with promising results reported using naive-Bayes (Kupiec et al., 1995), decision trees (Lin, 1999), maximum entropy classifier (Osborne, 2002), maximum likelihood estimate based on counts (Nenkova and Vanderwende, 2005), among others. Another early line of research developed summarizers which first identified the most important topics in a document –where a topic was modeled as a group of semantically related words– and then extracted sentences that maximized the coverage of those topics. Methods such as latent semantic analysis (Landauer et al., 1998) and lexical chains (Morris and Hirst, 1991; Barzilay and Elhadad, 1997) were successfully applied to summarization.

Graph-based Methods. The rapid development of the world wide web prompted the development of techniques to identify and rank relevant text content in vast collections of documents. Notably, the PageRank algorithm (Page, 1998) modeled the relative relevance of each element in an hyperlinked set of documents. The network was modeled as a directed graph where each node represents a website and edges represent a website contained an hyperlink to another. Then, the algorithm approximates the eigenvector centrality of each node in the network using an iterative strategy guaranteed to converge. The resulting centrality score of each node can be interpreted as the probability to reach a website. The algorithm was effectively ported to the unsupervised summarization scenario, in which a document or collection of documents were represented as a graph of content units. The TextRank algorithm (Mihalcea and Tarau, 2004) models a document(s) as an undirected graph of sentences where edge weights quantify lexical overlap between sentences. Then, the PageRank algorithm is applied and the score obtained is used as a proxy for the relevancy of a sentence. Concurrently, a similar system, LexRank (Erkan and Radev, 2004), modeled sentence similarity with TF-IDF followed by eigenvector centrality, reporting similar performance to TextRank. More recently, PacSum (Zheng and Lapata, 2019) proposed training a dedicated edge-weight model that learned to score the edge between two sentences, and then apply the PageRank algorithm to obtain sentence centrality scores. The model employs a transformer-based model (Devlin et al., 2019) to encode sentence pairs and produce an edge weight. Critically, all these approaches operate under the assumption that the whole graph of content units is available at each iteration of the algorithm. As we will elaborate in Chapter 3 and 4, the summarization systems proposed in this thesis maintain a data structure connecting a limited number of content units. This structure is incrementally updated so as to simulate the content of human working memory at a particular moment of reading.

Discourse-based Methods. Prior research was directed at modeling the discourse structure of the input text (a stage corresponding to step (i), document understanding). These efforts involved selecting a part of said structure (corresponding to step (ii), content selection), and subsequently building a summary from it, either by concatenating the corresponding text spans or by generating text conditioned upon this sub-structure.

The Rhetorical Structure Theory (RST; Mann and Thompson 1988) prominently featured in this context, wherein it was used to represent the discourse of a document as a directed tree. In an RST tree, the leaves represent text spans functioning as elementary discourse units (EDUs), while non-terminal nodes denoted broader spans covering EDUs related by specific discourse relations. Crucial to the detection of relevant information, RST assigns EDUs the status of nucleus or satellite; a nucleus being deemed the focal point of the text at a given point in reading. Early summarization systems, predominantly of extractive nature, exploited this distinction in status. Techniques involved either rewarding nuclei or penalizing satellites (Ono et al., 1994; Marcu, 1998) to facilitate the extraction of the most salient text spans. This approach proved to be effective for content selection in single-document summarization of news and scientific articles (Marcu, 2000).

Later, Abstract Meaning Representation (AMR; Banarescu et al. 2013) was employed to represent a document as a directed graph of concepts with edges labeled according to semantic roles. Similarly to RST-based summarizers, content selection is done by selecting a sub-graph representing the most relevant concepts in the global graph. Although effective for single and multi-document summarization (Liao et al., 2018; Mishra and Gayen, 2018), the approach relies heavily on high-quality treebanks, which poses a challenge for knowledge domains other than news as well as for other languages.

More recent lines of work combined these discourse structures with the representation power of neural networks. Notably, DiscoBERT (Xu et al., 2020) employs graph neural networks (Scarselli et al., 2009) to encode BERT representations (Devlin et al., 2019) of text spans in an RST tree and a co-reference graph. In this thesis, we focus on the summarization of long documents with highly technical terms. This setup poses a challenge to models like DiscoBERT given that the NLP tools (co-reference engine and an RST parser) have limited accuracy in domains other than newswire (for which they were trained) and because these tools cannot process long documents. We take into account these limitations and propose models that gradually (throughout the course of this thesis) alleviate the dependency on external NLP tools. Hence, the models proposed are specially designed to consume long documents and perform empirically well on highly technical domains.

Psycholinguistic Methods. In psycholinguistics, summarization as a task is often used as a method to investigate cognitive processes involved in text comprehension and production (Kintsch and van Dijk, 1978; Kintsch, 1990; Lehto, 1996; Kintsch and Walter Kintsch, 1998; Ushiro et al., 2013; Spirgel and Delaney, 2016). Such processes are in charge of generalizing, synthesizing, and coherently organizing content units. Comprehension, in turn, is modeled after psycholinguistic models of human reading comprehension (Kintsch and van Dijk, 1978; Kintsch, 1988) which provide a rich and robust theoretical foundation on how content units are discretized and manipulated by cognitive processes. For this reason, comprehension models such as the Micro-Macro Structure (KvD; Kintsch and van Dijk 1978) and Construction-Integration theory (CI; Kintsch 1988), have drawn the attention of researchers in automatic summarization in recent years (Fang and Teufel, 2014; Zhang et al., 2016; Fang, 2019). These theories outline procedures to discretize content into semantic propositions and build text representations that account for local and global coherence. However, computational implementations proposed so far (Fang and Teufel, 2014; Zhang et al., 2016) show a heavy reliance on NLP tools such as entity extractors and coreference resolution systems, as well as external resources like WordNet (Miller, 1995). These requirements greatly limit their application in highly technical domains such as scientific literature. Additionally, many design choices prevented these systems from exploiting properties of memory structures, modeling retrieval processes, or manipulating information at the right granularity level, e.g. ranking words or sentences instead of semantic propositions.

Neural Networks. More recently, summarization approaches rely instead on neural networks to obtain deep representations of content units by means of convolutional neural networks (Perez-Beltrachini et al., 2019; Narayan et al., 2019), recurrent neural networks (Narayan et al., 2018a,b; Cheng and Lapata, 2016), Transformers (Song et al., 2019; Dong et al., 2019) and lately by leveraging large pretrained language models (Zheng and Lapata, 2019; Liu and Lapata, 2019; Zhang et al., 2020). Building up on traditional methods, neural summarization models leverage discourse (Clarke and Lapata, 2010; Cohan et al., 2018), topical (Narayan et al., 2019), and graph representations (Bichi et al., 2021; Qiu and Cohen, 2022). Even though most research concentrates on summarization of middle-sized documents like news articles and Reddit posts (Völske et al., 2017), recent work has shifted attention to long document summarization and its challenges (Cohan et al., 2018; Sharma et al., 2019; Xiao and Carenini, 2019; Fonseca et al., 2022).

Among recent efforts, it is worth mentioning architectures tailored to consume longer inputs by reducing the time complexity of the attention mechanism (Beltagy et al., 2020; Wang et al., 2020; Huang et al., 2021) or leveraging the structure of the input document (Cohan et al., 2019; Narayan et al., 2020). The present work follows this line of research by introducing summarization systems capable of consuming long documents or multiple documents.

Large Language Models. In the last couple of years, the NLP field has witnessed the rise of massive text-generating models trained over the language modeling task, dubbed *large language models*, LLMs (Radford et al.; Touvron et al., 2023a; Jiang et al., 2023). These models are further finetuned to follow instructions and to be aligned to human preferences, resulting in highly capable general-purpose, task-agnostic assistants, including of course the summarization task. Given a text prompt with a query and optionally examples on how to solve the task, these models —most of them consisting of decoder-only architectures (Touvron et al., 2023a)– take the prompt as a prefix to continue generating text, in which the solution to the task will be included. These capabilities, often referred to as *reasoning*, seem to emerge as models are scaled in the number of parameters and training data size (Wei et al., 2022a; Hoffmann et al., 2022). However, answering a query successfully when the input includes a long text remains challenging. For instance, LLMs find it difficult to find relevant information that is located in the middle of a long prompt compared to relevant content at the beginning or closer to the end of the prompt (Liu et al., 2024). In this long context scenario, recent work has sought inspiration from comprehension processes as depicted by psycholinguistics, implemented in how the prompt is constructed and how generation is steered. For instance, Chain of Thought (Wei et al., 2022b) first generates a step-by-step explanation of how the solution is achieved before generating the answer to the query. It is important to note that, even though the final answer might be correct, the reasoning chain generated is not guaranteed to be correct or even faithful. Techniques such as ScratchPad (Nye et al., 2022) and Self-Notes (Lanchantin et al., 2024) instead allow the model to pause answer generation to generate partial reasoning spans, similarly to rehearsal in working memory (Goldstein, 2015). Going beyond a single LLM module, Lee et al. (2024) introduced an agent that consumes the input in one chunk at a time, generates a short summary of it, and stores it in a memory module. When asked to answer a query, the agent processes the intermediate summaries and further re-processes the relevant chunks for more details. The work in this thesis follows this line of work by introducing computational implementations of cognitive processes that simulate how content is organized in human memory.

Lately, a persistent debate has emerged concerning the true capabilities of large language models (LLMs) for summarization. One perspective posits that the summarization task is mostly solved, supported by evidence that LLMs generate highly coherent and near-faithful summaries (Pu et al., 2023). Conversely, concurrent work suggests that LLMs manifest signs of being learning shortcuts instead of truly generalizing (Du et al., 2023; Bihani and Rayz, 2024). Notably, these systems exhibit a degradation of output quality at long input regimes ($>$ 10k tokens), particularly evident in coherence quality (Chang et al., 2024). Regardless, consensus is reached on the limitation of current evaluation methodologies, making it difficult to draw conclusive insights. In this regard, recent efforts have followed suit on the comprehensive evaluation of LLM output not only on the summarization task but also on a plethora of tasks requiring different types of reasoning (Zellers et al., 2019; Srivastava et al., 2023; Gao et al., 2023; Liang et al., 2023).

Extractive vs Abstractive. As mentioned earlier in this section, extractive summarization systems extract content units (usually sentences) from the input document(s) and present them concatenated as the final summary. In contrast, abstractive summarization systems generate a novel text, usually token by token.

Despite significant advances in generative capabilities in recent years, there are several reasons why extractive systems are still attractive to the community. First, extractive summaries consist of grammatically correct and fluent text chunks, hence ensuring readability within these chunks. Moreover, extractive summaries are less prone to exhibit hallucinations compared to abstractive ones (Zhang et al., 2023), given that an extractive summary presents information from the source verbatim, particularly in highly technical domains. However, the complete summary might result incoherent and exhibit high content redundancy if these aspects are not accounted for.

Second, in scenarios where the writing style of the summary is required to match that of the input, extractive systems are guaranteed to produce a text with the same style. This aspect of modeling is becoming more prominent with recent large language models, which are trained over a plethora of styles and domains. Recent work showed that LLMs still struggle to replicate the writing style in the input or adhere to specifically requested styles (Cardenas et al., 2023).

Finally, extractive systems can be more computationally efficient than abstractive systems. Regardless of the architecture, most current generation techniques are primarily autoregressive, i.e. the summary is generated token by token, a crucial bottleneck in long-text generation setups. Furthermore, long inputs pose another bottleneck for decoder-only architectures such as LLMs, where each generation step requires the processing of representations of every token in the input. Despite recent efforts in addressing these bottlenecks (Cohan et al., 2018; Zaheer et al., 2020), long input processing and long text generation remain an open problem. In contrast, extractive systems focus on the representation of content which, upon encoded and grouped into the desired candidate chunks, is promptly selected.

In this thesis, we consider summarization scenarios where the input text is dense in technical content, in which improving the readability, grammatically, and factuality of the summaries is desirable. Moreover, we focus on the scenario where the produced summary preserves the writing style of the input document(s). For these reasons, only extractive systems are developed with a special focus on mechanisms to further control summary qualities.

1.2 The Role of Text Qualities in a Summary

In this section, we present a brief definition of the summary qualities we focus on in this thesis and elaborate on their roles and impact on the communication objective of a summary. We start the discussion with informativeness and how it differs from relevancy, then move to redundancy and repetition. Finally, the distinction and importance of coherence and cohesion are highlighted.

1.2.1 Informativeness

According to Jones 1999, a summary is deemed *informative* when it effectively covers the content within a source document. Informativeness, therefore, is inherently linked to coverage. Later, Peyrard (2019) defined *relevance* as the similarity between content distributions between the summary and the source document; informativeness, instead, is defined as the amount of new information relative to the background knowledge of the user. In the context of general summarization, user background knowledge is assumed to align with the background knowledge of the intended target audience, and in some scenarios (e.g. scientific articles) it is explicitly elaborated in the document. In such cases, relevance and informativeness become equivalent and thus we treat them interchangeably in this thesis.

Given the definitions above, the role of informativeness as a text property is to preserve relevant information from the source. Systems prioritizing informativeness produce summaries that function as information-rich substitutes for the documents or as previews aiding in the decision of whether to read the document in full (Jones, 1999).

1.2.2 Redundancy and Repetition

It is important to note the distinction between repetition and redundancy. Repetition, also known as *linguistic* or *grammatical* redundancy, refers to a linguistic mechanism that serves functional roles in natural language. Tauste (1995) notes its significance in dialogue, where repetition acts as a mechanism for comprehension checks, emphasis, reformulations, and readjustments. Moreover, Walker (1993) argued that the occurrence of repetition in natural language is attributed to human memory limitations and serves as a device to form cohesive ties, either through direct repetition or synonymous expressions. In a public speech scenario (e.g. giving a talk), Johnstone (1994) identified specific functions of repetition such as dealing with interruptions or maintaining the floor while thinking of something to say. Similarly, guides on technical writing portray repetition as a mechanism to complement complex concepts and hence, improve comprehension (Knuth et al., 1989).

On the other hand, *content* or *informational* redundancy, often called simply *redundancy*, refers to the seemingly purposeless use of repetition mechanisms. From a communicative point of view, redundant information does not add any new content unit to the discourse and instead harms readability and conciseness (Walker, 1993).

Addressing redundancy is a challenging task in summarization, requiring the consideration of semantic equivalence at various levels of granularity (Nenkova et al., 2007), with previous work aiming at minimizing redundancy while maximizing coverage (Carbonell and Goldstein, 1998a; Yogatama et al., 2015).

1.2.3 Coherence

The discussion of the role of coherence in a text calls for an initial clarification of the terms discourse and text. Discourse denotes the process of conceptual formulation where we use our linguistic resources (i.e. natural language) to make sense of reality. Text, on the other hand, is the linguistic product of a discourse process, serving as evidence of a pragmatic process of a communicative interaction (Bublitz et al., 1999). Despite this distinction, linguistic literature uses both terms interchangeably.

In this context, coherence is defined as the dynamic process of textual interpretation, involving the mapping of linguistic units within a text onto concepts in the readers' mental representation of content, establishing connections therein. Interestingly, coherence is not a text-inherent property but rather a process by which "meaning is read into a text" (Bublitz et al., 1999) within a socio-cultural context and subject to the readers' background knowledge. As such, coherence is dependent on interpretation.

From a computational linguistic point of view, a text is coherent if its discourse structure can be successfully reconstructed. Theoretical frameworks of discourse differ on the nature of the structure itself and the scope it covers. For instance, Rhetorical Structure Theory (Mann and Thompson, 1988) or Discourse Theory Representation (Kamp and Reyle, 1993) model the discourse of the entire text, aiming to capture *global* coherence. In contrast, other theories like Centering theory (Grosz et al., 1995), focus on the structure present in nearby sentences, referred to as *local* coherence. In the context of summarization and other multi-sentence generation tasks, coherence reflects how content is organized at a global level. This discourse organization depends on the writing style, the target audience, and the purpose of the text (Jones, 1999).

1.2.4 Cohesion

Cohesion is the property of a text that allows it to function as a unified whole, serving as a mechanism through which the texture of a text is expressed (Hassan et al., 1976). This mechanism ensures smooth transitions between semantic topics within a text through the use of thematic links known as *cohesive ties*, which are established between clauses in nearby sentences. These cohesive ties are explicitly indicated by grammatical constructions or linguistic units. Contrary to coherence, cohesion is invariant and independent of interpretation and instead relies on the explicit textualization of contextual connections. As such, cohesion does not model the discourse structure of a text but instead can be considered a device for achieving local coherence.

The role of cohesion in reading comprehension, the process of constructing a mental representation of content, has been extensively studied in psycholinguistics. Kintsch (1990) observed that the absence of cohesive ties in a text leads humans to perform *inference* –a cognitive process where prior knowledge is used to establish connections and make sense of a text. This cognitive demand during reading was found to negatively impact the cohesion of summaries written immediately after reading the text, a finding supported by subsequent work (Lehto, 1996; Ushiro et al., 2013; Spirgel and Delaney, 2016).

In the context of summarization, cohesion plays a crucial role in enhancing the comprehension of a summary. From an evaluation point of view, cohesion facilitates the proper evaluation of text properties specific to the summarization task. The production of a cohesive summary is the first step in ensuring that the content is comprehended before being judged for another property, e.g. relevance or redundancy. Barzilay and Elhadad (2002) demonstrated that humans prefer cohesive orderings of sentences in extractive summaries among many permutations, with cohesion positively impacting text comprehension. Moreover, cohesion proves beneficial in conditional text generation. Krishna et al. (2021) found that factuality and fluency improved when generation was conditioned on contiguous chunks of sentences rather than randomly selected, concatenated sentences. Zhang et al. (2020) acknowledged the importance of masking a contiguous chunk of text instead of masking discontinuous sentences during pretraining of a neural network was crucial for downstream summarization tasks.

1.3 Thesis Statement

In this thesis, we aim to develop extractive summarization systems equipped with **mechanisms to control text properties of the produced summary**, for cases in which the input text exhibits high content redundancy. The proposed control mechanisms are inspired by cognitive processes in charge of organizing content in human memory according to the Micro-Macro theory (Kintsch and van Dijk, 1978). Notably, these cognitive processes and the associated memory structures they operate on were modeled as general purpose, applicable to any task involving comprehension (reading) or production (written or spoken). As such, the devised mechanisms in this thesis do not require explicit task supervision and although in theory are task-agnostic, we address only the task of summarization. We implement these mechanisms at different stages of the summarization pipeline, namely during document understanding and during summary production, and investigate their effect on summary informativeness, redundancy, and cohesion.

First, we focus on controlling content selection in an unsupervised way during document understanding. At the core of our summarization system, an algorithm updates and reinforces relevant content in working memory while consuming the document sentence by sentence iteratively. Working memory is modeled as a tree of semantic propositions that is capped in size to emulate constraints in human memory (Baddeley, 2018) and is updated with fresh information in each iteration. Intuitively, propositions retained in working memory for more iterations are more central to the argumentation of the text and hence more relevant. Our key insight is that incorporating KvD-grounded intuitions into the estimation of the summary-worthiness of a proposition results in a sufficiently strong signal to extract highly relevant summaries in an unsupervised way. Furthermore, our system is capable of controlling the level of generality or technicality of the extracted content by manipulating the working memory capacity.

Next, we focus on the trade-offs summarization systems incur on when aiming to control redundancy and cohesion in summaries during document understanding, and their impact on informativeness. Two optimization scenarios are investigated: (i) when the summary property is modeled through proxies in an unsupervised setup, and (ii) when a specific summary property is optimized in a reinforcement learning setup. In the unsupervised setup, we introduce novel computational implementations of the KvD theory that explicitly model relevancy, non-redundancy, and cohesion among propositions in working memory. Similar to our previous content selection system, relevance is modeled by pruning the working memory structure down to a fixed number of units, keeping only the most relevant units read so far. Cohesion is modeled by ensuring that working memory consists of a connected graph at each iteration, in which two proposition nodes are connected if they present lexical overlap or are related by a grammatical function. Finally, redundancy is controlled by discarding redundant units from memory. When tested on single-document unsupervised summarization of scientific articles, our results show that our KvD systems manage to extract highly cohesive summaries across increasing levels of document redundancy. Notably, tailored human evaluations comparing our systems with strong unsupervised baselines indicate that KvD summaries were more informative and perceived as more cohesive. These results highlight the benefit of modeling these two properties concurrently. In the reinforcement learning setup, we compare systems that aim to balance informativeness and redundancy, against those which balance informativeness and local coherence. We model this trade-off as a linear combination of property-specific rewards, where the informativeness reward encourages high lexical overlap with a reference summary and a cohesion reward encourages more sensible continuations between adjacent sentences of the summary. Notably, the cohesion reward consists of a classifier trained to distinguish between shuffled from unshuffled text, and acts as a holistic quantifier of the preferred order a cohesive text should have. Extensive automatic –both quantitative and qualitative– evaluation revealed that systems optimizing for cohesion are better at organizing content in the produced summaries, compared to systems only optimizing for informativeness or redundancy. Moreover, cohesion-optimized models are able to obtain comparable –if not better– informativeness and coverage levels.

Finally, we focus on control summary properties with mechanisms implemented at different stages of the summarization pipeline. The first mechanism aims to control redundancy during input understanding, and the second one aims to balance informativeness and cohesion during summary extraction. On the one hand, summary redundancy is addressed by controlling the redundancy levels of the input text, following previous findings (Carbonell and Goldstein, 1998b; Xiao and Carenini, 2020). Our pipeline consumes input text in a cascaded way: first splitting the input into contiguous passages, then consuming passages one at a time so as to minimize their semantic similarity with already selected passages. On the other hand, informativeness and cohesion are directly modeled during summary extraction. Extraction is done in a sentence-by-sentence fashion, quantifying summary properties independently at each step. Informativeness is quantified by a strong neural model trained to select summary-worthy sentences in a supervised way. Cohesion, instead, is quantified by a sentence selector that incrementally builds cohesive chains of noun phrases and models chain interaction. Once again, the selection algorithm is inspired by KvD cognitive processes organizing content in memory, this time during production. Working memory is modeled as a limited-capacity buffer of lexical chains, forcing the model to keep only the most salient chains and send the rest to long-term memory. Then, cohesion is quantified based on the strength of semantic similarity between incoming units and units in active chains, encouraging connections to chains in working memory over chains in long term memory. Note that this selector does not need supervision for quantifying cohesion, in contrast to the informativeness quantifier.

We test our methodology on newswire multi-document summarization and single-long document summarization of scientific articles, patents, and government reports. Across domains, extensive experiments show that, first, our system is effective at incrementally building an input sequence with lower content redundancy, which translated to a significant reduction in summary redundancy. Second, the proposed sentence selector managed to maintain summaries informative while improving cohesion significantly, connecting more noun phrases through cohesive ties compared to a greedy selector. Additionally, tailored human evaluation campaigns revealed that cohesion has a positive impact on perceived informativeness, and that our extracted summaries exhibit chains covering adjacent or near-adjacent sentences. Closer inspection showed that topics flow smoothly across extracted summaries with no abrupt change or jumps.

In summary, the main contributions of this thesis are the following.

- We propose a mechanism to control content selection in an unsupervised way, inspired by processes and structures in human memory according to the KvD theory.
- We introduce two novel computational implementations of the KvD theory operating during document understanding, tailored to the extractive summarization task. The systems implement mechanisms to balance informativeness, cohesion, and redundancy in the produced summaries.
- We propose a summarization system equipped with a sentence selector specialized in modeling cohesion by simulation the KvD theory during production, as well as mechanisms to balance the trade-off between informativeness, cohesion, and redundancy.

1.4 Thesis Outline

This thesis is organized as follows:

- Chapter 2 presents background knowledge regarding the linguistic properties considered such as content redundancy, cohesion, and coherence. In addition, we elaborate in detail about the psycholinguistic theory, KvD, and the cognitive structures to be simulated throughout our experiments.
- Chapter 3 discusses our work on leveraging simulated cognitive structures for content selection during document understanding.
- Chapter 4 expands on our approaches to balance redundancy and cohesion during document understanding, and their impact on informativeness, under unsupervised and reward-guided scenarios.
- In Chapter 5, we introduce our control methodology aimed at balancing informativeness and cohesiveness, this time during sentence selection. Experiments showed that our approach was effective in single and multi-document summarization.
- Chapter 6 summarises our main findings, discusses the limitations of our approaches and elaborates on future research directions.

Chapter 2

Background

The research presented in this thesis spans concepts in the areas of psycho-linguists, linguistics, and computer science. Hence, a detailed definition of them and how they relate to the summarization task is provided in this chapter, organized as follows. We start by elaborating on the challenges reported in the literature concerning the modeling –and even measuring– of summary properties. Then, we describe in detail the Micro-Macro theory of reading comprehension and make a case for its appropriateness to the summarization task, pointing out key properties of the cognitive structures built during processing and how they can be exploited to model the summary properties we are concerned about in this thesis. Finally, standard metrics for measuring summary properties are explained and discussed.

2.1 Challenges in Controlling Summary Properties

2.1.1 Informativeness and Content Coverage

Previous research has sought to improve informativeness within a reinforcement learning framework, using ROUGE scores as reward signals. This approach has been investigated both in extractive (Narayan et al., 2018b; Zhou et al., 2018; Gu et al., 2022) and abstractive summarization (Dong et al., 2018). However, relying solely on such reward signals can result in deterioration of generation quality, leading to repetitive and potentially incoherent output. To address this issue, previous work explored downweighting the reward loss during training (Dong et al., 2018) or mixing it with reference-free rewards such as saliency (Pasunuru and Bansal, 2018).

In the context of coverage control, previous work aimed at maximizing the coverage of relevant topics within a document(s) by incorporating an explicit step for content selection or planning. The challenge inherent in controlling coverage can be seen from two fronts, content unit detection and summary length control. Firstly, while it may be more straightforward to operate over text spans such as sentences (Kedzie et al., 2018; Fonseca et al., 2022) or semi-structured input such as tables (Puduppully et al., 2019; Wiseman et al., 2017), controlling coverage of entities proved to be challenging and domain restrictive (Narayan et al., 2022). Secondly, previous work has focused on imposing constraints on summary length in order to control coverage in a more controlled evaluation setup (Chan et al., 2021; Fonseca et al., 2022), an approach we build upon in Chapter 3.

2.1.2 Redundancy

Controlling for content redundancy in a summary faces several challenges, including: (i) determining the granularity level at which content is to be compared, (ii) establishing a threshold for semantic overlap to ascertain when two concepts are the same, and (iii) distinguishing whether the repetition of a concept is the product of a cohesive connection or, lacking any functional purpose, is merely redundant.

Regarding the first point, prior work on semi-automatic evaluation (Nenkova et al., 2007; Zhang and Bansal, 2021) provided a comprehensive framework for extracting and comparing content units at different levels of granularity, although at the expense of great manual effort. Concerning the consideration of semantic overlap, approaches in extrative summarization have integrated information about the already selected units at the lexical level (Carbonell and Goldstein, 1998a; Paulus et al., 2018) and at the embedding level (Xiao and Carenini, 2020; Gu et al., 2022), with positive results.

However, it is important to note that these and related approaches (Zhou et al., 2018; Fabbri et al., 2019) aim to minimize repetitiveness and, more broadly, the presence of semantically related units. Whilst minimizing content redundancy is desirable, the minimization of linguistic redundancy can potentially hinder the communication efficacy of a summary. As discussed in Chapter 1, linguistic redundancy has a defined role in the communication pipeline. Hence, a cohesive text is likely to exhibit a non-trivial degree of repetitiveness, which would be compromised if the latter were to be overly minimized. The synergy between repetitiveness and cohesion is acknowledged in Chapter 4, where we propose control mechanisms that do not solely aim to minimize the repetitiveness in final summaries. Instead, these mechanisms aim to strike a balance with other properties, such as informativeness and cohesion.

2.1.3 Coherence and Cohesion

Coherence is a long studied area, with early research modeling local coherence as lexical cohesion (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013). In practice, these systems are limited by data sparsity —the limited lexical matching between nouns and entities– and the performance of coreference resolution models. More recent approaches have addressed these limitations by using neural networks pretrained on massive amounts of text as their basis for semantic similarity comparison (Mesgar and Strube, 2016; Zhao et al., 2023). Nevertheless, most research efforts concentrate on measurement and evaluation (Zhang and Bansal, 2021; Fabbri et al., 2021), aiming to develop metrics that highly correlate with human rankings of coherence. This avenue is extremely challenging due to the lack of community-wide standards and the surprising little consensus on what each summary property should measure (van der Lee et al., 2019).

Regarding the control of the local coherence in summary, previous work is limited. Wu and Hu (2018) aimed to balance informativeness and local coherence in a reinforcement learning setup, reporting heavy trade-off between the two properties. Coherence was modeled by a shuffling scorer similar to the one later analysed by Steen and Markert (2022). Pertaining global coherence, similar work in controlled generation has demonstrated that accounting for planning helps dealing with discourse organization of final summaries (Goldfarb-Tarrant et al., 2020; Sharma et al., 2019; Hua et al., 2021), although such insights come oftentimes from qualitative analyses rather than automatic or human evaluation.

2.2 The KvD Theory of Human Memory

Proposed by Kintsch and van Dijk (1978), the Micro-Macro Structure theory describes the cognitive processes involved in text (or speech) comprehension, and provides a principled way to make predictions about the content human subjects would be able to recall later, for instance, when asked to summarize the text.

However, it is important to note that summarization as a task remains challenging for humans, especially when the reader's background knowledge is insufficient for successful comprehension of all the details in the input text. Kintsch (1990) investigated summarization as a learnable skill developed throughout the educational journey of human subjects, placing into perspective the difficulty and knowledge requirements of the summarization task and natural language understanding in general.

2.2.1 Micro and Macro Level of Comprehension

In the KvD theory, discourse comprehension is performed at two levels, micro and macro-level, and discourse is represented with a characteristic structure of content at each level. At the micro level, content structure is modeled after working memory –a type of short-term memory– and KvD defines precise mechanisms that update and reinforce content in the structure. Content at this level is discretized in basic meaningful units by means of linguistic propositions. A proposition is denoted as $\texttt{predicate(arg}_1\texttt{,arg}_2\texttt{,...)}$ where arg_i is a syntactic argument of the predicate (e.g. argument to a transitive verb). As such, propositions can be interpreted as clauses or short sentences and hence provide more expressivity than words units during comprehension. The advantage of using propositions as content units goes beyond the amount of information it can pack. A proposition can be linked to another either syntactically or semantically, potentially building entire connected structures of propositions. According to KvD theory, working memory holds a cohesive organization of content units by making sure that all units are connected e.g. in a connected tree. Hence, the resulting micro-structure models cohesive ties in the text.

At the macro level, content structure represents the global organization of the text, built in a bottom-up matter starting from micro-propositions. The construction of this instruction is controlled by the *scheme*, the formal representation of the reader's goals, capturing the global discourse of a document. For instance, if the task is summarization, KvD defines macro-processes concerned with generalization, fusion, insertion of details from background knowledge, among others. Similarly to micro-processing, macro-processing is iterative, simulating multiple stages of content generalization, refinement, and planning that humans perform.

In this thesis, we consider only the structures represented at the micro level and leverage them for the task of extractive summarization. Structures and processes at the macro level would require human-like reasoning and intuition and even though recent work on neuro-symbolic systems (Garcez and Lamb, 2020; Bengio, 2017) and common-sense reasoning (Speer et al., 2017; Bosselut et al., 2019) showed a promising development path, we leave this path out of the scope of this thesis and for future work. Nevertheless, summarization systems based solely on micro-processing have been successfully explored (Fang and Teufel, 2014; Zhang et al., 2016) for cases in which the scheme of a document is relatively fixed, e.g. a scientific document is expected to be organized in sections. In such cases, content in micro-propositions provides enough evidence to model summary properties of interest. Moreover, a summarization system simulating micro-processes offers an interpretable model where researchers (or users) can track which concepts are connected or considered semantically similar, and which content units are considered relevant during reading or production. In case of suboptimal results, an explainable model provides troubleshooting information a researcher can gain insights on. For instance, if the input document is poorly written or contains highly technical terms, a model simulating micro-processes provides a transparent account of which units could not be connected, hence helping in deciding how to improve such connections, e.g. by using an external knowledge graph.

The KvD theory provides a principled way to operationalize the manipulation of content units during reading and is precise in many aspects of the simulation, e.g. the nature and properties of memory trees. However, Kintsch and van Dijk (1978) make clear that the theory does not specify details of cognitive processes involving inference, i.e. the KvD theory can tell you when an inference occurs and its end result will look like but the theory cannot tell you how this result is arrived at. Examples include how to construct propositions from text or how many nodes are retrieved by the recall mechanism. Hence, a computational implementation of this theory calls for design choices that allow us to define a tractable model with the NLP tools we have nowadays.

Throughout this thesis, we instantiate this theory by proposing computational implementations of it. First, we show how KvD can be used for document understanding, specifically for obtaining a numerical score for each sentence. In Chapter 3, we investigate ways of tailoring this scoring to better detect relevant content. In Chapter 4, we take a deeper look into how KvD can be used to model relevancy, redundancy, and cohesion during document reading. Finally, we show how KvD can be exploited during sentence selection to model cohesive ties more explicitly. In the rest of this section, we elaborate on how KvD works at the micro-level and provide an example simulating human reading step by step, highlighting properties of memory structures and how they can be exploited for the summarization task.

2.2.2 Memory Simulation at Micro Level

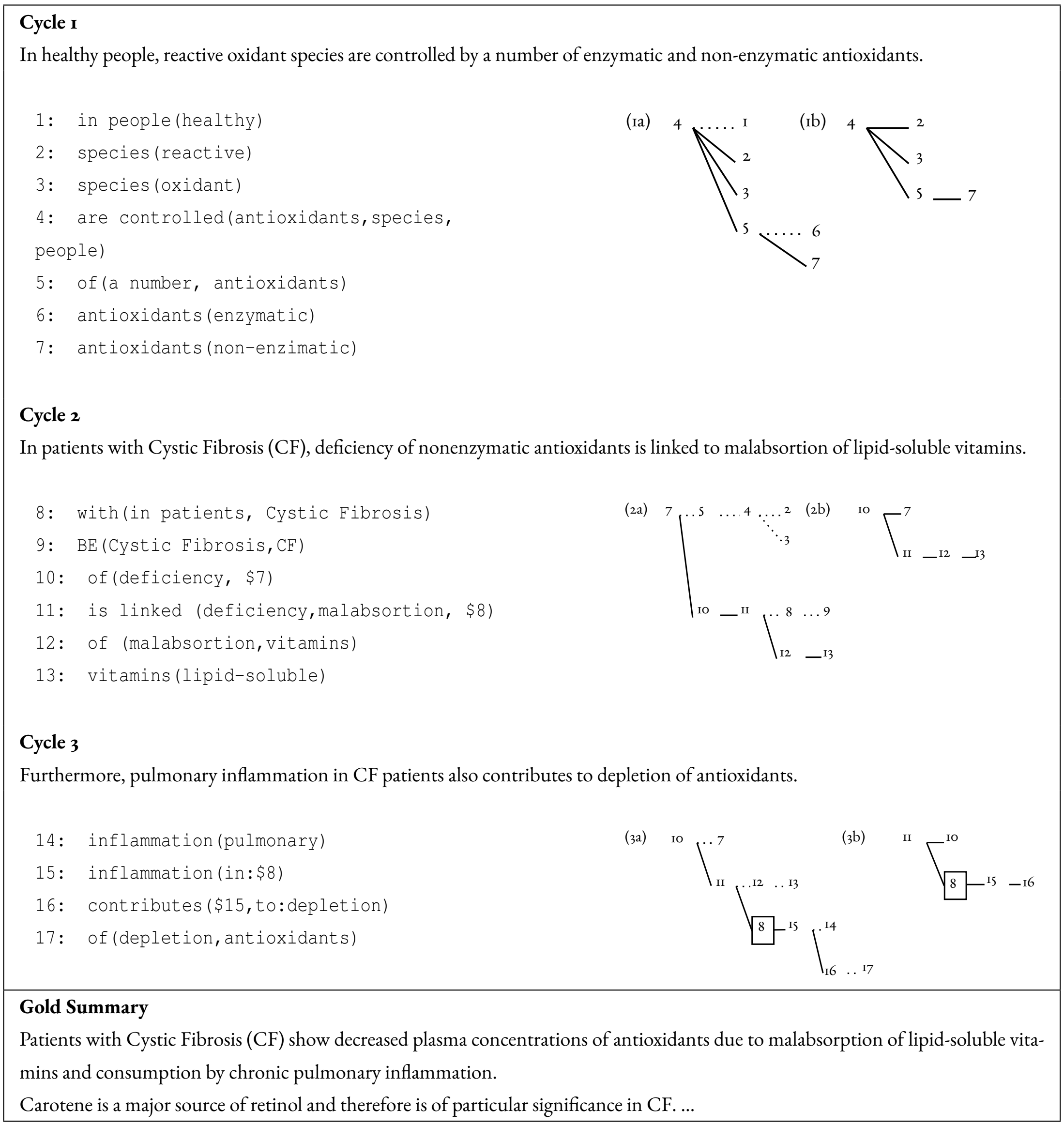
At the micro level, content is organized in a data structure representing working memory called the *memory tree*, where each node corresponds to a proposition and two propositions are connected if any of their arguments overlap.

According to KvD, reading is carried out iteratively in *memory cycles*. In each cycle, only one new sentence is loaded to the working memory, where its propositions are extracted and added to the current memory tree. The limits of memory capacity is modeled as a hard constraint in the number of propositions that will be preserved for the next cycle. Hence, the tree is pruned and some propositions are dropped or *forgotten*. However, if nodes cannot be attached to the tree in upcoming cycles, forgotten nodes can be recalled and added to the tree, serving as linking ideas that preserve the cohesiveness represented in the current tree.[1] Whenever the content in working memory is changed, whether adding propositions or removing them, the root is reassigned to the node containing information central to the argumentation represented in working memory. We now illustrate with an example how content units are captured, forgotten, and recalled during a KvD simulation of reading.

Consider the first three sentences of the introduction section of a biomedical article, along with its abstract, shown in Figure 2.1. At the beginning of cycle 1, propositions 1 to 7 are extracted from the incoming sentence and populate an empty working memory, resulting in tree (1a). Note that the root, node 4, includes the main verb of the sentence and links the main actors (`antioxidants`, `species`, and `people`). Note also that connected propositions present arguments in common, e.g. node 5 and 6 share the argument `antioxidants`. Then, the memory capacity constraint is enforced by pruning nodes until the tree is of a predetermined size. In this example, we set the memory limit to 5 propositions per cycle. KvD introduced the *leading edge* strategy for prunning, which traverses the tree in depth first order starting from the root and selects only the most recent node (in order of reading) at each step. In case a leaf node is reached and there is capacity left, the tree is traversed in breath first order starting from the root and selects nodes with the same criteria, until capacity is reached. In cycle 1, the selected nodes from tree (1a) are 4, 5, 7, 3, and 2, in that order. The remaining nodes, 1 and 6, are pruned. Since content in working memory has been reduced, the root must be reassigned if needed. However, node 4 remains central, hence it remains as root and we move on to the next cycle with tree (1b) as memory tree. These pruned trees constitute the final product of each cycle and will be used for our content selection experiments.

In cycle 2, propositions 8 to 13 are added to memory, tree (1b). In the presence of this new information, the root is reassigned to a proposition central to all the propositions in memory. In this case, node 7 is made root because it presents information common to both sentences (`nonenzimatic antioxidants`), hence being central. Note also that the new tree (2a) showcases clearly two ramifications of the current topic, namely that `$7` '*control a specific kind of molecules*' and '*deficit of* `$7` *causes certain condition*'. Then, we apply the *leading edge* strategy to select nodes 7, 10, 11, 12, and 13, in that order, and prune the rest. Since the content of the working memory has changed again, node 10 is now deemed as central and assigned root status, resulting in tree (2b).

[1] It is worth noting that Kintsch and van Dijk (1978) did not specify how many nodes can be recalled at a single time, however recent implementations (Fang, 2019) limit this number to at most 1.



Cycle 1

In healthy people, reactive oxidant species are controlled by a number of enzymatic and non-enzymatic antioxidants.

```
1:  in people(healthy)
2:  species(reactive)
3:  species(oxidant)
4:  are controlled(antioxidants,species,
people)
5:  of(a number, antioxidants)
6:  antioxidants(enzymatic)
7:  antioxidants(non-enzimatic)
```

Cycle 2

In patients with Cystic Fibrosis (CF), deficiency of nonenzymatic antioxidants is linked to malabsortion of lipid-soluble vitamins.

```
8:  with(in patients, Cystic Fibrosis)
9:  BE(Cystic Fibrosis,CF)
10:  of(deficiency, $7)
11:  is linked (deficiency,malabsortion, $8)
12:  of (malabsortion,vitamins)
13:  vitamins(lipid-soluble)
```

Cycle 3

Furthermore, pulmonary inflammation in CF patients also contributes to depletion of antioxidants.

```
14:  inflammation(pulmonary)
15:  inflammation(in:$8)
16:  contributes($15,to:depletion)
17:  of(depletion,antioxidants)
```

Gold Summary

Patients with Cystic Fibrosis (CF) show decreased plasma concentrations of antioxidants due to malabsorption of lipid-soluble vitamins and consumption by chronic pulmonary inflammation.

Carotene is a major source of retinol and therefore is of particular significance in CF. ...

Figure 2.1: Simulation of KvD reading during three cycles. Each row shows the sentence consumed (top), the propositions extracted (left), and memory trees before (1a, 2a, 3a) and after (1b, 2b, 3b) applying a memory constraint of 5 nodes. Argument `$N` means that proposition `N` is used as argument. Squared nodes are recalled propositions. Solid lines connect nodes selected to keep in memory, and dotted lines connect nodes to be pruned.

In cycle 3, the newly extracted nodes (14 - 17) cannot be attached to the current tree because the linking node, `$8`, was pruned in the previous cycle. Therefore, proposition 8 is *recalled* and re-attached to the tree, showed as a squared node in tree (3a) and (3b). Finally, the selection strategy is applied and node 11 is selected as new root, obtaining (3b).

2.2.3 Reproduction Probability

In each cycle, a proposition can either be selected to stay in the working memory tree or removed from it and sent to long-term memory. At the end of the simulation, a previously removed proposition can still be used in the summary if it was relevant enough.

KvD formalizes this intuition through *reproduction probability*, expressing the probability of a proposition to be reproduced (i.e. written down) when asked to write the summary of a text. Given a proposition p that was retained in working memory for k cycles (not necessarily consecutive), let ρ be the probability of forgetting p in each cycle, i.e. sending it to long-term memory. Then, the reproduction probability of p at the of the simulation is defined as

$$rp_k(p) = 1 - (1-\rho)^k. \tag{2.1}$$

In practice, KvD defined ρ as constant throughout an entire simulation and for all propositions. We elaborate on a generalized version of this expression in Chapter 3, to better account for the influence a proposition had on each cycle it participated on.

2.2.4 Properties Relevant to Summarization

The procedure for content manipulation described in § 2.2.2 imposes constraints on the shape, size, and content of memory trees during simulation. Such constraints bestow memory trees with special properties relevant to the task of summarization, specifically with respect to lexical cohesion, relevancy, and redundancy.

Local Coherence and Cohesion. A memory tree constitutes a connected structure in which two propositions are connected if any two of their arguments refer to the same concept. Connectivity, Kintsch and van Dijk (1978) argued, is a consequence of the text being well-structured and locally coherent, although connectivity is not a necessary condition for coherence –a disconnected structure can still be coherent for a reader. In this way, KvD enforces local coherence in a memory tree in the form of cohesive ties between the propositions in it. For instance, proposition 8 in cycle 3 of Figure 2.1 serves as a bridge to keep the memory tree connected, since propositions talking about *CF patients* (propositions 8 and 9) were discarded in the previous cycle.

This connectivity property has the following implication for cohesion in a final summary. By retaining a set of cohesive content units in working memory, their reproduction probability is increased. Consequently, cohesive groups of propositions will present similar scores at the end of the simulation, encouraging the selection of content that reads more cohesive as a whole.

Relevancy. In addition to being locally coherent, memory micro-structure takes the form of a tree for the following reasons. KvD states that the root of a memory tree should contain information central to the argumentation represented in the working memory; hence, the root is deemed as the most relevant proposition in memory, and the more relevant a proposition is, the closer to the root it will be. This property could be exploited by a summarization system by designing a scoring function that takes the position of a tree node into account. Additionally, tree branches can be seen as ramifications of the current topic, each branch adding more specialized content as it grows deeper.

However, a KvD-based sentence ranking system that relies on proposition scoring would first need to capture the right propositions in working memory. Let us look at the first sentence of the gold summary in Figure 2.1). On the one hand, many propositions (7, 8, 12, 13, and 15) appear verbatim in this sentence, although sometimes only partially (e.g. 7 and 15). The capture of proposition 8 in cycle 3 highlights the importance of the recall mechanism in KvD to bring back relevant information. On the other hand, fine-grained information relevant to the summary might also be lost, such as node 14, in which a crucial property of a noun is not captured (‘*pulmonary*’).

Redundancy. Finally, KvD processes influence redundancy reduction in two accounts. First, propositions in a memory tree are connected such that each proposition adds new details about a concept without encoding more redundant arguments than necessary. For instance, consider again proposition 2 and 3 in Figure 2.1, where both propositions add relevant details (*reactive* and *oxidant*) about a concept (*species*). Hence, memory trees constitute a representation with the maximum amount of relevant details that can fit in working memory whilst minimizing the redundancy of arguments.

Second, in case the recall mechanism needs to be used, KvD retrieves only the minimum amount of propositions to serve as a bridge and connect the incoming propositions. Specifically, the recall mechanism only adds one recall path to the memory tree instead of many other alternative paths. By not loading redundant paths into memory, a system could avoid increasing the score of redundant content and update only one recall path at a time. This behavior, as we will demonstrate later, contributes immensely to decreasing redundancy in the final summary and becomes particularly important for highly redundant documents, e.g. scientific articles that repeat information in several sections.

2.3 Automatic Summary Evaluation

The automatic evaluation of summaries properties –and text properties in general– is an active line of research (Fabbri et al., 2021; Steen and Markert, 2022), where a metric is good if it correlates well with human judgments of that property. However, eliciting human judgment in a controlled and reproducible setup is challenging, with many studies pointing out the lack of consensus on what each summary quality should entail or represent (van der Lee et al., 2019; Gehrmann et al., 2023), e.g. requesting subjects to judge grammaticality as a proxy for coherence. In this section, we give a detailed account of the automatic metrics commonly used in the literature to quantify summary properties.

2.3.1 Informativeness and Content Coverage

ROUGE. Standing for Recall-Oriented Understudy for Gisting Evaluation (Lin, 2004), ROUGE is a framework providing a set of metrics that quantify the amount of content overlap, modeled as n-grams, between summaries produced by summarization systems (system summaries) and reference summaries (ideally written by human experts).

Given a system summary $\hat{S}$ and a set of reference summaries $\mathcal{S}$, the following metrics are defined in terms of recall, precision, and F_β measure.

- ROUGE-N. Measuring n-gram overlap between $\hat{S}$ and reference summaries in $\mathcal{S}$ as

$$\begin{aligned}\text{ROUGE-N}_R &= \frac{\sum_{S\in\mathcal{S}} |u_N(\hat{S}) \cap u_N(S)|}{\sum_{S\in\mathcal{S}} |u_N(S)|}\\ \text{ROUGE-N}_P &= \frac{\sum_{S\in\mathcal{S}} |u_N(\hat{S}) \cap u_N(S)|}{|u_N(\hat{S})|}\\ \text{ROUGE-N}_{F_\beta} &= \frac{(1+\beta^2)\text{ROUGE-N}_R \cdot \text{ROUGE-N}_P}{\text{ROUGE-N}_R + \beta^2\text{ROUGE-N}_P},\end{aligned} \tag{2.2}$$

  where $u_N(S)$ is the set of unique n-grams in S. Even though Lin (2004) introduced ROUGE-N only as a recall measure (ROUGE-N_R), subsequent summarization work reported recall, precision, and F_1 measures in order to coincide with the formulation of ROUGE-L, explained below. When $N = 1$, ROUGE-1 operates over unigrams and it is equivalent to compare bag of words between system and reference summaries.

- ROUGE-L. Measuring the longest common subsequence between each sentence s in reference summary $S \in \mathcal{S}$ and $\hat{S}$, defined as

$$\begin{aligned}
\text{ROUGE-L}_{R} &= \frac{\sum_{s \in S} \text{LCS}_{\cup}(s, \hat{S})}{|S|} \\
\text{ROUGE-L}_{P} &= \frac{\sum_{s \in S} \text{LCS}_{\cup}(s, \hat{S})}{|\hat{S}|} \\
\text{ROUGE-L}_{F_{\beta}} &= \frac{(1+\beta^{2})\text{ROUGE-L}_{R} \cdot \text{ROUGE-L}_{P}}{\text{ROUGE-L}_{R} + \beta^{2}\text{ROUGE-L}_{P}},
\end{aligned} \tag{2.3}$$

where $|S|$ is the length of S in number of tokens and $\text{LCS}_{\cup}(s, \hat{S})$ is the length of the *union* longest common subsequence. This definition of ROUGE-L corresponds to the *summary-level* flavour of the metric introduced in Lin (2004), also denoted ROUGE-LSum in the `rouge-score` library.[2]

- The framework also includes other more elaborated metrics, such as ROUGE-W (a ROUGE-L variant with weighted common subsequences rewarding consecutive span matches), ROUGE-S (considering overlap of skip-grams), and ROUGE-SU (an extension to ROUGE-S that also includes unigram counting).

It is important to note the setup ROUGE was proposed for consisted of a summarization setup where summaries were required to have closely similar lengths, and comparison against more than one reference summary per sample was possible. This is no longer the evaluation setup for modern summarization benchmarks such as the summarization of newswire (Hermann et al., 2015), scientific articles (Cohan et al., 2018), or books (Kryściński et al., 2022). Hence, it can be said that ROUGE is a less reliable metric when comparing a single reference summary and candidate summaries that were generated without any length control. Nevertheless, even though many issues have been identified when using ROUGE outside its proposed setting (Liu and Liu, 2008; Cohan and Goharian, 2016; Schluter, 2017), ROUGE has shown a high correlation with human judgments of relevancy and coverage (ROUGE-1 and ROUGE-2), as well as fluency (ROUGE-L; Graham (2015); ShafieiBavani et al. (2018); Fabbri et al. (2021), in generation setups where the summary length is controlled to some extent.

Usually, summarization research reports ROUGE-1, ROUGE-2, and ROUGE-LSum. For content selection, and for extractive systems specifically, literature reports recall metrics since systems have no mechanism to control for precision when extracting entire text spans from the source.

[2] https://github.com/google-research/google-research/tree/master/rouge

BERTScore. One limitation of ROUGE is that it is not designed to appropriately reward semantic and syntactic variation in summaries. In order to account for semantic variation and paraphrasing in the evaluation of generated text, Zhang et al. (2019) introduced BERTScore, which calculates the average cosine similarity between contextual token embeddings given by a pretrained BERT model (Devlin et al., 2019).

Given tokenized system summary $\hat{S} = \langle \hat{x}_1, .., \hat{x}_{|\hat{S}|} \rangle$ and tokenized reference summary $S = \langle x_1, .., x_{|S|} \rangle$, let $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ be the token-level embedding of x_i and $\hat{x}_i$, respectively. The metric is defined in terms of recall, precision, and F_1 measure, as follows.

$$\begin{aligned}
\text{BERTScore}_R &= \frac{\sum_{x_i \in S} \text{idf}(x_i) \max_{\hat{x}_j \in \hat{S}} \mathbf{x}_i^T \hat{\mathbf{x}}_j}{\sum_{x_i \in S} \text{idf}(x_i)} \\
\text{BERTScore}_P &= \frac{\sum_{\hat{x}_i \in \hat{S}} \text{idf}(\hat{x}_i) \max_{x_j \in S} \mathbf{x}_j^T \hat{\mathbf{x}}_i}{\sum_{\hat{x}_i \in \hat{S}} \text{idf}(\hat{x}_i)} \\
\text{BERTScore}_{F_1} &= 2 \frac{\text{BERTScore}_R \cdot \text{BERTScore}_P}{\text{BERTScore}_R + \text{BERTScore}_P},
\end{aligned} \tag{2.4}$$

where $\texttt{idf}(x)$ is the inverse document frequency score of token x computed from the test corpus. In this way, BERTScore incorporates importance weighting in order to diminish the effect of non-content words.

Previous work has pointed out limitations in BertScore (Hanna and Bojar, 2021; Nimah et al., 2023), mostly related to the semantic significance of embedding similarity, e.g. favouring lexically close but incorrect translations and penalizing lexically divergent but correct translations. Nevertheless, BertScore has been proven a reliable metric when equipped with importance weighting in highly technical domains such as medical texts (Miura et al., 2021; Hossain et al., 2020). In all our experiments, we report scores using RoBERTa (Liu et al., 2019) as underlying model unless otherwise stated, and apply importance weighting.

Lite^{3}Pyramid. The Pyramid method (Nenkova et al., 2007) is a robust and comprehensive strategy to extract content units from summaries (both reference and system summaries) and evaluate their relevance and coverage w.r.t. a source document(s). Despite its effectiveness, the great amount of manual annotation the method requires has led the community to propose simplified alternatives (Passonneau, 2010; Shapira et al., 2019). In this context, Zhang and Bansal (2021) introduced Lite^{3}Pyramid, a metric that fully automates the Pyramid method. The human annotation of summary content units (SCUs) is replaced by the extraction of *summary triplet units* (STUs), i.e. semantic triplets obtained using a semantic role labeling model. The step of verifying whether an SCU is present in a system summary is replaced by the entailment probability given by a Natural Language Inference

(NLI) model.

Let $\mathcal{M}$ be the set of STUs extracted from reference summary S, and let $f_{\text{NLI}}(e|a,b)$ be a function that quantifies whether a is entailed by b. Then, the coverage of system summary $\hat{S}$ w.r.t. $\mathcal{M}$ is defined as

$$\text{Lite}^3\text{Pyramid} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} f_{\text{NLI}}(e|m, \hat{S}). \tag{2.5}$$

Empirically, Zhang and Bansal (2021) found that defining f_{NLI} as the output probability of the entailment class in a 3-class or 2-class setting worked best.

Auto-J. The outstanding capabilities demonstrated by large language models (LLMs) have prompted their usage as *generative judges*, i.e. as flexible evaluators of text according to predefined human preferences (Zhong et al., 2022; Fu et al., 2023). Li et al. (2023) proposed Auto-J, an LLM fine-tuned to judge how aligned a query response(s) is w.r.t. human preferences of a predetermined criterion. The system, based on Llama 2 (Touvron et al., 2023b), quantifies preference in the following two setups.

Pairwise Judgement. Auto-J is given a query and two responses and asked to write a well-structured critique followed by a verdict of which response is preferred, stated as category label 'Win', 'Tie', or 'Lose'.

Single-Response Judgement. Similar to the pairwise setup, Auto-J is given a query and a single response and asked to write a critique followed by a final rating on a scale of 1 to 10.

Even though Auto-J (Li et al., 2023) is capable of handling a plethora of task scenarios and evaluation protocols, in this thesis we employ it to evaluate the perceived content coverage in candidate summaries.

2.3.2 Redundancy

Content redundancy in a text is assessed with the following metrics, each of which computes a value in the range of $[0; 1]$, where a higher value indicates higher redundancy in a text.

Inverse Uniqueness (IUniq). Defined as $\text{IUniq} = 1 - \text{Uniq}$, where Uniq refers to *uniqueness* (Peyrard et al., 2017), a metric that measures the ratio of unique n-grams to the total number of n-grams. We report the mean among values for unigrams, bigrams, and trigrams.

Sentence-wise ROUGE (RdRL). Defined as the average F_1 ROUGE-L score among all pairs of sentences (Bommasani and Cardie, 2020). Given candidate summary $\hat{S}$,

$$\text{RdRL} = \underset{(x,y) \in \hat{S} \times \hat{S}, x \neq y}{\text{average}} \text{ROUGE-L}(x, y).$$

2.3.3 Local Coherence

We employ a scorer trained to perform the shuffling test to quantify local coherence. The shuffling test, introduced by (Barzilay and Lapata, 2008), is a binary classification task that assigns label $y = 1$ to a multi-sentence text if its sentences are presented in a coherence order (i.e. in the original order) and $y = 0$ if they are presented in a permuted order. Recent analyses of coherence measures (Steen and Markert, 2022) reported that such a shuffle test classifier, dubbed CCL, can obtain promising correlations with human rankings of coherence.

The CCL scorer receives a multi-sentence text and assigns a score between $[0, 1]$ quantifying its local coherence, the higher the better. Following the methodology of Steen and Markert (2022), we train a RoBERTa model (Liu et al., 2019) to distinguish shuffled from unshuffled reference summaries. The model is trained in a binary classification setup with chunks of N consecutive sentences as positive class and their shuffled versions as negative class. Then, the cohesion score of a summary is defined as the positive class probability, averaged over a window of N sentences taken with padding of one sentence.

Previous work (Jwalapuram et al., 2022; Steen and Markert, 2022) has pointed out that local coherence models trained to score a short text might be sensitive to the total number of tokens in the text, possibly scoring shorter texts higher. Steen and Markert (2022) identified the training objective as one of the possible reasons for such lack of robustness. Local coherence models trained with a margin-based ranking loss (Moon et al., 2019; Mesgar et al., 2021) required including shuffle and unshuffled versions of the *same* text in each mini-batch. As such, this training procedure does not impose any constraint on the length difference between positive and negative examples in each batch, which results in a model biased towards short texts. In contrast, local coherence models trained using a binary cross-entropy objective (Laban et al., 2021), such as CCL, can include positive (unshuffled) examples from one document and negative (shuffled) examples from a different document. Steen and Markert (2022) found that including positive and negative examples of varied length in each minibatch alleviated the model's length bias almost in its entirety. In this thesis, we train CCL models using the binary cross-entropy objective with a mixed-length batch recommended by Steen and Markert (2022), hence addressing the reliability issue of metrics for coherence.

2.3.4 Cohesion

Finally, the following measures of cohesion are used in this thesis.

Extended Entity Grid (EEG). The Entity Grid (Barzilay and Lapata, 2008) models lexical cohesion in a text (as a proxy for local coherence) by obtaining the probability of an entity appearing in a determined syntactic role (subject, object, or other) in a sentence, given its role in the previous two sentences. Then, a discriminative model learns a score using entity role transition probabilities and saliency features such as frequency. Later, the feature set was extended to include entity-specific features such as the presence of proper mentions, number of modifiers, among others Elsner and Charniak (2011).

Entity Graph (EGr). (Guinaudeau and Strube, 2013) Models a text as a graph of sentences with edges connecting sentences that have at least one noun in common. Following Zhao et al. (2023), averaged adjacency matrix is reported as a proxy for cohesion.

Lexical Graph (LGr). (Mesgar and Strube, 2016) Lexical Graph (LexG) computes the adjacency matrix of the sentence graph of a text, where two sentences are connected if they have at least two similar-enough content words, i.e. if the cosine similarity between their embeddings is greater than a threshold (zero).

DiscoScore. Zhao et al. (2023) introduced two novel measures that aim to capture the readers' focus of attention when processing a summary. The first measure, FocusDiff (DS-Focus), quantifies the semantic overlap between foci in a reference summary and a system summary. The second measure, SentGraph (DS-Sent), computes the semantic similarity between foci in adjacent sentences, inversely weighted by the distance between sentences. In both cases, foci are modeled as wordpiece tokens. Both measures showed promising correlations with human judgments of coherence, with DS-Focus also being highly correlated with coverage. In this thesis, we use the noun variants, DS-Focus[NN] and DS-Sent[NN], which restrict foci to nouns.

Consecutive ROUGE (CoRL). Similar to RdRL, this metric calculates the ROUGE-L F_1 score between consecutive sentences. We discard unigrams composed of punctuation or stopwords from the calculation of common subsequences.

In general, cohesion metrics rely on linking subsentential units (i.e. words or phrases) located in different sentences. Hence, the reliability of these metrics depends on how well these units can be identified. For instance, metrics EEG, EGr, and DiscoScore rely on noun extraction, whereas LGr and CoRL rely on the identification of punctuation and stopwords (usually pre-defined in a list). Given that the recognition of nouns, punctuation, and stopwords is fairly accurate with modern NLP tools across domains, we can consider these cohesion metrics reliable, especially when measuring *lexical* cohesion. However, as we will see in Chapters 4 and 5, it is imperative to examine cohesion measurements along with redundancy measurements in order to make an informed judgment about the cohesion of a summary.

Chapter 3

Content Selection as Human Memory Simulation

Text summarization systems face the core challenge of identifying and selecting important information in the original text. In this chapter, we focus on controllable content selection in unsupervised extractive summarization of long, structured documents. We introduce summarization strategies that leverage how information is discretized in content units and organized in human memory according to the Micro-Macro Structure theory (KvD; Kintsch and van Dijk 1978). We find that these simulated structures of content in human memory can be exploited to capture the relevance of content units in highly technical documents, and use scientific document summarization as case study. Extensive automatic and human evaluations demonstrate that the proposed summarization system is capable of controlling the level of generality or technicality of the extracted content by manipulating the constraints of the memory structure.

3.1 Introduction

Content selection plays a pivotal role in the summarization pipeline, where the system determines the content units to be reproduced in the final summary. Following the discretization of content in the source, content selection reduces this content pool to a subset by either fusing or selecting parts of it.

When producing a general summary for informative purposes, the goal is to cover as much relevant content from the source as possible, prioritized by relevance and subject to a pre-determined length budget. In the context of extractive summarization, content selection is reduced to the step of extracting text spans from the source document, usually sentences.

Despite notable advances in recent years, neural networks still face challenges in selecting content based on relevance. On the one hand, they often resort to learning task shortcuts such as selecting based on sentence position or keyword presence (Narayan et al., 2018b; Kedzie et al., 2018). On the other hand, modern summarization benchmarks such as CNN/DailyMail (Hermann et al., 2015) only provide one reference summary per document. Hence, when training on these benchmarks, there is limited information on what a well-covering summary should look like, compared to training scenarios where there is more than one reference summary per document (Over et al., 2007).

Addressing these challenges, prior research (Fang and Teufel, 2014; Zhang et al., 2016; Fang, 2019) took a step back and investigated how the human mind organizes and select content. Their approaches sought to model key parts of the pipeline, such as source representation and content selection, after cognitive processes known to be involved in memory management. Psycho-linguistic theories of human reading comprehension such as the Construction-Integration (CI; Kintsch 1988) and the KvD theory (Kintsch and van Dijk, 1978) provide a rich theoretical foundation on how content is represented and selected from human memory during tasks such as summarization.

In this chapter, we contribute to this line of research by leveraging how content is explicitly represented and organized in human working memory –a type of short-term memory– according to the KvD theory, in order to inform a summarization model about which information bits are relevant in a document. According to KvD, human working memory stores knowledge bits in discrete, indivisible units called 'content units' which we model as semantic propositions. Furthermore, these content units are organized in a tree structure, called *memory trees*, where edges represent cohesive ties among units. As reading progresses, new content units are added to the memory tree, and older units are discarded. We exploit the properties of nodes in these evolving trees to quantify relevance of content units in a document, such as their position in the memory tree or their degree of connection.

In summary, we tackle the problem of content selection in single-document extractive summarization in an unsupervised manner, taking as case study the summarization of scientific articles from the PubMed and arXiv dataset (Cohan et al., 2018), using the body of the article as documents and their abstracts as summaries. This chapter presents the following contributions:

- We investigate a range of system configurations that leverage structures of content units obtained from a reading comprehension cognitive model.
- We demonstrate that these configurations are effective at ranking highly relevant con-

tent units, which are then used to guide the production of extractive summaries from long, structured documents.

- We formulate the problem of sentence selection as an optimization problem with a budget in number of tokens as soft constraint. The resulting summaries present less variability in length, hence ensuring a fairer comparison among models in terms of automatic metrics.

Extensive experiments show that configurations of our summarizer exploiting properties of memory trees outperform systems that rely only on frequencies in terms of informativeness and content coverage metrics. Further human evaluations and error analysis reveal that our best system configuration provides users with less specific yet relevant key content.

3.2 Problem Setup

We formulate the problem of unsupervised content selection in extractive summarization as the task of scoring sentences in a document followed by the selection of a subset of sentences as the summary. In this section, we start by providing an overview of the proposed summarization pipeline. Then, the procedure used to build propositions and the KvD algorithm proposed by Fang and Teufel (2014) are briefly explained. Afterward, we introduce the proposed sentence scoring strategies, inspired by KvD simulation, followed by our sentence selection strategy.

3.2.1 Pipeline Overview

The pipeline of our summarization system is depicted in Figure 3.1. Input document $\mathcal{D}$ is consumed one sentence at a time by the reading simulator. At each step, propositions are extracted from the incoming sentence, and one memory cycle is executed. At the end of the memory cycle, the scores of the propositions in the working memory tree are updated. Once the document has been completely read, the final score of propositions is aggregated into sentence scores, which are then used by our Knapsack selector to select the final summary. Next, we elaborate on each step of the pipeline.

3.2.2 Proposition Building

The first step in our summarization pipeline consists of extracting KvD micro-propositions of the form `predicate(arg0,arg1,...)` from each incoming sentence. We reproduce

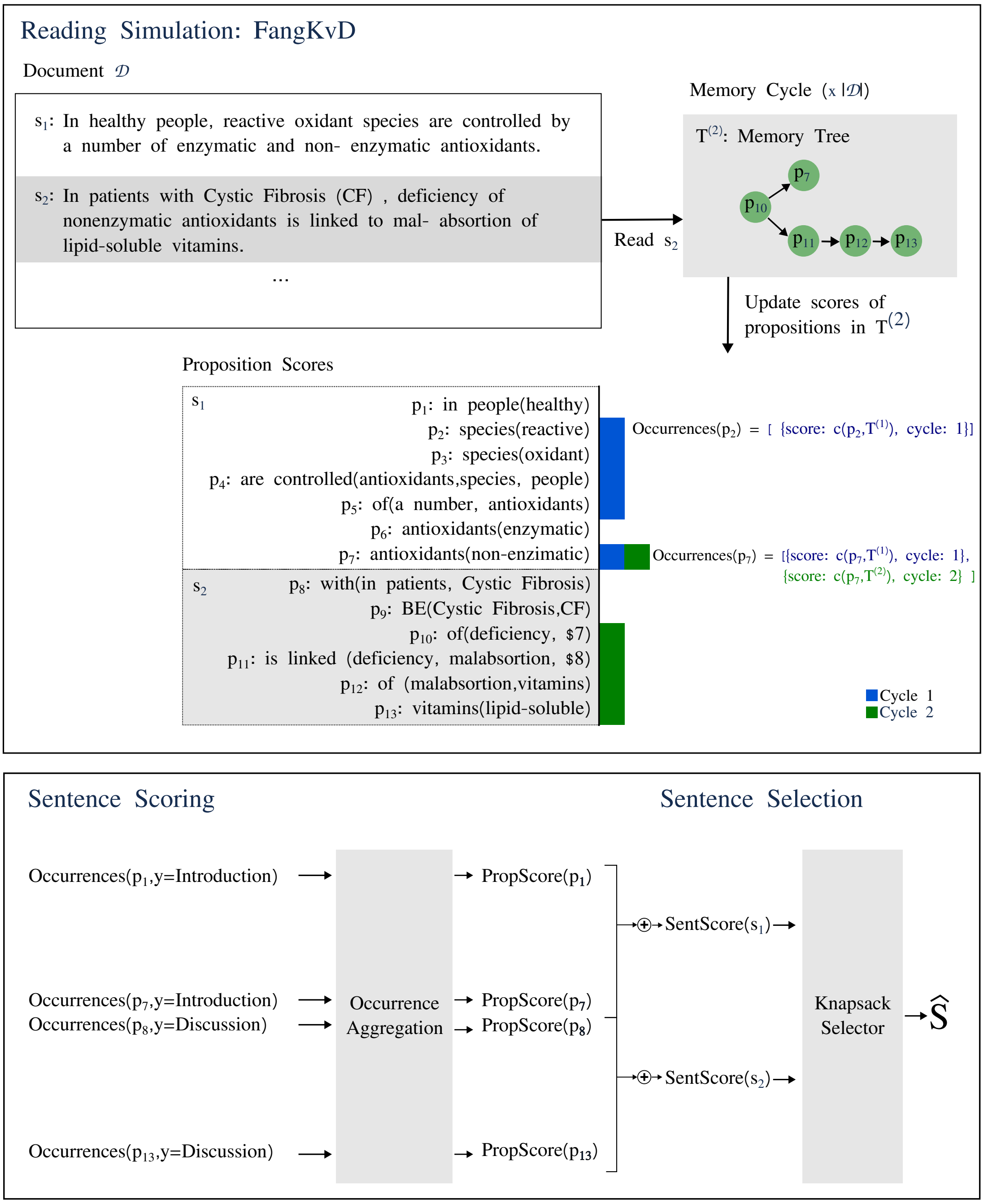

Reading Simulation: FangKvD
Document $\mathcal{D}$
s_1: In healthy people, reactive oxidant species are controlled by a number of enzymatic and non- enzymatic antioxidants.
s_2: In patients with Cystic Fibrosis (CF) , deficiency of nonenzymatic antioxidants is linked to mal- absorption of lipid-soluble vitamins.
...
Read s_2
Memory Cycle (x $|\mathcal{D}|$)
$T^{(2)}$: Memory Tree
p_7
p_{10}
p_{11}
p_{12}
p_{13}
Update scores of propositions in $T^{(2)}$
Proposition Scores
s_1
p_1: in people(healthy)
p_2: species(reactive)
p_3: species(oxidant)
p_4: are controlled(antioxidants,species, people)
p_5: of(a number, antioxidants)
p_6: antioxidants(enzymatic)
p_7: antioxidants(non-enzimatic)
s_2
p_8: with(in patients, Cystic Fibrosis)
p_9: BE(Cystic Fibrosis,CF)
p_{10}: of(deficiency, \$7)
p_{11}: is linked (deficiency, malabsortion, \$8)
p_{12}: of (malabsortion,vitamins)
p_{13}: vitamins(lipid-soluble)
Occurrences(p_2) = [{score: c(p_2,$T^{(1)}$), cycle: 1}]
Occurrences(p_7) = [{score: c(p_7,$T^{(1)}$), cycle: 1}, {score: c(p_7,$T^{(2)}$), cycle: 2}]
🟦Cycle 1
🟩Cycle 2
Sentence Scoring
Sentence Selection
Occurrences(p_1,y=Introduction)
Occurrences(p_7,y=Introduction)
Occurrences(p_8,y=Discussion)
Occurrences(p_{13},y=Discussion)
Occurrence Aggregation
PropScore(p_1)
PropScore(p_7)
PropScore(p_8)
PropScore(p_{13})
⊕ SentScore(s_1)
⊕ SentScore(s_2)
Knapsack Selector
$\hat{S}$

Figure 3.1: Pipeline of the proposed summarization system consisting of KvD reading simulation, sentence scoring, and sentence selection, using the simulation example in Fig. 2.1.

the procedure proposed by Fang and Teufel (2014), which exploits grammatical dependencies in dependency and constituency trees to aggregate subjects and complements of a predicate into propositions.

The procedure returns proposition set P and consists of the following main steps, showcased with the example in Figure 3.2,

- Starting from a dependency tree, children nodes are merged into their respective head nodes if the children are nominal or non-core dependants, such as determiners, quantifiers, negations, auxiliaries, or part of a multi-word expression. In Fig. 3.2a, single-token modifiers are collapsed into their head nodes (e.g. `this`→`model`), and compound phrases are joint (e.g. `galaxy`→`formation`).

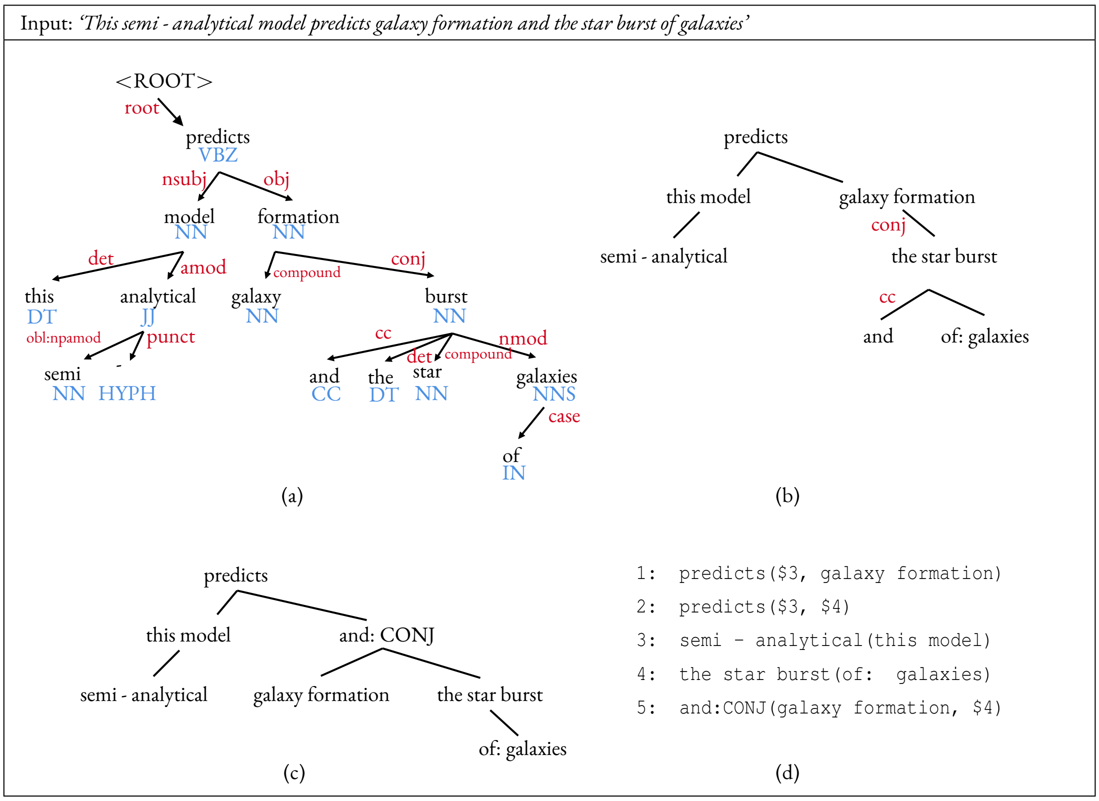
- In case a coordination relation exists between two tokens, a new coordination node, CONJ, is created and all coordinated nodes are transplanted under it. In Fig. 3.2c, `and: CONJ` is created and coordinated nodes `galaxy formation` and `the start burst` are transplanted under it.

- Head nodes are made into predicates with their children nodes as arguments. For special coordination nodes, each coordinated node is propagated into the coordinating parent proposition as an argument. In Fig. 3.2d, coordinated node `galaxy formation` is propagated into `predicts` as its argument.

Moreover, the procedure unifies active and passive voice, clauses are treated as embedded propositions, and objects in conjoined preprositional phrases are aggregated and attached to their head nodes. Non-subsective and subsective adjectives (Kamp and Partee, 1995) are also taken into account: subsective adjectives are turned into predicates with their respective noun as an argument, whilst non-subsective adjectives are collapsed into their nouns like a compound phrase. In order to make this distinction, the procedure makes use of the lexicon proposed by Lin (1998). For more details, please refer to Appendix A in Fang (2019).

Proposition Overlap. Propositions are connected by quantifying the semantic overlap between their functors –predicates and arguments. We reproduce the procedure in Fang and Teufel (2016), which employs coreference chains (enhanced by lexical chains) and 'semantic transitions' between two functors to quantify their degree of membership to a common semantic concept. Proposition overlap is defined as the average overlap score calculated pairwise between functors in one proposition and functors in another, where functor overlap is calculated in the following way.

Given document D, the procedure starts by performing coreference resolution of all entities and referring expressions in D. Then, coreference chains are expanded using the lexical chaining algorithm proposed by Galley and McKeown (2003), although allowing verbs as chain members. The chaining algorithm employs a disambiguation graph in which nodes correspond to word types in a document and edges between words are labeled after their



```
1:  predicts($3, galaxy formation)
2:  predicts($3, $4)
3:  semi - analytical(this model)
4:  the star burst(of:  galaxies)
5:  and:CONJ(galaxy formation, $4)
```

Figure 3.2: Step-by-step construction of propositions from an input sentence, starting from obtaining its dependency tree in Stanford Dependency format (a), merging dependent nodes into head nodes (b), promoting coordinating conjunctions to head status (c), to finally build propositions from non-leaf nodes (d).

semantic relation under any of their respective sense, possibly leading to multiple edges between nodes. Senses in edges are labeled using WordNet (Miller, 1995) but the algorithm ultimately restrains all occurrences of a word in a document to take the same sense.

Then, let G_{funct} be the disambiguation graph derived from D, p, q propositions extracted from D, and let $a \in p$ and $b \in q$, be functors in their respective propositions. The procedure defines functor overlap function $\omega : V[G_{\mathrm{funct}}] \times V[G_{\mathrm{funct}}] \mapsto \mathbb{R}$ with range $[0; 1]$, where 0 denotes no semantic overlap between functors and 1 means that functors refer to the same concept.

The objective of the procedure is to favour fewer semantic transitions between functors a and b by defining distance $d_e = \alpha_{\mathrm{overlap}}^{-t}$, where edge e connects a and b in G_{funct}, and $\alpha_{\mathrm{overlap}} \in [0; 1]$ is an attenuation factor. Hyper-parameter t is set depending on the semantic relationship between a and b (encoded in the edge label) with empirical values defined in

Dist. Param. (t)	Noun	Verb	Derivation
0	synonymy	-	-
1	hypernymy	synonymy	noun-to-verb
2	sibling	hypernymy	-

Table 3.1: Configuration of the distance hyper-parameter t (Dist. Param.) dependent on the semantic relation between nodes in the disambiguation graph. Reproduced from Fang and Teufel (2016).

Table 3.1. Hence, $\omega(a, b)$ is defined as

$$\omega(a,b) = \begin{cases} \frac{1}{\sum_{e \in F} d_e} & \text{if } L(a) = L(b) \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $L(a)$ is the coreference chain a is member of, and F is the shortest path between a and b in G_{funct}.

Finally, proposition overlap is calculated as follows. Let us assume that proposition p is the parent of q in the memory tree, and let $A^*(p, q)$ be the optimal alignment between all functors in p and all arguments in q, i.e. the predicate of the child does not participate in the alignment. Alignment A^* is defined as the maximum matching that can be obtained greedily in the weighted bipartite graph formed from both comparing sets. Then, the proposition overlap score between p and q, $\phi_{\text{Fang}}(p, q)$, is defined as:

$$\phi_{\text{Fang}}(p,q) = \frac{1}{|A^*|} \sum_{\langle a,b \rangle \in A^*} \omega(a,b). \tag{3.2}$$

It is important to note that ϕ_{Fang} defines *asymmetric* overlap between propositions, a property the simulation procedure exploits when searching for an appropriate place to attach incoming propositions to the current memory tree. We elaborate more on this in the next section.

3.2.3 FangKvD

In this part, we describe FangKvD, our proposed sentence scoring system simulating KvD reading. Although the system reproduces KvD simulation as implemented by Fang (2019), our contribution lies in strategies for proposition scoring that exploit the shape of memory trees more efficiently for the task of summarization. The system incrementally builds a directed graph where nodes represent propositions and edges represent proposition overlap.

Working memory and long-term memory (LTM) are differentiated by a special attribute in each node, indicating which nodes are 'active' or currently in working memory.

At a high level, FangKvD performs the following steps. Reading of document D proceeds in memory cycles, one sentence at a time. In each cycle, propositions are extracted from the incoming sentence and attached to the current memory tree, one by one to the most promising candidate according to a criterion that favours the strength of proposition overlap. Then, the algorithm prunes the memory tree to a predefined size following a strategy that favours the most recently read propositions. At this point, the root of the tree is adjusted if necessary, favouring more connected nodes as roots. Afterward, the score of the remaining propositions is calculated considering their position in the memory tree. Then, this score is updated in a dedicated data structure that keeps track of each proposition occurrence to aggregate their scores once reading is completed.

Before diving into the details of each of the aforementioned steps, let us lay down some notation. Let s_k be the sentence read in cycle k, G the proposition graph built until cycle $k-1$, and T the working memory tree at the beginning of the cycle, with node set $V[T] \subset V[G]$ and edge set $E[T] \subset E[G]$. Similarly, let G_{LTM} be the long-term memory graph with $V[G_{LTM}] \subset V[G]$, $E[G_{LTM}] \subset E[G]$, and $V[T] \cap V[G_{LTM}] = \emptyset$.

Extracting and Attaching Incoming Propositions. Given sentence s_k, proposition set P_k is extracted according to the procedure detailed in Section 3.2.2. Propositions in P_k are added to T one at a time in an iterative way. In each iteration, an attachment score is calculated for each (p, q) pair, where $p \in P'_k$ – the set of nodes not yet attached– and $q \in V[G]$. The pair with the highest attachment score is chosen and attached accordingly. Formally, the attachment score is defined as

$$\text{AttachmentScore}(p, q) = \phi_{\text{fang}}(p, q) \cdot \alpha_{\text{lvl}}^{\text{depth}(p)} \cdot \alpha_{\text{rec}}^{\text{recall}(p)} \tag{3.3}$$

where $\phi_{\text{fang}}(p, q)$ is the functor overlap function when p is added as a child of q, α_{lvl} and α_{rec} are model parameters in range $(0, 1]$, and $\text{depth}(p)$ is the depth of node p w.r.t. the root. The term $\text{recall}(p)$ is the cost of connecting q to p when $p \in V[G_{LTM}]$, i.e. when p is not in the current memory tree and hence needs to be recalled. The recalling of p consists of bringing the shortest path connecting p and any node in T back to T, with $\text{recall}(p)$ being the length of this path. As such, $\text{AttachmentScore}(p, q)$ favours not only strong proposition overlap but also the connection of incoming propositions to positions closer to the root, and penalizes the recall of long proposition paths from long-term memory. Then, the optimal attachment pair is defined as

$$(p^*, q^*) = \operatorname*{argmax}_{p \in P'_k, q \in V[G]} \text{AttachmentScore}(p, q). \tag{3.4}$$

Once all propositions in P_k have been attached to T, the procedure proceeds to control the shape of the resulting memory tree. In practice, Fang (2019) restricts the recalling path to have at most one non-activated node, i.e. only one proposition can be recalled from LTM, for computational reasons.

Adjusting the Root. When adding all propositions to T during the first cycle, $k = 0$, the root is chosen as follows. Taking T as a directed graph with edge weight set to $1/\phi_{\text{fang}}(p, q), \forall(p, q) \in E[T]$, the node with the highest closeness centrality is chosen as the root. The closeness centrality of a node in a graph is defined as the inverse of the sum of all shortest paths from said node to all other nodes in the graph. In this way, nodes with weak proposition overlap with their neighbours will not be chosen as roots. In case of a tie, i.e. two or more propositions having the same highest degree of centrality, the proposition with functors closer to the root of the dependency tree is chosen.

Afterward, when $k > 0$, the root is changed only if the change promotes more nodes closer to the new root than it demotes further away from the root. Let v be the root of T at cycle k, T_v the subtree rooted at v, and $\text{children}(v)$ the set of children nodes of v. Then, the optimal candidate for new root is determined by the weighted size of its subtree,

$$u = \underset{\hat{u} \in \text{children}(v)}{\text{argmax}} \ \phi_{\text{fang}}(v, \hat{u}) \sum_{z \in T_{\hat{u}}} w_z, \tag{3.5}$$

where w_z is a weighting parameter set to 0.05 for nodes that were recalled from LTM (i.e. non-active) and 1.0 otherwise. Overlap score $\phi_{\text{fang}}(v, \hat{u})$ takes into account the change in edge direction when promoting $\hat{u}$ to root and v to its child. However, a root change is only performed if the weighted size of T_u is greater than that of T_v, i.e. if $\phi_{\text{fang}}(v, u) \sum_{z \in T_u} w_z > \phi_{\text{fang}}(u, v) \sum_{z \in T_v} w_z$. This procedure is applied recursively until no more root changes are possible.

Pruning Working Memory. Next, memory tree T is pruned to have at most `WM` nodes. The selection of which nodes remain follows KvD's leading edge strategy. Starting from the root, T is traversed in topological order until reaching a leaf node, selecting each node visited along the way. At this point, if the amount of select nodes is less than `WM`, nodes are selected in breadth-first traversing order (starting from the root) until capacity is reached or until all nodes are traversed. Finally, nodes not selected are pruned from T and 'moved' to LTM, i.e. they are deactivated.

Scoring of Proposition Occurrence. We call an *occurrence* of a proposition p during simulation to every instance of p where it appears as a node in a memory tree. Since a memory cycle can keep a proposition in the tree for the next cycle, there can be many such instances for a certain p.

Given memory tree T, the following occurrence score functions $c_*(p, T), \forall p \in T$, are defined

$$\begin{aligned} c_{cnt}(p) &= 1, \\ c_{lvl}(p) &= \frac{1}{\text{depth}(p)}, \\ c_{deg}(p) &= \text{degree}(p), \\ c_{sub}(p) &= |T_p|, \end{aligned} \tag{3.6}$$

where degree(p) is the total degree (indegree + outdegree) of p and T_p is the size of subtree rooted at p.

Then, the occurrences of nodes in T are logged into an auxiliary data structure along with its occurrence score and the cycle it was calculated on. Let us denote Occurrences(p) the entry in the data structure for proposition p, which would return the set of occurrences in the entire document, along with each occurrence information.

Aggregation of Occurrence Score. Once the document has been completely read, we aggregate the scores of occurrences into a single proposition score to be used for sentence selection. At this stage, the organization of the document is taken into account, e.g. whether the document is divided into sections as is the case, for example, in scientific articles.

Occurrence scores are aggregated depending on whether we consider occurrences in the entire document or occurrences by section, as follows

$$\begin{aligned} n_{cnt} &= \sum_{x \in \text{Occurrences}(p)} c_*(x), \\ n_{wgt} &= \sum_{y \in Y} \left[r_y \cdot \left(\sum_{x \in \text{Occurrences}(p,y)} c_*(x) \right) \right], \\ n_{exp} &= \sum_{y \in Y} \left[\sum_{x \in \text{Occurrences}(p,y)} c_*(x) \right]^{r_y}, \end{aligned} \tag{3.7}$$

where $c_*(\cdot)$ is any of the occurrence scoring strategies in Eq. 3.6. Term Occurrences(p, y) is the set of occurrences of p during simulation of section y of the document, and r_y is the ratio of sentences in y. For instance, for the *Introduction* section of a scientific article,

$$r_i = \frac{\text{Number of sentences in the introduction}}{\text{Total number of sentences in the document}}.$$

Finally, the score of a proposition p is defined as

$$\text{PropScore}(p) = 1 - (1 - \rho)^{n_*(p)}, \tag{3.8}$$

where $n_*(\cdot)$ is any of the aggregation strategies in Eq. 3.7 and ρ is a parameter denoting the probability of reproducing any proposition during summary production. For ease of notation, PropScore(p) with strategy configuration c_a and n_b will be referred to as `a-b`. For instance, system `Lvl-Exp` refers to a configuration that combines occurrence scoring by node depth (c_{lvl}) and aggregates the scores by document section as an exponentially weighted sum (n_{exp}). Moreover, notation `a-b[X]` denotes an `a-b` system configuration with working memory capacity of `X`.

Note that Equation 3.8 can be interpreted as a generalization to the concept of reproduction probability proposed by KvD. As such, KvD's expression in Equation 2.1 corresponds to the case when using raw frequency counts, i.e. using occurrence scoring c_{cnt} and aggregation strategy n_{cnt}. Moreover, the flexibility in configuration allows one to choose to exploit either the shape of memory trees or to exploit the structure of a document, or both at the same time. First, c_{lvl}, c_{deg}, and c_{sub} do exploit the shape and configuration of the trees, whereas c_{cnt} does not. Second, n_{wgt} and n_{exp} leverage the fact that the document is divided into sections, whereas n_{cnt} does not.

Sentence Scoring. Finally, the score of sentence s_i is defined as the sum of the score of all its composing propositions, as follows

$$\text{SentScore}(s_i) = \sum_{p \in P_i} \text{PropScore}(p), \tag{3.9}$$

where P_i is the set of propositions extracted from s_i.

3.2.4 Sentence Selection

Previous work has pointed out that ROUGE score is sensitive to the length of the summary and summarization models should only be compared against each other if they produce summaries of similar length (Narayan et al., 2018a; Schumann et al., 2020). For this reason, we extract summaries according to a budget of tokens instead of picking a fixed number of sentences regardless of their length as is normally done in the literature.

We pose the problem of selecting a subset of sentences from a document as a 0-1 knapsack problem with a *soft* budget constraint, i.e. the optimal summary is allowed to have more tokens than the budget as long as the length difference (in number of tokens) is minimal. In other words, the objective is to maximize the total score of selected sentences while keeping the total number of tokens as close to a budget `B` as possible. Our approach is simple and is based on previous work applying knapsack optimization for sentence selection and compression (Naserasadi et al., 2019; Shichel et al., 2021), with the crucial difference that we set a soft budget constraint.

Given document $D = \langle s_0, .., s_{|D|} \rangle$, where each sentence has a weight equal to its number of tokens and value $\text{SentScore}(s_i)$; and given weight limit $\mathtt{B}$, the objective is to obtain sentence subset $S = \{s_i \mid x_i = 1\}$ that maximizes:

$$\begin{aligned} &\sum x_i \cdot \text{SentScore}(s_i), \text{ s.t.} \\ &|\sum_{i \in [0,|D|)} x_i |s_i| - \mathtt{B}| \leqslant \epsilon, x_i \in \{0,1\} \end{aligned} \tag{3.10}$$

where $|s_i|$ is the number of tokens in s_i and ϵ is the margin S is allowed to overflow the budget.

3.2.5 Limitations

The proposed summarization system presents the following limitations. On the one hand, the reproduced KvD reading simulator depends heavily on out-of-the-box NLP tools and external linguistic resources. First, proposition quality crucially depends on the quality of dependency and constituency parse trees, as well as an external adjective lexicon. Given that the available parsers are domain-dependent, the quality of parse trees can be severely compromised when used in domains other than the ones they were trained for. In our case, the parsers employed were trained over newswire, while being used on scientific text. Second, the reproduced proposition overlap procedure heavily depends on an out-of-the-box coreference resolution model. Word disambiguation during lexical chaining depends instead on WordNet, which is quite limiting to use even in the newswire domain. Third, during reading simulation, the constrained amount of content units in working memory at any given time poses a limitation to how much information the system has access to when updating the score of memory tree nodes. It is entirely possible that some propositions are pruned away and never recalled again, in which case their score will be zero.

On the other hand, the proposed proposition scoring and sentence selection strategies present the following limitations. Even though aggregation of occurrences over sections does leverage document structure, its effectiveness depends on whether the document has clearly separated sections, e.g. proper segmentation of sections in a scientific article. Finally, the Knapsack sentence selection is quite effective at controlling the length of candidate summaries, resulting in a length distribution with a very low standard deviation. While this property results in an accurate and reliable calculation of ROUGE scores for instances with reference summary length close to the pre-defined budget, the scores become less reliable as reference summary length differs more and more from the budget. In practice, however, as we show in our experiments, the reference summary length in the analyzed datasets shows low standard deviation, meaning that extreme differences between candidate and reference length are the exception rather than the rule.

3.3 Experimental Setup

We investigate all possible combinations of the occurrence scoring and aggregation strategies presented in Sect. 3.2.3. We refer as 'FangKvD system' to any instantiation of occurrence scoring–aggregation configuration applied to the FangKvD reader alongside the Knapsack selector.

3.3.1 Datasets

We use PubMed and arXiv datasets collected by Cohan et al. (2018), composed of scientific articles in English with their abstracts as reference summaries. Articles with abstracts with less than 50 tokens and more than 300 tokens were discarded, as well as articles with documents with less than 100 tokens. We only consider the Introduction, Discussion, and Conclusion sections in each article, as preliminary experiments showed that most information needed to summarize the document is found there. After filtering out articles without none of these sections, PubMed was left with 104 814 articles in the training set, 5344 in the validation set, and 6025 in the test set. For arXiv, these number were— respectively— 183 799, 5623, and 5803. It is worth noting that we found a discrepancy in both datasets. Text from the 'article' field (in theory the concatenated sections) would not always have the same text as the 'sections' field. Hence, we chose data from the 'sections' field as input document.

3.3.2 Implementation Details

Proposition Building and Overlap. The proposition extraction procedure employs Stanford CoreNLP v3.9.2 (Manning et al., 2014) for parsing of constituency trees and dependency trees in the Stanford Dependency formalism (De Marneffe et al., 2006). For proposition overlap, the Stanford coreference resolver (Raghunathan et al., 2010) is used to obtain initial coreference chains, which are then extended using the lexical chaining algorithm proposed by Galley and McKeown (2003). Regarding hyper-parameters, we use the default values stated in Fang and Teufel (2016), i.e. the attenuation factor α_{overlap} is set to 0.7.

FangKvD. In order to further account for section organization in scientific articles, we start reading each section with an empty memory tree. This action allows us to generate memory trees that reflect only the argumentation of the current section but still have access to the complete set of propositions in the document, in case a content unit is referenced back to a previous section. In this way, the reading simulator produces memory trees with nodes only relevant to the current section.

Similar to proposition building, hyper-parameters are set to their default values reported in Fang (2019), $\alpha_{lvl} = 0.6$ and $\alpha_{rec} = 0.05$, and the recalling path is restricted to have at most one non-activated node. Reproduction probability is set to 0.3, a value found empirically by KvD for human summarization tasks (Kintsch and van Dijk, 1978). Working memory capacity `WM` is set to values $\{5, 20, 50, 100\}$. In terms of notation, as mentioned previously, system `a-b[X]` denotes a configuration with strategies c_a and n_b, and $\texttt{WM} = X$.

For occurrence aggregation in PubMed, we set ratios of sentences per section to $r_i = 0.33$, $r_d = 0.53$, and $r_c = 0.14$, pre-calculated from the training set. For arXiv, we set $r_i = 0.62$, $r_d = 0.16$, and $r_c = 0.22$. During sentence selection, we set budget `B` to 205 for PubMed and to 190 for arXiv–the average reference summary length in each corresponding training set. For both datasets, we set budget margin $\epsilon = 50$, the standard deviation in reference summaries.

3.3.3 Comparison Systems

We compare our models against unsupervised and supervised baselines. For all baseline systems, the Knapsack selector is applied on top of their respective sentence scores to ensure a fair comparison between systems.

Extractive Oracle. The extractive oracle extracts candidate summary sentences that maximize their ROUGE scores w.r.t. a reference summary. Knapsack selection is applied over partial candidate summaries, where the score of candidate sentences is modeled as the sum of ROUGE-1 and ROUGE-2 recall values of the partial summary w.r.t. a reference summary. This baseline is labeled as Ext-Oracle.

Unsupervised Baselines. We make the distinction between *completely* unsupervised systems and unsupervised systems that require some form of finetuning using data in the target knowledge domain, the latter being marked with (*). The following systems are compared:

- Lead. First sentences until budget is reached.

- Random. The score of each sentence is its probability, drawn from a uniform distribution. Then the selection strategy is applied.

- Random-Wgt. The score of each sentence is its probability, proportional to the ratio of the section it belongs to.

- NoTree. System configuration that counts proposition occurrences in the source document instead of occurrences in memory trees.

- TextRank (Mihalcea and Tarau, 2004). Completely unsupervised system that models a document as a graph of sentences where an edge connects sentences s_i and s_j if they share content words (tokens left after discarding stopwords) with edge weight $E_{TR}^{(i,j)}$ defined as
$$E_{TR}^{(i,j)} = \frac{|w_k; w_k \in s_i \wedge w_k \in s_j|}{\log(|s_i|) + \log(|s_j|)}, \tag{3.11}$$
where $|s_i|$ denotes the number of content words in s_i. Then, TextRank employs the PageRank algorithm (Brin and Page, 1998) with damping factor $d = 0.85$ to obtain the eigen-vector centrality of nodes (sentences), which we then use as $\mathtt{SentScore}(s_i)$ in Eq. 3.10 to be used by the Knapsack selector. Following `GenSim` (Rehurek and Sojka, 2010), we remove stopwords and apply stemming.

- LexRank (Erkan and Radev, 2004). Similar to TextRank, LexRank models the relevance of sentences in a document as the node eigen-centrality in a graph where the edge weight between sentence nodes s_i and s_j is defined as
$$E_{LR}^{(i,j)} = \cos(\text{TF-IDF}(s_i), \text{TF-IDF}(s_j)) \tag{3.12}$$
where TF-IDF(s_i) is the TF-IDF vector representation of content words in sentence s_i and $\cos(a, b)$ is the cosine similarity between a and b. This modeling corresponds to the *Continuous* LexRank in Erkan and Radev (2004).

- PacSum (Zheng and Lapata, 2019). Also modeling a document as a sentence graph, PacSum defines the *unnormalized* edge weight matrix as $\hat{E}_{PC}$, where $\hat{E}_{PC}^{(i,j)} = v_i^T v_j$ represents the similarity between vector representations (as the dot product) of sentences s_i and s_j. Then, the *normalized* edge weight matrix is defined as
$$E_{PC}^{(i,j)} = \max(0, \hat{E}_{PC}^{(i,j)} - [\min \hat{E}_{PC} + \beta(\max \hat{E}_{PC} - \min \hat{E}_{PC})]). \tag{3.13}$$
In contrast to TextRank and LexRank, PacSum models node centrality as the degree centrality weighted by sentence relative position. Formally, the score of a sentence is defined as
$$\mathtt{SentScore}_{PC}(s_i) = \lambda_1 \sum_{j<i} E_{PC}^{(i,j)} + \lambda_2 \sum_{j>i} E_{PC}^{(i,j)}, \tag{3.14}$$

where λ_1 and λ_2 are hyper-parameters controlling the contribution of similarity between previous sentences and future ones, respectively; and $\lambda_1 + \lambda_2 = 1$. For computational purposes, we limit connection to sentences in a window of size 200.[1] Finally, we report results for two sentence representation strategies investigated in Zheng and Lapata (2019), TF-IDF and a BERT-based sentence embedding, which we initialize with SciBERT (Beltagy et al., 2019). These systems are dubbed PacSum[TF-IDF] and PacSum[SciBERT], respectively. We use the default hyper-parameters ($\lambda_1 = -2$, $\lambda_2 = 1$, and $\beta = 0.6$) reported in Zheng and Lapata (2019) and consider these systems completely unsupervised.

- PACSUM-FT*. PacSum[SciBERT] system with hyper-parameters fine-tuned over a sample of 1000 documents from each training set (uniformly sampled) following the procedure therein. For PUBMED, we set $\lambda_1 = 0.4$, $\lambda_2 = 0.6$, and $\beta = 0.9$, whereas for ARXIV, $\lambda_1 = 0.5$, $\lambda_2 = 0.5$, and $\beta = 0.9$.

Supervised Baseline. We add a linear classifier layer on top of SciBERT (Beltagy et al., 2019) and fine-tune it over the same subset used for PACSUM-FT, and dub this system EXT-SCIBERT. Similarly to Cohan et al. (2019), we consume each document in chunks of fixed numbers of sentences and use the pretrained model served by HuggingFace.[2] Fine-tuning is performed for two epochs using a batch size of 8, and document chunk size of 5. For optimization, we use Adam (Loshchilov and Hutter, 2019) with a fixed weight decay parameter of 0.1. Additionally, we use a slanted triangular learning rate scheduling with 10% of total training steps as warm-up and a top value of $1e^{-5}$. We accumulate gradients for 16 training steps and clip gradients by norm value at 0.1.

Finally, it is important to note that, given the high technicality of the domain analyzed in this chapter –the scientific domain–, we do not include supervised baselines that require the calculation of coreference chains or rhetorical structure trees over the input document, such as DiscoBERT (Xu et al., 2020), because of their limited applicability in out-of-domain scenarios.

3.3.4 Automatic Evaluation

We report ROUGE recall scores (Lin, 2004) –instead of F_1 scores– to evaluate lexical coverage and lexical relevance, as well as BERTScore recall score (Zhang et al., 2019) for seman-

[1] Such a limitation was possibly not considered by Zheng and Lapata (2019) since their model was not designed for long documents, and instead was tested on the CNN/DM dataset in which documents are 50 sentences long in average.

[2] https://huggingface.co/allenai/scibert_scivocab_uncased

tic relevance. Content coverage at the sub-sentential level is measured with Lite^{3}Pyramid (Zhang and Bansal, 2021), which calculates the average entailment score between a candidate summary and semantic triplets (STUs) extracted from the reference summary, as detailed in §2.3. Following the recommended setup, we report the average probability of the entailment class given by an NLI model without any finetuning, l^{3c},[3] and finetuned on the summary content unit coverage step annotated in TAC 2008 (Dang et al., 2008), denoted as p^{2c}.[4]

Furthermore, we measure content coverage at the propositional level by quantifying how many propositions extracted by a system are also present in the extractive oracle summary. Let $\mathcal{P}$ be the set of propositions present in the extractive oracle summary of document $\mathcal{D}$, and let $\hat{\mathcal{P}}$ be the set of propositions in candidate summary $\mathcal{S}$. We define recall (R) and precision (P) as follows

$$R = |\mathcal{P} \cap \hat{\mathcal{P}}|/|\mathcal{P}|, \; P = |\mathcal{P} \cap \hat{\mathcal{P}}|/|\hat{\mathcal{P}}|. \tag{3.15}$$

A higher value of R means that more summary-worthy content is being captured. Along with recall and precision, we also report the F_1 score.

Finally, statistical significance is tested using the Bootstrap method with a 95% confidence interval and 1000 iterations, with metric scores reported corresponding to the mean of the central bootstrap bin.

Metric Reliability. As mentioned in § 2.3, metrics such as ROUGE and BERTScore present limitations that might impact their reliability if not properly accounted for.

For ROUGE, reliability is impacted by the length difference between reference summaries and candidate summaries. In this chapter, we mitigate against this issue by discarding dataset instances where the reference summary is too short or too long (see § 3.3.1) and by employing a sentence selector that effectively maintains candidate summary length within a tight range.

For BERTScore, reliability might be impacted by the quality of semantic representation of domain-specific content words. In preliminary experiments, we employed SciBERT (Beltagy et al., 2019) as base model, given that SciBERT was pretrained over scientific text. The system ranking obtained was the same as the one obtained using RoBERTa (Liu et al., 2019), and hence we report the latter in this chapter.

[3] HuggingFace checkpoint: `ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli`.

[4] HuggingFace checkpoint: `shiyue/roberta-large-tac08`.

3.3.5 LLM Agent Evaluation

In addition to quantifying coverage and informativeness using standard summarization metrics, we generate critiques stating an LLM agent's preference w.r.t. content coverage. We employ Auto-J Li et al. (2023) to compare candidate summaries in pairwise and single-response setups. We compared the best FangKvD configurations in each `WM` regime, as well as baselines NoTree, TextRank, and PacSum[SciBERT], labeled as simply PacSum, for a subset of 100 samples taken uniformly randomly from the test set of the analyzed datasets.

In the pairwise setup, we use a ranking protocol, commonly used in human evaluation studies (Wu and Hu, 2018; Luo et al., 2019), to aggregate preferences, i.e. summaries with 'Win' labels are assigned a rank of 1 and those with 'Lose', a rank of 2, whereas both candidates are assigned a rank of 1 when tied. All system pairs were compared and the final rank of a system is defined as the average rank over all of its comparisons.

In the single-response setup, we report the final ranking assigned by Auto-J on a scale of 1 to 10. For all our generations, we employ the HuggingFace checkpoint `GAIR/autoj-scenario-classifier`, which is in turn fine-tuned from `meta-llama/Llama-2-13b-chat-hf` (Touvron et al., 2023b). Table 3.2 showcases the complete prompt template employed in our experiments in both generation setups.

3.3.6 Human Evaluation

Finally, human judgment is elicited to evaluate the degree to which the proposed systems capture key content in a scientific article. To this end, we employ a question-answering (QA) paradigm (Clarke and Lapata, 2010; Narayan et al., 2018b, 2019) with Cloze style queries instead of factoid questions (Hermann et al., 2015). Human subjects on Amazon Mechanical Turk (AMT) were presented with a system summary and a query, and asked to write down the answer to the query. Queries are constructed by replacing all occurrences of one factual detail in the reference (gold) summary with an 'X'.

We sampled 50 documents from the test set of PubMed and manually constructed 3 Cloze queries per document, for systems Oracle, Sub-Exp, NoTree, and PacSum[SciBERT]. System `Sub-Exp` with tree size 20 is chosen because it had the highest sum of Rouge-1 and Rouge-2 scores. Oracle is included because it gives us an upper-bound as to how much information can be captured in the optimal scenario. Please refer to Appendix B.1 for more details on the AMT campaign.

After collecting answers, we proceed to score them as follows. We expand the partial-matching scoring system proposed by (Clarke and Lapata, 2010) with fine-grained error cat-

Pairwise-Response Setup
[INST] You are assessing two submitted responses on a given user's query and judging which response is better or they are tied. Here is the data: [BEGIN DATA] *** [Query]: Which response covers more content from the following reference text? Reference: {reference summary} *** [Response 1]: {system summary A} *** [Response 2]: {system summary B} *** [END DATA] Here are the instructions to assess and compare the two responses: 1. Pinpoint the key factors to distinguish these two responses. 2. Conclude your comparison by providing a final decision on which response is better, or they are tied. Begin your final decision statement with "So, the final decision is Response 1 / Response 2 / Tie". Ensure that your decision aligns coherently with the comprehensive evaluation and comparison you've provided. [/INST]
Single-Response Setup
[INST] Write critiques for a submitted response on a given user's query, and grade the response: [BEGIN DATA] *** [Query]: How much content does the response cover from the following reference text? Reference: {reference summary} *** [Response]: {response} *** [END DATA] Write critiques for this response. After that, you should give a final rating for the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]". [/INST]

Table 3.2: Complete prompt template for generating coverage critique using Auto-J, for pairwise comparison setup and single-response setup.

egories, showcased in Table 3.3. These categories aim to provide a better picture of the semantic relation between human answers and reference answers. Each human answer is manually labeled by the authors into one of these error categories and receives the score associated with it. Then, the performance of a system is defined as the average score over all the answers for that system. Statistical significance is tested using a one-way ANOVA ($p < 0.01$) with posthoc Tukey-HSD tests and 95% confidence interval.

3.4 Results and Discussion

In this section, we report and discuss our findings from our automatic and human experiments on modeling content relevance using the proposed KvD reader configurations.

3.4.1 Content Selection at the Sentence Level

We start by analyzing the performance of our systems at selecting relevant sentences from the input document. Relevance and coverage metric scores are showcased in Table 3.4 and

Category	Score	Description	Example
Exact Match	1.0	Exact string match with reference answer.	*R: infected macrophages* *A: infected macrophages*
Synonymity	0.8	Answer is a synonym or a rephrase of reference answer.	*R: infected macrophages* *A: contaminated macrophages*
Specificity	0.6	Answer is a hypernym of reference answer.	*R: temporal lobe epilepsy* *A: seizures*
Incompleteness	0.3	Answer phrase is missing words	*R: anaphylaxis diagnosis and management* *A: anaphylaxis diagnosis*
Incorrectness	0.0	Answer is totally unrelated.	*R: biomarker* *A: measure*
Not found	0.0	Answer could not be found on the provided summary.	-

Table 3.3: Categories of human answers for the campaign.

Table 3.5 for PUBMED and ARXIV, respectively. Results are grouped by working memory parameter, showing the best and worst FANGKVD system configuration chosen according to their average ROUGE recall scores ((R1-R+R2-RL-R)/3) in the corresponding validation set.

Statistical significance at the system level is tested pairwise using Bootstrap with a 95% confidence interval. For PUBMED, Table 3.4, we found no pairwise statistical difference between R1-R scores of systems `Lvl-Exp[5]`, TEXTRANK), and `Sub-Exp[20]`). Similarly, no pairwise difference between `Lvl-Exp[50]`, `Lvl-Exp[100]`, and NOTREE. For ARXIV, Table 3.5, no pairwise statistical difference in R1-R scores was found between systems `Cnt-Wgt[5]`, `Cnt-Cnt[50]`, and `Cnt-Cnt[50]`; and between systems `Lvl-Exp[100]`, RANDOM-WGT, and NOTREE. Analogously, Table 3.4 and Table 3.5 indicate system groups in which no pairwise difference was found, one group per marker, for each metric reported.

FangKvD Configurations. First, when comparing amongst FANGKVD systems, we find that for every `WM` setup, the worst system belongs to a configuration that only uses proposition frequencies in the memory trees (c_{cnt}) and not properties of the tree shape. In contrast, all the best FANGKVD systems employed configurations that score proposition occurrences by exploiting tree shapes (c_{sub}, c_{deg}, c_{lvl}).

In PUBMED, Table 3.4, subtree size-based aggregation yielded comparable informativeness and n-gram coverage in terms of ROUGE recall scores for working memory capacities of 5 and 20 (`Sub-Cnt` and `Sub-Exp`), while BERTScore recall favours `WM` $= 5$. In terms of reference STU coverage, we observe comparable performance of all FangKvD systems according to p^{2c} and l^{3c}, with `Sub-Cnt` exhibiting slightly higher coverage.

Similarly, in arXiv, Table 3.5, `Lvl-Exp` with $\texttt{WM} = 20$ outperforms other configurations in terms of ROUGE recall scores. However, `Cnt-Cnt` with $\texttt{WM} = 50$ shows higher semantic coverage, as indicated by BERTScore recall and p^{2c} scores. It is also worth noting that, for both datasets, p^{2c} shows higher entailment scores than l^{3c} across the board, confirming the usefulness of finetuning an NLI model on the STU presence task (Zhang and Bansal, 2021).

Second, when comparing metrics scores across varying levels of working memory capacity, we found a non-linear relationship between them, indicating that indeed there exists a memory capacity optimal for content coverage modeling. We hypothesize that a smaller memory tree forces the KvD reader to keep only the most relevant nodes at each memory cycle, but too-small a buffer risks discarding potentially relevant nodes. In larger memory trees, instead, more propositions get to accumulate scores during simulation, hence making -potentially irrelevant- longer sentences obtain higher scores. In fact, a closer look at the summary lengths revealed that the average number of sentences per summary decreased as the `WM` parameter increased, for both datasets.

Note, however, that this memory capacity effect seems to only hold for arXiv, whereas PubMed shows consistently lower ROUGE scores as the capacity is incremented. This could be due to the slight difference in domain between both datasets, realized in the difference between section lengths. A closer inspection revealed that sections in PubMed articles are 16.8 sentences long on average, whereas sections in arXiv, 28.8. Given that the memory tree is flushed at the beginning of each section during reading, processing a short section with a large `WM` capacity would allow the model to keep all propositions read so far in memory. This means that potentially irrelevant content units are never discarded from working memory. In PubMed, the sections seem to be short enough for this phenomenon to happen, whereas sections in arXiv are long enough to always prompt the model to discard units while reading.

Unsupervised Baselines. Regarding standard heuristic baselines and unsupervised baselines, we observed the following. First, we observe that the organization of information in scientific articles poses a challenge for trivial baselines. For instance, selecting the first sentences of the introduction section (Lead) performs worse than randomly picking sentences (Random and Random-Wgt). Second, note that all FangKvD systems perform better in both datasets than baseline NoTree, which ranks propositions according to their frequency in the document. This result highlights the importance of using the memory trees to model the relevance of content units, regardless of the memory capacity or aggregation strategy employed.

Next, we turn our attention to baselines modeling content relevance through sentence node centrality. In both datasets, LexRank and PacSum[TF-IDF] showed poor performance on all metrics, even under-performing random baselines. In PubMed, TextRank outperforms PacSum[SciBERT] on all metrics, and even slightly outperforms the best FangKvD system, Sub-Cnt with $\mathtt{WM} = 5$, in terms of n-gram coverage according to ROUGE recall. Regarding semantic coverage, however, TextRank and Sub-Cnt perform comparably according to BERTScore recall, p^{2c}, and l^{3c} scores. In contrast, in arXiv, TextRank and PacSum[SciBERT] do lag behind all best FangKvD systems on all metrics. Lastly, regarding the choice of sentence representation in PacSum, using dense embeddings from a pretrained neural model (PacSum[SciBERT]) improves upon –as expected– a sparse representation (PacSum[TF-IDF]) in both datasets, although not enough to catch up to TextRank.

These rather mixed results indicate again the impact of restricting the update of scores in each processing iteration. For FangKvD systems, each iteration corresponds to a memory cycle and the update is limited to content in working memory. Whereas for baselines using PageRank to calculate node centrality, the score of all nodes (in this case, sentences) in a document graph is updated in each iteration, until convergence. Similarly, PacSum systems calculate degree centrality using not only previous but also future sentences when processing each sentence in turn. In contrast, our FangKvD systems do not look forward when scoring content and instead, a memory cycle is only allowed to look at content in working memory and in long-term memory if needed –both consisting of already processed content.

Supervised and Non-Completely Unsupervised Baselines. Finally, consider the supervised upper-bound in our analyses, Ext-SciBERT, and the fine-tuned PacSum-FT*. As expected, finetuning on in-domain data dramatically improves PacSum's capacity to detect relevant content –although still falling behind Ext-SciBERT–, with the improvement being striker for arXiv. In PubMed, PacSum-FT outperforms our best FangKvD system in terms of ROUGE and BERTScore; however, both systems perform comparably in terms of semantic coverage scores p^{2c} and l^{3c}. In contrast, in arXiv, Lvl-Exp gets the upper-hand in all metrics.

3.4.2 Content Selection at the Proposition Level

In this analysis, we move on from quantifying coverage at the sentence level to evaluating coverage at the proposition level using Equation 3.15. We compared FangKvD systems using

System	WM	R1-R	R2-R	RL-R	BSc-R	p^{2c}	l^{3c}
Sub-Cnt	5	**44.10**†	14.50†	39.39‡	**84.43**†	**0.25**	**0.15**†
Cnt-Wgt	5	43.35	13.65‡	38.55	84.28‡	0.24	0.13
Sub-Exp	20	44.00†	14.70†	39.40‡	84.28‡	0.23‡	0.14‡
Cnt-Wgt	20	42.90	13.44‡	38.18§	84.19	0.23‡	0.14‡
Lvl-Exp	50	43.51‡	13.99‡	38.81	84.29‡	0.24†	0.14‡
Cnt-Cnt	50	42.75	13.30	38.07§	84.10	0.23‡	0.14
Lvl-Exp	100	43.20‡	13.46‡	38.48	84.31‡	0.24†	**0.15**†
Cnt-Cnt	100	42.72	13.18	37.99§	84.05	0.23	0.13
Lead		41.12	13.36	36.72	83.90	0.24	0.13
Random		42.91	13.06	38.04	84.19	0.23	0.10
Random-Wgt		42.60	12.71	37.73	84.33	0.22§	0.11
NoTree		43.20‡	13.27	38.51	84.25	0.22§	0.12
TextRank		**44.10**†	**14.77**†	**40.45**†	84.42†	0.24†	**0.15**†
LexRank		43.10‡	13.82‡	39.85†	84.20	0.22§	0.13
PacSum[TF-IDF]		36.46	11.53	32.84	83.43	0.20	0.10
PacSum[SciBERT]		42.87	13.74	39.52	84.32	0.23	0.12
PacSum-FT[SciBERT]*		45.81	16.36	41.04	84.51	0.25	0.15†
Ext-SciBERT		47.16	17.37	42.88	84.68	0.26	0.19
Ext-Oracle		60.08	28.74	54.46	87.27	0.35	0.28

Table 3.4: Summary informativeness in terms of ROUGE recall (R1-R, R2-R, RL-R), BERTScore recall (BSc-R), and coverage in terms of Lite^{3}Pyramid's p^{2c}, l^{3c}, over the test set of PubMed. Best (top) and worst (bottom) FangKvD systems are presented for each memory capacity WM. (†,‡,§): no statistical difference between system pairs in the same column. Best completely unsupervised systems are **bolded**.

occurrence scorers that exploit a tree property (strategies c_{sub}, c_{deg}, and c_{lvl}) against those using only frequency (c_{cnt}), for tree size $\mathtt{WM} = 20$. For completeness, baseline NoTree was also included. Same as in previous analyses, statistical significance was tested pairwise using the Bootstrap test with a 95% confidence interval.

In PubMed, Lvl-Exp obtained an F_1 score (%) of 29.80, closely followed by Sub-Exp (29.79) and Deg-Exp (29.76). In contrast, FangKvD systems using only frequency,

System	WM	R1-R	R2-R	RL-R	BSc-R	p^{2c}	l^{3c}
Deg-Wgt	5	50.11	16.33†	44.10	84.69†	0.23	**0.12**†
Cnt-Wgt	5	49.53	15.54	43.44‡	84.66	0.23	0.11‡
Lvl-Exp	20	**51.04**	**16.94**‡	**44.87**†	84.58§	0.24‡	**0.12**†
Cnt-Cnt	20	49.66†	16.11†	43.53‡	84.57§	0.24‡	0.11‡
Cnt-Cnt	50	49.57†	15.98	43.62‡	**84.72**†	**0.26**	**0.12**†
Cnt-Wgt	50	47.75	14.16	41.81§	84.51	0.25†	0.11‡
Lvl-Exp	100	48.69‡	15.02	42.78	84.64†	0.25†	0.11‡
Cnt-Wgt	100	47.71	14.51§	41.94§	84.55§	0.25	0.10§
Lead		44.91	12.52	39.08	83.85	0.17	0.07
Random		47.81	14.50§	41.66§	84.51	0.19§	0.09§
Random-Wgt		48.25‡	15.06	41.98§	84.52	0.19§	0.10§
NoTree		48.81‡	14.28§	42.45	84.20	0.22	0.09§
TextRank		47.27	16.20†	42.37	84.19	0.24‡	0.11‡
LexRank		41.91	11.55	37.85	83.83	0.16	0.07
PacSum[TF-IDF]		38.87	12.01	33.40	83.79	0.19§	0.05
PacSum[SciBERT]		44.19	13.38	39.88	84.04	0.20	0.08
PacSum-FT[SciBERT]*		50.05	16.70‡	44.36†	84.03	0.24	0.07
Ext-SciBERT		53.32	18.93	48.01	84.97	0.27	0.12†
Ext-Oracle		70.08	35.35	62.82	87.92	0.38	0.25

Table 3.5: Summary informativeness in terms of ROUGE recall (R1-R, R2-R, RL-R), BERTScore recall (BSc-R), and coverage in terms of Lite^{3}Pyramid's p^{2c}, l^{3c}, over the test set of arXiv. Best (top) and worst (bottom) FangKvD systems are presented for each memory capacity WM. (†,‡,§): no statistical difference between systems in the same column. Best completely unsupervised systems are **bolded**.

Cnt-Exp (29.62) and NoTree (27.15), seem to capture fewer oracle propositions, although they do not fall far away behind. Similar trends were observed in arXiv, for systems Sub-Cnt (23.51), Deg-Cnt (23.49), and Lvl-Exp (23.11), followed by baselines Cnt-Cnt (21.28) and NoTree (21.55).

In both datasets, scores from tree-informed systems (Sub-Exp, Deg-Exp, Lvl-Exp) were found to be statistically different, pairwise, from scores from systems not informed by tree properties (`Cnt-Exp`, `Cnt-Cnt`, NoTree).

It is also worth noting that the proportion of oracle propositions captured by the systems is low ($23-30\%$). Preliminary experiments showed that, even though larger memory limit setups capture a larger number of oracle propositions (around 70% for $\texttt{WM} = 100$), more noise is also scored higher, hence making sentence selection harder.

3.4.3 LLM Agent Critique on Coverage

Next, we asked an LLM agent, specially fine-tuned to provide preference critiques, to judge the coverage in candidate summaries w.r.t. reference summaries. We start our analysis with results in both pairwise and single-response ranking, aggregated and averaged by system. System rankings in both setups were tested pairwise for statistical significance using a one-way ANOVA ($p < 0.01$) with posthoc Tukey-HSD tests and 95% confidence interval. Finally, we analyzed the distribution of coverage judgment categories to gain further insight into the agent preferences in the pairwise setup.

Pairwise and Single-Response Ranking. Table 3.6 showcases the ranking results for pairwise and single-response setups. In the pairwise setup, the difference in ranking scores of systems `Sub-Cnt[5]`, TextRank, PacSum, and NoTree was found statistically significant from all other systems; as well as the difference between `Sub-Exp[20]` and`Lvl-Exp[100]`, with all other pairs were found to be not significant. In the single-response setup, the difference between the following system pairs was found significant: (`Sub-Cnt[5]-Lvl-Exp[100]`), (TextRank-`Lvl-Exp[100]`), as well as PacSum and NoTree against all other systems.

In PubMed, Auto-J preferred `Sub-Cnt[5]`, with pairwise preference decreasing as `WM` capacity increases. However, single-response ranking seems to increase with `WM`, a trend similar to p^{2c} and l^{3c} which quantify coverage at the semantic triplet level. Note also that TextRank is ranked second to `Sub-Cnt[5]` in pairwise preference but at the same rank as the least preferred FanKvD system in single-response ranking. Finally, baselines PacSum and NoTree are significantly less preferred than FangKvD systems in both setups.

In arXiv, we find that `Lvl-Exp[20]` is slightly more preferred pairwise among FangKvD systems, whereas `Lvl-Exp[50]` is more preferred in the single-response setup. Again, single-response ranking seems to correlate with coverage at the semantic triplet level. Note that NoTree ranks better than TextRank in the pairwise setup and comparably in the single-response setup. Moreover, both systems rank comparably to the least preferred FangKvD system in the single setup. However, along with PacSum, these baselines are again less preferred than FangKvD systems in the pairwise setup.

System	PubMed		System	arXiv	
	Pairwise↓	Single↑		Pairwise↓	Single↑
Sub-Cnt[5]	**1.25**(0.43)	3.71(0.57)	Deg-Wgt[5]	1.23(0.42)	3.40(0.53)
Sub-Exp[20]	1.40(0.49)	3.68(0.63)	Lvl-Exp[20]	**1.22**(0.42)	3.45(0.67)
Lvl-Exp[50]	1.42(0.49)	3.77(0.58)	Cnt-Cnt[50]	1.23(0.42)	**3.52**(0.57)
Lvl-Exp[100]	1.45(0.50)	**3.80**(0.55)	Lvl-Exp[100]	1.27(0.45)	3.33(0.62)
TextRank	1.31(0.46)	3.71(0.77)	TextRank	1.68(0.47)	3.40(0.60)
PacSum	1.52(0.50)	2.76(0.68)	PacSum	1.96(0.20)	3.28(0.80)
NoTree	1.82(0.38)	2.84(0.61)	NoTree	1.65(0.48)	3.39(0.52)

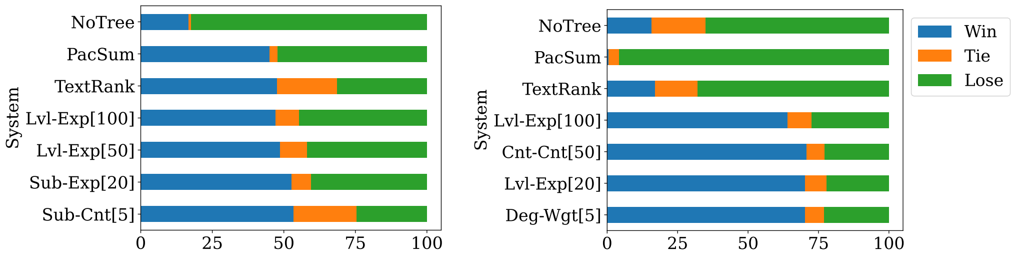
Table 3.6: Preference ranking of content coverage by Auto-J when comparing system summaries pairwise (Pairwise) and as standalone (Single) responses, for a subset of the test set in PUBMED (left) and ARXIV (right). WM capacity is shown in squared brackets, and standard deviation is shown in parentheses. (↑,↓): higher and lower is better, respectively; best system in **bold**.

Pairwise Judgement Categories. To gain further insights into the critiques of Auto-J, we analyzed the distribution of judgment categories, 'Win', 'Tie', and 'Lose', showcased in Figure 3.3. On the one hand, in PUBMED, we observed that all FangKvD systems have comparable win rates, although slightly decreasing as WM capacity increases. Notably, Sub-Cnt[5] presents a significantly higher tie rate (and hence, a lower lose rate) than the other FangKvD systems, indicating that WM capacity has a direct impact on the coverage of the extracted summaries. This insight further adds evidence to our finding in the preceding sections, indicating that a lower WM capacity allows the system to keep less noise (potentially irrelevant information) in the memory tree. Note also that TextRank presents a similar distribution to Sub-Cnt[5] but with a lower win rate.

On the other hand, in ARXIV, we observed similar category distributions for all FangKvD systems, with LVL-EXP[100] showing a slightly lower win rate. Additionally, baselines TextRank and NoTree exhibit similar distributions, whereas PacSum shows an extremely low win rate.

3.4.4 Human Evaluation and Error Analysis

Our Cloze QA evaluation obtained average accuracies of 72.78, 64.96, 62.58, and 57.87 (percentual points) for systems ORACLE, SUB-EXP, NOTREE, and PACSUM, respectively.



a PubMed

b arXiv

Figure 3.3: Distribution of coverage judgment categories (Win, Tie, and Lose) by Auto-J, normalized by system, for pairwise response comparison in PubMed (left) and arXiv (right).

All pairwise system comparison were found statistically significant using a one-way ANOVA ($p < 0.01$) with posthoc Tukey-HSD tests and 95% confidence interval.

Regarding the error categorization, we found that, from a total of 1800 answers, 976 were exact matches, 110 were categorized as Synonymity, 74 as Specificity, 178 as Incompleteness, 307 as Incorrectness, and 155 as Not found, among all evaluated systems.

Figure 3.4 provides a closer look into the distribution of answer categories per system. This reveals that all systems obtain a comparable proportion of answers in Synonymity (between 5.5 and 6.4%) and Incompleteness (between 9.1 and 10.4%). Incorrect answers are found to be high for NoTree (23.5%) and Sub-Exp (19.7%), followed by Oracle (15.1%) and PacSum (9.7%). Most notably, we find that Sub-Exp obtains significantly more Specificity answers (7.3%) than the closest system, NoTree (4%).

Two observations can be made from these results. First, we see that Oracle and PacSum are more reserved, precision-oriented systems given that they presented less non-exact-match answers. However, PacSum summaries led to 103 empty or `not-found` answers, hence the low accuracy of the system. In contrast, we find that Sub-Exp and NoTree are more lenient, recall-oriented systems, given their higher number of non-exact-match answers. However, NoTree summaries did not provide the appropriate content, hence the higher number of incorrect answers.

Second, we observe that Sub-Exp summaries provided sentences with synonyms in the same proportion as Oracle did (6.4%) and led to fewer cases of incomplete content (9.1%). In cases where sentences with the exact-match answer were not extracted, Sub-Exp provided instead with less specific but relevant content, hence the higher number of Specificity answers.

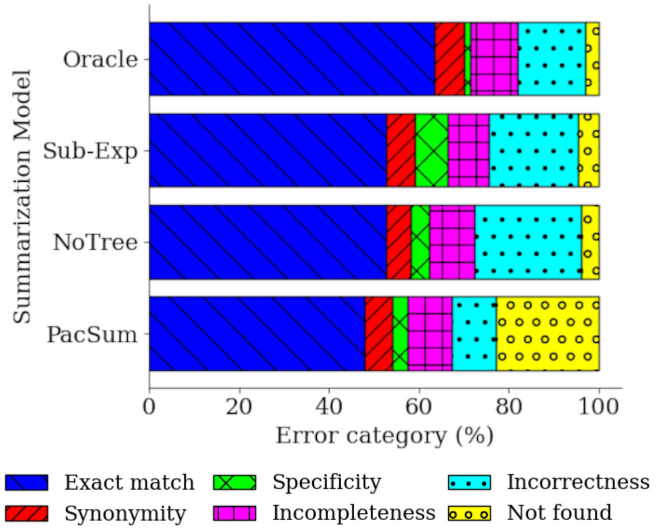


Figure 3.4: Distribution of error categories in answers of human evaluation, aggregated by evaluated system.

3.4.5 Summary Length Control

Next, we analysed the effectiveness of our Knapsack selector on controlling summary length distribution. We compared the length distributions of FangKvD systems when using a greedy selector against using our Knapsack selector.

Figure 3.5 presents the distribution of reference summaries and the best FangKvD configurations at WM $= 20$ for each validation dataset. For PubMed, we show Sub-Exp and for arXiv, Lvl-Exp. When using the Knapsack selector, summary lengths are heavily concentrated around the predefined budget, in both datasets. In contrast, summaries extracted by a greedy selector exhibit lengths going way over the budget.

Furthermore, Table 3.5 presents the mean and standard deviation of these length distributions, as well as their corresponding informativeness performance in terms of average ROUGE-F$_1$ score. As mentioned before, previous work (Narayan et al., 2018b; Schumann et al., 2020) pointed out that ROUGE scores are sensitive to summary length, with longer summaries potentially inflating scores. Nevertheless, Narayan et al. (2018b) reported that ROUGE-F$_1$ is less sensitive to summary length, although its effect is still non-trivial. For this reason, we opt for reporting F$_1$ for this analysis instead of recall scores.[5]

The following insights can be drawn from these results. First, our Knapsack selector extracts summaries with a mean length much closer to that of the reference summaries and with a much lower standard deviation, compared to the greedy selector, in both datasets.

[5] ROUGE recall is best for evaluation under fair comparison setups, e.g. when summary length is being controlled for.

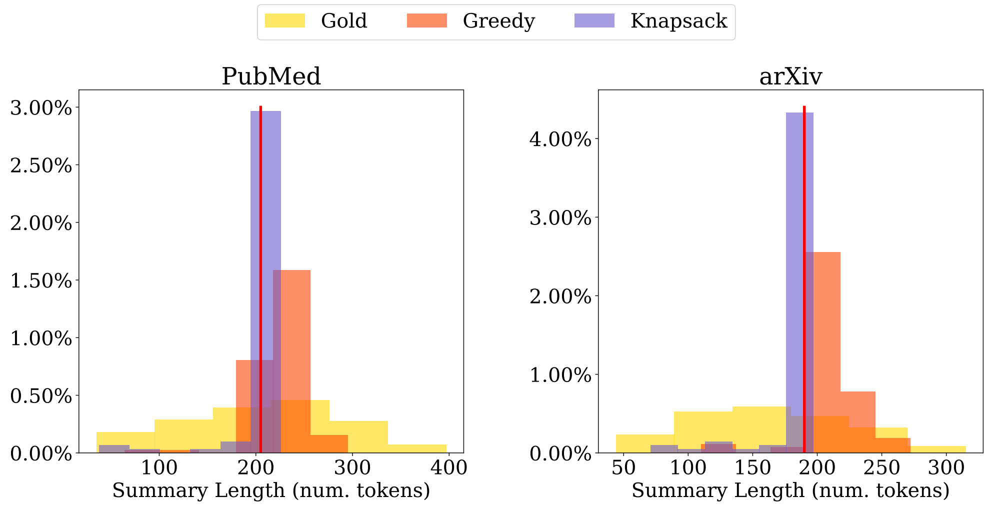


Figure 3.5: Distribution of summary lengths (in number of tokens) in reference summaries (Gold) and our best FANGKVD systems with $\texttt{WM} = 20$, extracted using our Knapsack selector (Knapsack) and a greedy selector (Greedy), for PUBMED (left) and ARXIV (right) validation sets. The budget used is shown as a vertical red line.

Selector	**PubMed**			**arXiv**		
	$\mu_{\lvert S \rvert}$	$\sigma_{\lvert S \rvert}$	avg. R-F_1	$\mu_{\lvert S \rvert}$	$\sigma_{\lvert S \rvert}$	avg. R-F_1
Gold	205.30	67.11	-	179.32	49.60	-
Greedy	224.61	20.46	28.76	208.04	22.86	29.45
Knapsack	199.34	8.16	28.69	185.47	16.25	28.68

Table 3.7: Performance of sentence selectors in terms of average ROUGE-F_1 score (avg. R-F_1), as well as mean ($\mu_{|S|}$) and standard deviation ($\sigma_{|S|}$) of the length distributions of their respective extracted summaries.

One immediate benefit from these properties is that ROUGE scores are much more reliable than scores obtained using a greedy selector. Second, the greedy selector obtains slightly higher R-F_1 scores (less than 0.8 points of difference). This can be attributed to the significantly longer summaries it extracts, which ultimately inflates the ROUGE scores.

3.4.6 Qualitative Analysis

Finally, we present a qualitative analysis of system summaries to determine the extent they cover summary content units in the corresponding gold summary. Figure 3.6 presents the abstract (gold summary) of an article taken from PUBMED, along with summaries extracted by TextRank, PacSum[SciBERT], and FangKvD system `Sub-Exp` with `WM` = 20. Each system summary is presented along its respective ROUGE recall (R1-R and R2-R) and p^{2c} scores. Clauses covering the same gold summary content are coloured the same.

First, it is important to note that content coverage w.r.t. the reference summary is low across all systems: roughly, content from over a quarter of reference sentences were covered by any system summary, which could explain the rather low bigram coverage (R2-R) and semantic triplet coverage (p^{2c}) scores. Nevertheless, `Sub-Exp` exhibits the highest coverage among all systems, with 5 out of its 7 sentences covering content in the reference, compared to a surprising 2/8 and 2/9 in TextRank and PacSum[SciBERT], respectively. The higher content coverage of `Sub-Exp` is also reflected in its higher ROUGE-2 recall (R2-R) and p^{2c} scores. Upon closer inspection, we noticed that the extracted sentences are in close vicinity of each other. This behaviour can be explained by the tendency of memory trees to retain propositions reflecting the topic being discussed at certain point during the reading simulation.

When comparing TextRank and PacSum[SciBERT], we observe the following. First, both system summaries show relatively high unigram coverage (R1-R), even close to `Sub-Exp`. This can be attributed to their coverage of relevant tokens (e.g. 'temperature', 'productivity', 'Nicaragua') in otherwise less informative sentences. Second, although both summaries seemingly show the same degree of coverage (two spans each), TextRank scores higher ROUGE and p^{2c} scores. We hypothesize that this difference is due to one content span in TextRank (*relationship between...*,) having more lexical overlap with its correspondent span in the reference (*Heat stress also has...*). In contrast, one content span in PacSum[SciBERT] (*This was part of a...*) is found heavily rephrased int the reference (*A first assessment...*). Such difference in metric values further highlights the limitations of current automatic evaluation methodologies. Lastly, is it worth noting that these systems cover similar topics, such as humidity information, recommendations, and the potential benefits of implementing these recommendations. These topics, however, are not covered in the reference summary.

3.5 Summary

We considered the problem of content selection in unsupervised extractive summarization, experimenting with long, structured documents–scientific articles in two datasets. We explored a wide variety of system configurations that exploit properties of tree structures of content units as modeled by a psycho-linguistic model of reading comprehension, KvD

(Kintsch and van Dijk, 1978). Our methodology included a specialized sentence selector capable of maintaining summary length tight to a predefined budget, hence ensuring a fair comparison at the system level. Results showed that systems leveraging tree properties, such as node depth and size of subtree, perform better than systems using plain frequency counts according to automatic metrics of informativeness and content coverage.

Furthermore, human evaluations and error analysis of human answers revealed that our system preferred to provide less specific yet relevant content rather than content not relevant at all, a behaviour confirmed by a thorough qualitative analysis.

Gold

Background. Heat illness is a major cause of preventable morbidity worldwide. Workers exposed to intense heat can become unable to activate compensation mechanisms, putting their health at risk. Heat stress also has a direct impact on production by causing poor task performance and it increases the possibility of work-related morbidity and injuries. During the sugarcane harvest period, workers are exposed to excessive sunlight and heat from approximately 6 am to 3 pm. A first assessment of heat stress during the 2006/2007 harvesting season served to redesign the existing rehydration measures. In this project, sugarcane workers were provided with more rehydration solutions and water during their work schedule. Objective. To assess heat stress preventive measures in order to improve existing rehydration strategies as a means of increasing productivity. Methods. A small group of 22 workers was followed up for 15 days during working hours, from 6 am to 3 pm. Selection criteria were defined: to have worked more than 50% of the day's working schedule and to have worked for at least 10 days of the follow-up period. A simple data recollection sheet was used. Information regarding the amount of liquid intake was registered. Production output data were also registered. Temperature measurements were recorded using a portable temperature monitoring device (EasyLog, model EL-USB-2). Results. The average temperature measurements were above the Nicaraguan Ministry of Labour thresholds. Seven workers drank 78L of liquid, improving their production. Output production increased significantly (p=0.005) among those best hydrated, from 5.5 to 8 tons of cut sugarcane per worker per day. Conclusions. Productivity improved with the new rehydration measures. Awareness among workers concerning heat stress prevention was increased.

TextRank (R1-R$=29.50$, R2-R$=4.33$, $p^{2c}=0.13$)

Historically, monitoring of toxins in the work environment has been the primary focus for identifying risks. However, the INETER humidity data are quite different from the relative humidity registered on the farm located in western Managua. Temperature data were not available at INETER's website. Other authors have shown the relationship between heat stress health effects and the ability to perform different tasks, as well as the increased risk of suffering work-related injuries (14). In this study, the workers drank more liquid as temperature values increased to maximum peaks. Since dehydration reduces the capacity for absorption from the gut, workers must be educated regarding the importance of drinking enough water during work and continuing generous rehydration during off-duty hours (14). Certainly, more effort in terms of intervention strategies and scientific investigation needs to be carried out among workers in Nicaragua who perform jobs in which they are exposed to high ambient temperatures. More funds should also be designated by companies' decision-makers for improving basic working conditions, in order to increase overall productivity (and workers' satisfaction in terms of better wages).

PacSum[SciBERT] (R1-R$=26.62$, R2-R$=1.81$, $p^{2c}=0.04$)

However, the INETER humidity data are quite different from the relative humidity registered on the farm located in western Managua. Temperature data were not available at INETER's website. In this study, the workers drank more liquid as temperature values increased to maximum peaks. This was part of a rehydration process that was well planned in advance by the company's decision-makers. The basis of this principle is that drinking to satisfy thirst is not enough to keep a person well-hydrated. Some of the reasons for this can be attributed to their low educational level, and feeling that nothing bad has ever happened to me before, etc. These include farm workers, construction workers, miners, and fishermen, especially those employed in the informal sector, which occupies about half of Nicaragua's economically active population. More funds should also be designated by companies' decision-makers for improving basic working conditions, in order to increase overall productivity (and workers' satisfaction in terms of better wages). This would also translate into safer and healthier workers, less absenteeism from sick leave, fewer accidents, and other incidents.

FangKvD, `Sub-Exp` `WM` $=20$ (R1-R$=30.22$, R2-R$=6.50$, $p^{2c}=0.16$)

Historically, monitoring of toxins in the work environment has been the primary focus for identifying risks. Some potential biomarkers linked to cell injury are immunological factors, lymphokines, growth factors, prostaglandins, endothelins, collagen, adhesion molecules, thromboxanes, leukotrienes, platelet-activating factors, and heat shock proteins (10). As mentioned earlier, heat illness is a major cause of preventable morbidity worldwide (1), and although human beings possess considerable ability to compensate for naturally occurring heat stress, many occupational environments and/or physical activities expose workers to heat loads that are so excessive as to threaten their health and productivity (11). Although only 22 subjects were followed up for a short period of time in this study, important results were obtained. Other authors have shown the relationship between heat stress health effects and the ability to perform different tasks, as well as the increased risk of suffering work-related injuries (14). In this study, the workers drank more liquid as temperature values increased to maximum peaks. This was part of a rehydration process that was well planned in advance by the company's decision-makers.

Figure 3.6: Gold summary (abstract of the article) and summaries extracted by TEXTRANK, PACSUM[SCIBERT], and `Sub-Exp` with `WM` $=20$, for an article taken from PUBMED. Text spans mentioning the same content as the gold summary are colored the same. Text was detokenized and truecased for ease of reading.

Chapter 4

Trade-off Control during Document Understanding

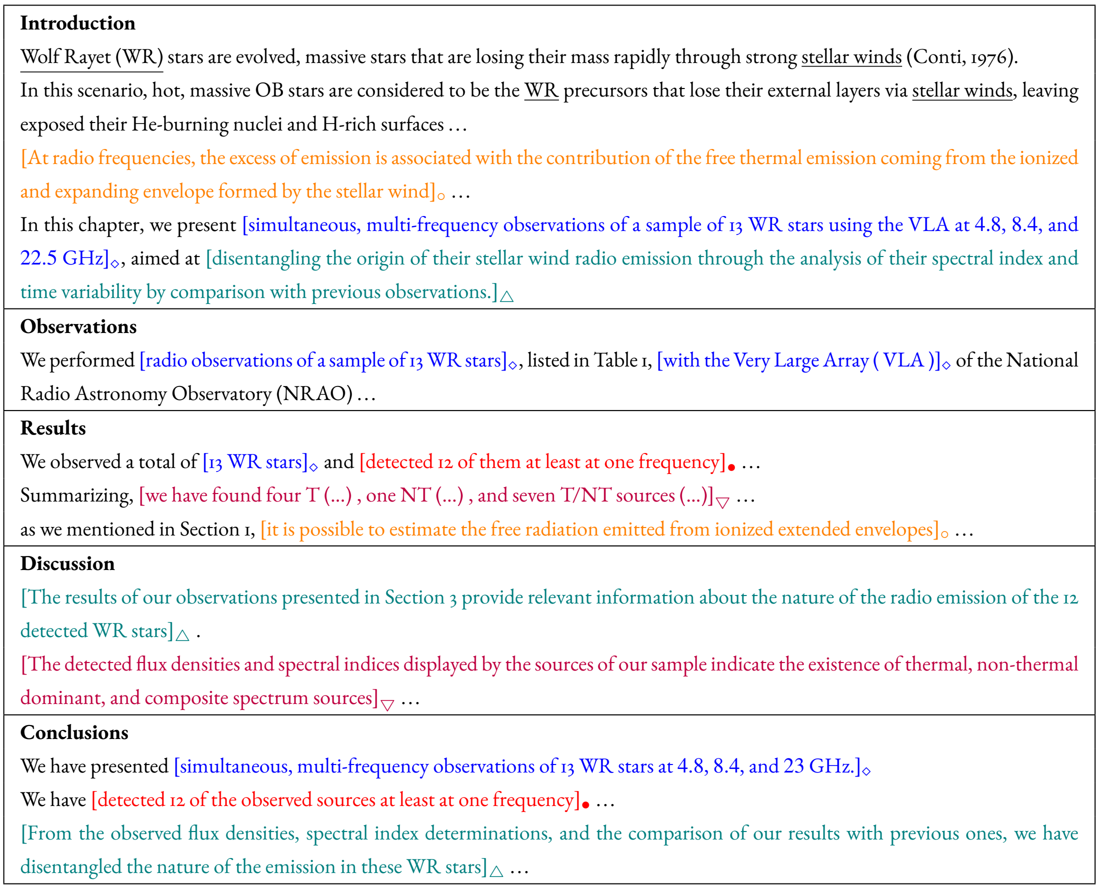
Extractive summaries are usually presented as lists of sentences with no expected cohesion between them and with plenty of redundant information if not accounted for. In this chapter, we investigate the trade-offs incurred when aiming to control for inter-sentential cohesion and redundancy in extracted summaries, and their impact on their informativeness. As case study, we focus on the summarization of long, highly redundant documents and consider two optimization scenarios, reward-guided and with no supervision. In the reward-guided scenario, we compare systems that control for redundancy and cohesion during sentence scoring. In the unsupervised scenario, we introduce two systems that aim to control all three properties –informativeness, redundancy, and cohesion– in a principled way. Both systems employ novel implementations of the KvD theory that simulate how cohesion and non-redundancy constraints are applied in short-term memory during reading. Extensive automatic and human evaluations reveal that systems optimizing for –among other properties– cohesion are capable of better organizing content in summaries compared to systems that optimize only for redundancy, while maintaining comparable informativeness. We find that the proposed unsupervised systems manage to extract highly cohesive summaries across varying levels of document redundancy, although sacrificing informativeness in the process. Finally, we lay evidence as to how simulated cognitive processes impact the trade-off between the analysed summary properties.

4.1 Introduction

As discussed in Chapter 1, the task of automatic summarization can be divided into the following three general steps: (i) discretization of the information in the source document into semantic content units and building a representation of these units, (ii) selection of content units such that they are informative to the end-user and non-redundant among themselves; and finally, (iii) production of a summary text that is coherent and cohesive. From the many variations of the summarization task investigated in recent years (Litvak and Vanetik, 2017; Shapira et al., 2017; Narayan et al., 2019; Xiao and Carenini, 2019; Amplayo et al., 2021), most extractive summarization approaches choose sentences as the indivisible content unit, assign a numerical score to each sentence, select a subset of them, and finally concatenate them into a single text to be presented as the summary.

Even though recent advances in machine learning brought promising results –mostly involving increasingly larger neural networks– in all stages of the summarization pipeline, core challenges such as redundancy (Xiao and Carenini, 2020; Jia et al., 2021; Gu et al., 2022) remain critically open. Notably, Xiao and Carenini (2020) reported that modern extractive summarization systems are prone to produce highly redundant excerpts when redundancy is not explicitly accounted for. The problem becomes particularly acute when the source document is highly redundant, i.e. information is repeated in many parts of the document. Some examples of highly redundant documents include scientific articles, and in general, long-structured documents. Consider the example in Figure 4.1 showcasing how information is repeated across sections in a scientific article. Information redundancy is characteristic of the writing style in scientific literature: the 'Introduction' section is expected to lay down the research questions addressed in the paper, each of which will be elaborated upon in the following sections, and the 'Conclusion' section (or equivalent) gathers insights and summarizes the answers to each research question.

Another open challenge in summarization –and in open text generation in general– is the production of coherent text (Sharma et al., 2019; Hua et al., 2021; Steen and Markert, 2022; Goyal et al., 2022). In particular, local coherence –the property by which a text connects semantically similar content units between neighbouring sentences– has proven challenging to capture computationally (Moon et al., 2019; Jeon and Strube, 2020, 2022) and to incorporate into the summarization task without sacrificing performance in other aspects such as informativeness (Wu and Hu, 2018; Xu et al., 2020). When the connection between adjacent sentences is not explicitly clued by linguistic units, humans resort to *inference*, the cognitive process by which prior knowledge is incorporated in order to force a connection



Introduction

Wolf Rayet (WR) stars are evolved, massive stars that are losing their mass rapidly through strong stellar winds (Conti, 1976).

In this scenario, hot, massive OB stars are considered to be the WR precursors that lose their external layers via stellar winds, leaving exposed their He-burning nuclei and H-rich surfaces . . .

[At radio frequencies, the excess of emission is associated with the contribution of the free thermal emission coming from the ionized and expanding envelope formed by the stellar wind]$_\circ$ . . .

In this chapter, we present [simultaneous, multi-frequency observations of a sample of 13 WR stars using the VLA at 4.8, 8.4, and 22.5 GHz]$_\diamond$, aimed at [disentangling the origin of their stellar wind radio emission through the analysis of their spectral index and time variability by comparison with previous observations.]$_\triangle$

Observations

We performed [radio observations of a sample of 13 WR stars]$_\diamond$, listed in Table 1, [with the Very Large Array (VLA)]$_\diamond$ of the National Radio Astronomy Observatory (NRAO) . . .

Results

We observed a total of [13 WR stars]$_\diamond$ and [detected 12 of them at least at one frequency]$_\bullet$ . . .

Summarizing, [we have found four T (...) , one NT (...) , and seven T/NT sources (...)]$_\triangledown$ . . .

as we mentioned in Section 1, [it is possible to estimate the free radiation emitted from ionized extended envelopes]$_\circ$ . . .

Discussion

[The results of our observations presented in Section 3 provide relevant information about the nature of the radio emission of the 12 detected WR stars]$_\triangle$.

[The detected flux densities and spectral indices displayed by the sources of our sample indicate the existence of thermal, non-thermal dominant, and composite spectrum sources]$_\triangledown$ . . .

Conclusions

We have presented [simultaneous, multi-frequency observations of 13 WR stars at 4.8, 8.4, and 23 GHz.]$_\diamond$

We have [detected 12 of the observed sources at least at one frequency]$_\bullet$ . . .

[From the observed flux densities, spectral index determinations, and the comparison of our results with previous ones, we have disentangled the nature of the emission in these WR stars]$_\triangle$ . . .

Figure 4.1: Sections of a scientific article taken from the ARXIV dataset showcasing information redundancy and cohesion. Repeated content is marked by text chunks with the same color and symbol, whilst consecutive sentences present cohesive phrases underlined.

and make sense of a text. A special case of local coherence, *cohesion*, makes the connection between adjacent sentences explicit by means of cohesive ties (Hassan et al., 1976) such as word repetitions, pronouns, anaphoric expressions, and conjunctions (Garrod and Sanford, 1977). Psycholinguistic research has found that cohesion improves text comprehension –the building of a mental representation of content– especially when the subjects' background knowledge is insufficient to perform inference successfully (Kintsch, 1990; Garrod and Sanford, 1994). Critically, when human subjects were asked to read a document and write a summary immediately after, higher cognitive demand during comprehension was found to severely impact the cohesion and redundancy in the produced summaries (Lehto, 1996; Kintsch and Walter Kintsch, 1998; Ushiro et al., 2013; Spirgel and Delaney, 2016).

In this chapter, we investigate the trade-offs automatic summarization systems incur on when aiming to control for redundancy and cohesion in produced summaries, and the impact on their informativeness. We focus on control strategies performed during sentence scoring, resorting to greedy selection of the top-scoring sentences until a predefined budget is met. We study the case of long, highly redundant documents from complex knowledge domains –scientific articles collected from arXiv and PubMed (Cohan et al., 2018). Two optimization scenarios are investigated, (i) when a specific summary property is optimized for under a reinforcement learning (RL) setup, and (ii) when the summary property is modeled through proxies in an unsupervised setup. The objective is to compare how properties are learned and balanced when explicit property measures are provided during training vs when no explicit measures are available.

In the RL setup, we compare systems that aim to balance informativeness and redundancy, against those that balance informativeness and local coherence, in a cohesion setup that goes beyond lexical cohesion ties. We build upon previous work that combines property-specific rewards linearly (Xiao and Carenini, 2019; Wu and Hu, 2018) and propose a model capable of combining a reward that encourages high ROUGE scores with a reward that encourages high local coherence.

In the unsupervised setup, we introduce two novel models that aim to control all three properties –informativeness, redundancy, and local coherence –specifically, lexical cohesion. These models implement the Micro-Macro Structure (KvD) theory of text comprehension (Kintsch and van Dijk, 1978), which provides a principled way of discretizing content into semantic units and organizing them in short and long-term memory. Similarly to Chapter 3, reading is performed one sentence at a time in *memory cycles*, applying constraints to a representation of working memory –a type of short-term memory– that explicitly model relevancy, non-redundancy, and cohesion among content units. In each memory cycle, relevancy is modeled by pruning working memory down to a fixed number of content units, keeping only the most relevant units read so far; cohesion, by ensuring lexical overlap between units in memory; and non-redundancy, by discarding redundant units from memory. Note that these models do not employ any reward signal and instead are completely unsupervised.

In the reward-guided scenario, extensive automatic –both quantitative and qualitative– evaluation revealed that systems optimizing for cohesion are better at organizing content in the produced summaries, compared to systems only optimizing for informativeness or redundancy. Moreover, cohesion-optimized models are able to obtain comparable –if not better– informativeness and coverage levels. In the unsupervised scenario, we found that simulated KvD reading is effective at balancing cohesion and redundancy during sentence scoring, however at the expense of reduced informativeness. Most notably, the proposed

KvD systems manage to extract highly cohesive summaries across increasing levels of document redundancy. We corroborated our findings with two human evaluation campaigns comparing our KvD systems against a strong unsupervised baseline that optimizes for cohesion. In the first study, we found that participants find KvD summaries more informative than summaries extracted with baselines based on node centrality, indicating the effectiveness of constraining working memory to keep only the most relevant units, adding evidence to our findings in Chapter 3. In the second study, annotators were able to identify significantly more cohesive links connecting sentences in KvD summaries compared to the baselines, with KvD summaries also exhibiting a smooth topic transition between adjacent or near-adjacent sentences. Finally, we lay extensive evidence as to how the simulated cognitive processes impact the trade-off between informativeness, redundancy, and lexical cohesion in final summaries.

The rest of the chapter is organized as follows. The problem formulation of the reward-guided control scenario is presented in § 4.2, followed by that of the control strategies in the unsupervised scenario in which we provide a detailed description of the KvD systems proposed (§4.3). Lastly, Sections 4.4 and 4.5 describe our experimental setup and discuss our results, respectively.

4.2 Reward-guided Control

In this section, we formulate the first scenario in which sentence scoring is guided by explicit rewards that encourage informativeness, non-redundancy, and local coherence in candidate summaries, in a reinforcement learning training setup. We posit the task of extractive summarization as the task of scoring the sentences in a document followed by a selection step in which an optimal set of sentences is chosen as the summary. The scoring step is formulated as a sequence labeling task where each sentence in a document $\mathcal{D} = \langle s_0, .., s_k, ..., s_{|D|} \rangle$ is labeled with $y_i \in \{0, 1\}$, indicating whether sentence s_i should be selected or not. A summarization system M assigns score $p(y_i = 1 \mid s_i)$ indicating the preference in selecting s_i according to a criteria modeled by M. Then, candidate summary $\hat{S}$ is obtained by concatenating the top-scoring sentences, selected greedily and with a predefined budget in number of tokens. We focus on informativeness, non-redundancy, and local coherence, as preference modeling criteria.

We build upon the model proposed by Xiao and Carenini (2020), consisting of an encoder that incorporates local and global context, a feed-forward layer as a decoder, and trained with the Cross-Entropy loss ($\mathcal{L}_{CE}$) over the sequence labeling task outlined above. In the rest of this chapter, we refer to this supervised model as E.LG.

Then, we adapt previous work on reinforcement learning-based approaches that aim to optimize for informativeness and either redundancy or local coherence. We define reward r_I, aimed at encouraging the selection of informative summaries (Dong et al., 2018), as

$$r_I = \frac{1}{3}\Big(\text{ROUGE-1} + \text{ROUGE-2} + \text{ROUGE-L}\Big),$$

where ROUGE F_1 scores are calculated using the reference summaries. Next, we provide details about the encoder modeling informativeness and define models employing policy gradient methods that maximize a reward function combining r_I with redundancy or coherence-aware rewards.

4.2.1 Informativeness Encoder

We employ the model proposed by (Xiao and Carenini, 2019) optimized to encode only informativeness during sentence scoring. The model incorporates local and global information by taking into account the document structure (e.g. section separation) and The model, which we label E.LG in this chapter, consists of a document encoder and a decoder that classifies whether a sentence should be selected or not.

Document Encoder. Given document $\mathcal{D} = \langle s_0, .., s_k, ..., s_{|D|}\rangle$, where s_i is a sequence of tokens, sentence embedding h_i, is defined as the average token embedding of its constituent tokens. Then, global sentence representations are obtained using a bi-directional RNN (Schuster and Paliwal, 1997) with GRU cells (Cho et al., 2014), i.e. $h^g_i = [f_i, b_i]$, where f_i and b_i are the forward and backward hidden state at step i, respectively. Moreover, let $d = [f_{|D|}; b_0]$ be the representation of the whole document.

The document structure is incorporated explicitly with section representations. Let $\mathcal{D}$ bet organized in sections represented as a list of sentences, $[[s_0, .., s_i], [s_{i+1}, .., s_j], [s_{j+1}, ..s_k]...]$, the embedding of each section is defined as the difference of hidden states corresponding to sentences in the section borders. For instance, the embedding of section $[s_{i+1}, .., s_j]$ is defined as $l_1 = [f_{j+1}\check{}f_{i+1}; b_{i+1} - b_j]$.

Decoder. After obtaining sentence as well as global (the entire document) and local context representations (sections), the decoder will combine them using attention, as follows. Given document embedding d, sentence global embedding h^g_i, and section embedding l_t, where s_i belongs to section t, the final sentence representation z_i is obtained as follows,

$$\begin{aligned} e_i^d &= v^T \tanh(W^a[d; h_i^g]), \\ e_i^l &= v^T \tanh(W^a[l_t; h_i^g]), \\ w_i^d &= \frac{e_i^d}{e_i^d + e_i^l}, \\ w_i^l &= \frac{e_i^l}{e_i^d + e_i^l}, \\ c_i &= w_i^d d + w_i^l l_t, \\ z_i &= [h_i^g; c_i], \end{aligned} \tag{4.1}$$

where v^T, W^a are weight parameters. Finally, the probability of selecting s_i is given by $p(y_i = 1|s_i; \theta) = \sigma(\mathrm{ReLU}(W^o z_i))$, where θ represents the model parameters and W^o is a weight parameter.

The E.LG model just described is trained with the Cross-Entropy loss (CE) over the sequence labeling task outlined at the beginning of this section.

4.2.2 Informativeness and Redundancy

We adapt MMR-Select+ (Xiao and Carenini, 2020), the strategy most capable of balancing informativeness and redundancy. Model E.LG is trained using a combined loss that aims to minimize Cross Entropy loss and maximize the expected reward of greedily sampled summary $\hat{S}$ (Qian et al., 2019), defined as:

$$\begin{aligned} \mathcal{L} &= \gamma_R \cdot \mathcal{L}_R + (1 - \gamma_R) \cdot \mathcal{L}_{CE} \\ \mathcal{L}_R &= -(r_I(\hat{S}) - r_I(\bar{S})) \sum_{s_i \in \hat{S}} \log p(y_i \mid s_i) \end{aligned}$$

where $r_I(\bar{S})$ is the informativeness of a baseline summary, used to improve convergence in a self-critic fashion (Paulus et al., 2018). Baseline summary $\bar{S}$ is extracted using greedy selection directly over $p(y_i)$, whereas $\hat{S}$ is extracted greedily using redundancy-aware score p_{MMR}:

$$p_{MMR}(y_i|s_i) = \lambda_R \cdot p(y_i \mid s_i) - (1 - \lambda_R) \cdot \max_{s_j \in \hat{S}} \mathrm{Sim}(s_i, s_j)$$

where $\mathrm{Sim}(s_i, s_j)$ is the cosine similarity between embeddings of sentences s_i and s_j and λ_R controls the redundancy level in $\hat{S}$. This scoring strategy is an extension of MMR (Carbonell and Goldstein, 1998b) that aims to minimize semantic similarity between sentences in $\hat{S}$. In our experiments, we dub this model as E.LG-MMRSel+.

4.2.3 Informativeness and Local Coherence

Building upon Wu and Hu (2018), we define a reward that combines informativeness and local coherence, $r = \lambda_{LC} \cdot r_I + (1 - \lambda_{LC}) \cdot r_{LC}$, where λ_{LC} controls the trade-off between informativeness and coherence and r_{LC} is the score assigned by the CCL classifier outlined in § 2.3.3. Then, E.LG is trained using the REINFORCE algorithm (Williams, 1992) with policy gradient:

$$\nabla \mathcal{L} = -r(\hat{S}) \sum_{s_i \in \hat{S}} \nabla \log p(y_i \mid s_i)$$

where $\hat{S}$ is a candidate summary extracted greedily directly form $p(y_i \mid s_i)$. In our experiments, we label this model as E.LG-CCL.

4.3 Cohesion Control through Memory Simulation

As mentioned in Chapter 2, the KvD theory provides a principled way to operationalize the manipulation of content units during reading and is precise in many aspects of the simulation, e.g. the nature and properties of memory trees.

In this section, two sentence scoring systems are introduced, TreeKvD and GraphKvD, which at their core simulate human working memory during reading, according to the KvD theory. We start by providing an overview of the implemented summarization pipeline. Then, we elaborate on the procedure used to build propositions from syntactic structures automatically extracted from text. Finally, we present the proposed sentence scoring systems in detail, discuss the design choices made, and complement the explanation with a simulation example.

4.3.1 Pipeline Overview

The pipeline for sentence scoring is depicted in Figure 4.2. Input document $\mathcal{D}$ is consumed one sentence at a time by the reading simulator. At each step, one memory cycle is executed and the scores of the propositions in the working memory tree are updated. Once the document has been completely read, the final score of propositions is aggregated into sentence scores, which are then used to select the final summary.

Reading Simulation. The proposed KvD simulators model how content is moved from working memory to long-term memory and vice versa. Working memory is represented as a proposition tree, pruned at the end of each cycle in order to simulate short-term memory limitations in humans. In contrast, long-term memory is represented as an undirected graph

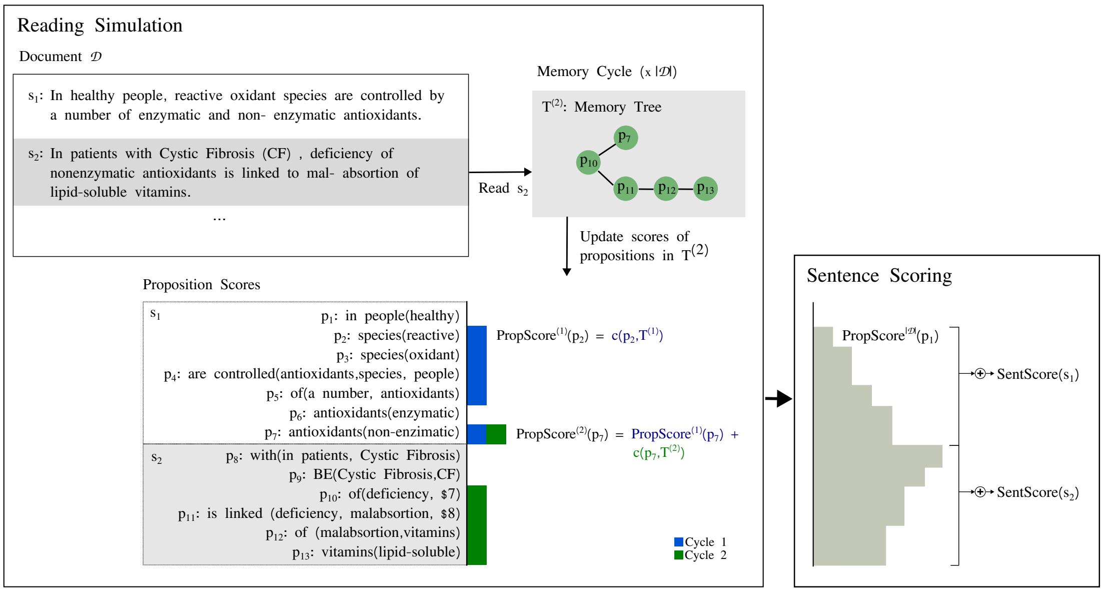

Reading Simulation
Document $\mathcal{D}$
s_1: In healthy people, reactive oxidant species are controlled by a number of enzymatic and non- enzymatic antioxidants.
s_2: In patients with Cystic Fibrosis (CF) , deficiency of nonenzymatic antioxidants is linked to mal- absortion of lipid-soluble vitamins.
...
Read s_2
Memory Cycle (x $|\mathcal{D}|$)
$T^{(2)}$: Memory Tree
p_7
p_{10}
p_{11}
p_{12}
p_{13}
Update scores of propositions in $T^{(2)}$
Proposition Scores
s_1
p_1: in people(healthy)
p_2: species(reactive)
p_3: species(oxidant)
p_4: are controlled(antioxidants,species, people)
p_5: of(a number, antioxidants)
p_6: antioxidants(enzymatic)
p_7: antioxidants(non-enzimatic)
s_2
p_8: with(in patients, Cystic Fibrosis)
p_9: BE(Cystic Fibrosis,CF)
p_{10}: of(deficiency, \$7)
p_{11}: is linked (deficiency, malabsortion, \$8)
p_{12}: of (malabsortion,vitamins)
p_{13}: vitamins(lipid-soluble)
$\text{PropScore}^{(1)}(p_2) = c(p_2,T^{(1)})$
$\text{PropScore}^{(2)}(p_7) = \text{PropScore}^{(1)}(p_7) + c(p_7,T^{(2)})$
🟦 Cycle 1
🟩 Cycle 2
Sentence Scoring
$\text{PropScore}^{|\mathcal{D}|}(p_1)$
$\oplus$ SentScore(s_1)
$\oplus$ SentScore(s_2)

Figure 4.2: Pipeline of KvD reading simulation and sentence scoring for simulation example in Fig.2.1.

of propositions populated by nodes demoted from working memory as reading progresses.

The outline of the the simulation procedure is presented in Algorithm 4.1. The algorithm consumes a document $\mathcal{D} = \langle s_0, \ldots, s_k, \ldots, s_{|\mathcal{D}|} \rangle$ iteratively in memory cycles, updating working memory and long-term memory in each cycle. At the beginning of cycle k, the algorithm reads sentence s_k, extracts its proposition tree P_k (Line 6), and attaches it to the current memory tree T (Line 7). The resulting tree is pruned to a constant size (Line 10) in order to simulate human memory constraints, and pruned nodes are added to the long-term memory graph G. Then, the score of proposition t in cycle k (Line 11) is updated to

$$\text{PropScore}^{k}(t) = \text{PropScore}^{k-1}(t) + c(t,T), \forall t \in T, \tag{4.2}$$

where $c(t,T)$ quantifies the relevance of proposition t by taking into account its position in T. We generalize the idea of reproduction probability in § 2.2.3 by incrementally scoring propositions based on how often they appeared in memory trees and in which part of said trees they were attached. Then, simulation continues to the next cycle until all sentences in $\mathcal{D}$ are consumed. The specific behavior of subroutines `getPropositionTree`, `attachPropositions`, `memorySelect`, and `updateScore` is instantiated by TreeKvD and GraphKvD and their details will be elaborated upon in the following parts of this section.

Sentence Scoring. Once the document has been completely read, the final score of proposition p is $\text{PropScore}(p) = \text{PropScore}^{|\mathcal{D}|}(p)$. We define the score of sentence s_k as the sum of the score of all its composing propositions as

Algorithm 4.1 KvD reading simulation. Subroutines `getPropositionTree`, `attachPropositions`, `memorySelect` and `updateScore` are instantiated by TreeKvD and GraphKvD.

```
Require: $\mathcal{D}$, source document as a list of sentences
Require: WM, size of working memory
Require: $\Psi$, maximum tree persistence
procedure runSimulationKvD($\mathcal{D}$, WM, $\Psi$)
    $T \leftarrow \emptyset$                                        ▷ Memory tree, initially empty
    $G \leftarrow \emptyset$                                        ▷ Long-term memory, initially empty
    $\psi \leftarrow 0$                                             ▷ Tree persistance counter
    for $s_k \in \mathcal{D}$ do
        $P_k \leftarrow$ getPropositionTree($s_k$)
        $T$, attached $\leftarrow$ attachPropositions($P_k, T, G$)
        if attached then
            adjustRoot($T$)
            memorySelect(WM, $T$)
            updateScore($T$)
            $\psi \leftarrow 0$
        else
            $\psi \leftarrow \psi + 1$
        if $\psi = \Psi$ then
            $T \leftarrow \emptyset$
```

$$\mathrm{SentScore}(s_k) = \sum_{p \in V[P_k]} \mathrm{PropScore}(p), \tag{4.3}$$

where $V[P_k]$ is the set of nodes in proposition tree P_k extracted from s_k.

Sentence Selection. We resort to a greedy selection strategy, i.e. selecting the top-scoring sentences according to Eq. 4.3 until the budget of B tokens is met.

4.3.2 Proposition Building

Propositions are obtained by recursively merging and rearranging nodes in dependency trees, extending the procedure outlined in § 3.2.2. Given sentence $s = \langle w_0, w_1, ..., w_N \rangle$ and its corresponding dependency tree Q with nodes $\{q_0, .., q_N\}$,[1] the objective is to obtain proposition tree P with nodes $\{p_0, ..., p_M\}$, $M \leqslant N$, as follows.

First, we merge dependent nodes into head nodes in Q in a bottom-up fashion. Given $u, v \in Q$ where u is head of v, operation merge(u, v) adds all tokens contained in v to node u and transplants children(v) –if any– to children(u). Let dep(u, v) be the grammatical

[1] We follow Universal Dependencies (Nivre et al., 2017), a dependency grammar formalism.

relation between u and v, dependant v is merged into head u if and only if

- Node u is a nominal or non-core dependant of a clausal predicate and v is a function word or a discourse modifier (e.g. interjections or non-adverbial discourse markers).
- Node u is any kind of dependant of a clausal predicate and v is a single-token modifier.
- Nodes u and v form part of a multi-word expression or a wrongly separated token (e.g. $\text{dep}(u,v) = \text{goeswith}$).

Consider the example in Figure 4.3. Starting from dependency tree Q (Fig. 4.3a), single-token modifiers are collapsed into their head nodes (e.g. merge(`model`,`this`) and merge(`galaxy`,`of`)), and compound phrases are joint (e.g. merge(`formation`,`galaxy`)).

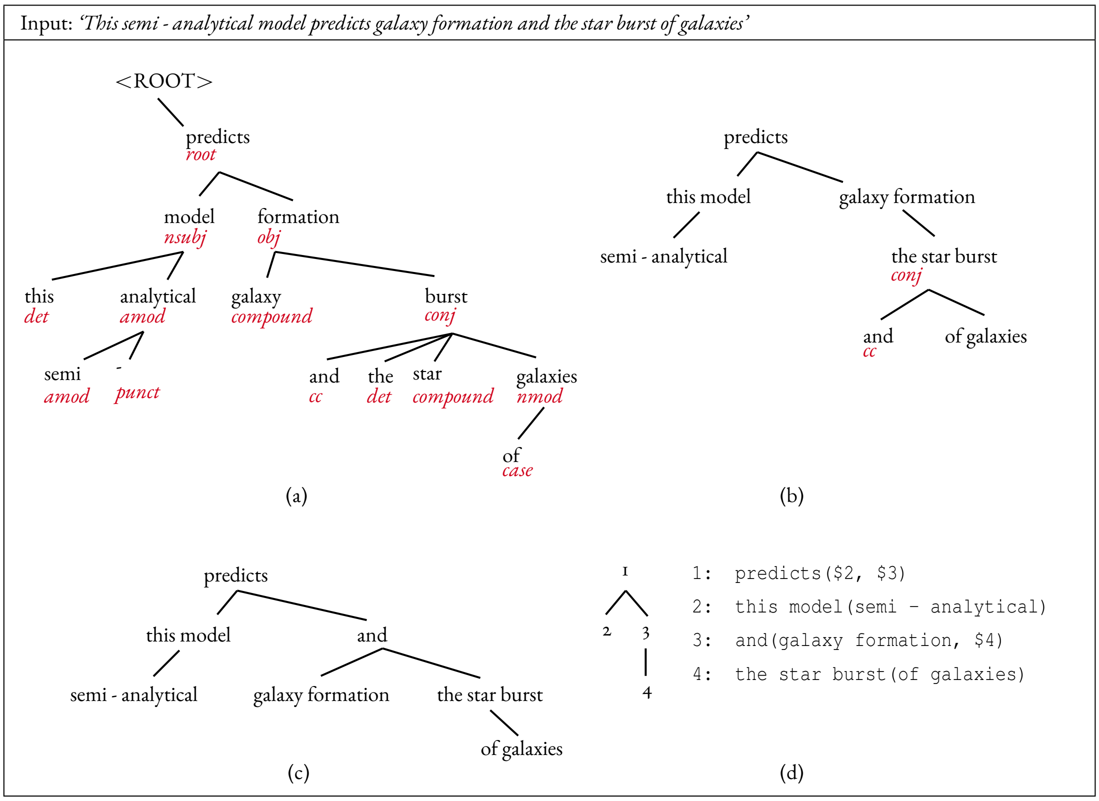
Second, we promote coordinating conjunctions to head status as follows. Given $u, v \in Q$, let v be a node with relation cc among children or grandchildren of u. We transplant node v to u's position and put u and all its children with relation `conj` as children of v. In our example (Fig. 4.3.b), node '`and`' is promoted and nodes '`galaxy formation`' and '`the star burst`' are transplanted as its children. Note that at this point in the procedure Q is still a tree (Fig. 4.3.c) but its nodes might now contain more than one token.

Then, for each non-leaf $u \in Q$ we build proposition $p = w_u(\text{arg}_{v_0}, \text{arg}_{v_1}, ...)$, where w_u is the sequence of tokens contained in node u and $v_i \in \text{children}(u)$. We set $\text{arg}_{v_i} = w_{v_i}$ if v is a leaf node, otherwise arg_{v_i} is a pointer to the proposition obtained from v_i. For instance, proposition `3` in Fig. 4.3.d, `and(galaxy formation,$4)`, presents proposition `4` as one of its arguments since node '`the start burst`', from which proposition `4` is derived, is not a leaf.

Finally, edges between nodes in Q are used to connect their corresponding propositions and form proposition tree P, and we say that two propositions are connected if one proposition has among its arguments a pointer to the other proposition. For instance, proposition `1` in Fig. 4.3.d points to propositions `2` and `3` and hence, they are connected in P.

Under this procedure, connection among propositions in the same sentence takes a syntactic nature. However, propositions from different sentences –and hence different proposition trees– can still be connected if the lexical overlap amongst their arguments is strong enough. Next, we define connection through proposition overlap and how it is quantified.

Proposition Overlap. Differently to the overlap strategy outlined in § 3.2.2 for FangKvD, in this section we present a simplified procedure that does not rely on external linguistic resources. We connect propositions from different sentences by quantifying the lexical overlap between their functors –predicates and arguments. Let $\text{functors}(p)$ be the set of the functors –predicate and arguments– in proposition p. Given $p_1 \in P_x$ and $p_2 \in P_y$, let



```
1:  predicts($2, $3)
2:  this model(semi - analytical)
3:  and(galaxy formation, $4)
4:  the star burst(of galaxies)
```

Figure 4.3: Step-by-step construction of proposition tree from an input sentence, starting from obtaining its dependency tree in UD format (a), merging dependent nodes into head nodes (b), promoting coordinating conjunctions to head status (c), to finally build propositions from non-leaf nodes (d).

$A^*(p_1, p_2)$ be the optimal alignment between functors(p_1) and functors(p_2). Alignment A^* is defined as the maximum matching that can be obtained greedily in the weighted bipartite graph formed from sets functors(p_1) and functors(p_2). The edge weight between two functors is defined as $e(a, b) = \text{jaccard}(L_a, L_b)$, the Jaccard similarity between their sets of lemmas after discarding stopwords, punctuation, and adjectives $-L_a$ and L_b. Then, the average overlap score between p_1 and p_2, $\phi(p_1, p_2)$, is defined as:

$$\phi(p_1, p_2) = \frac{1}{|A^*|} \sum_{\langle a_1, a_2 \rangle \in A^*} \text{jaccard}(a_1, a_2). \tag{4.4}$$

This overlap score function becomes useful when searching an appropriate place to attach incoming propositions to the current memory tree or to pull propositions from long-term memory. We elaborate more on this in the next section.

4.3.3 TreeKvD

In this part, we introduce TreeKvD, the first sentence scoring system simulating KvD reading. The system models working memory and long-term memory as two separate weighted undirected graphs where each node represents a proposition and an edge connecting two propositions indicates the existence of overlap between their arguments, with the edge weight quantifying this overlap. Furthermore, working memory is constrained to be a tree, whereas long-term memory is modeled as a forest of trees pruned from memory trees during simulation. Let s_k be the sentence read in cycle k, T the working memory tree at the beginning of the cycle, with node set $V[T]$ and edge set $E[T]$. Similarly, let G be the long-term memory graph with $V[G]$ and $E[G]$ as node and edge set, respectively. We now elaborate on the details of each step of the TreeKvD's implementation of Algorithm 4.1.

Extracting and Attaching Incoming Nodes. First, subroutine `getPropositionTree` (Line 6) receives s_k as input (as a sequence of tokens) and returns its corresponding proposition tree P_k following the procedure presented in section 4.3.2.

Then, subroutine `attachPropositions` (Line 7) attempts to attach P_k to T, receiving as input structures P_k, T, and G, and returning the updated tree T along with flag `attached` to indicate whether T was modified or not. The attachment of P_k to T and proceeds as follows. We define the optimal place to attach P_k to T as the pair (t^*, p^*) where $t^* \in V[T], p^* \in V[P_k]$ such that

$$(t^*, p^*) = \underset{t \in V[T], p \in V[P_k]}{\operatorname{argmax}} \phi(t, p), \tag{4.5}$$

where $\phi(\cdot)$ is the proposition overlap function defined in Equation 4.4. In case that no attachment pair can be found, i.e. $\phi(t, p) = 0, \forall t \in V[T] \wedge \forall p \in V[P_k]$, `attachPropositions` resorts to two cascaded backup plans.

As first backup attachment plan, the procedure *recalls* a path of forgotten propositions from long-term memory G to serve as bridge to connect P_k and T. Let $\mathcal{F}(R)$ be the set of all paths of length at most R in G, we define the optimal attachment place aided by $\mathbf{f} \in \mathcal{F}$ as the tuple $(t^*, \mathbf{f}^*, p^*)$, such that

$$(t^*, \mathbf{f}^*, p^*) = \underset{t \in V[T], p \in V[P_k], \mathbf{f} \in \mathcal{F}(R)}{\operatorname{argmax}} \phi(t, f_1) + \sum_{i=2}^{n} \phi(f_{i-1}, f_i) + \phi(f_n, p),$$

where $\mathbf{f} = \langle f_1, ..., f_n \rangle, f_i \in V[G] \wedge n \leqslant R$. In this way, P_k is attached to T by retrieving a path $\mathbf{f}^*$ from G with at most R forgotten nodes that maximizes argument overlap between placement candidates t^* and p^*.

In case that no suitable recall path can be found (total overlap score is still zero), procedure `attachPropositions` resorts to a second backup attachment strategy, which consists of deciding whether to keep T as memory tree during the current cycle or whether to replace it completely with P_k. Among both trees, we keep the one whose root node presents the highest closeness centrality. The closeness centrality of a node in an undirected graph is defined as the inverse of the sum of all shortest paths from said node to all other nodes in the graph. As we will discuss in the root adjustment section, a root closer to all other nodes is an indication of a well-balanced tree and allows for efficient pruning, hence a desirable property. In case T is not replaced, the procedure returns flag `attached` as `False`.

Now consider the case when `attachPropositions` fails to attach propositions to T for more than one consecutive cycle. We name this phenomenon *tree persistence*. A highly persistent tree is undesirable since it can potentially block important connections between more recently read propositions. In order to avoid this scenario, we reset the memory tree (line 16 in Algorithm 4.1) if its persistence reaches the maximum permissible value, Ψ. Furthermore, we avoid over-scoring nodes in persistent trees by only updating their score if any form of attachment took place (Line 8).

Choosing and Adjusting the Root. After attachment takes place, subroutine `adjustRoot` will select the most appropriate node in the updated T as the root (Line 9). An important property of working memory trees in the KvD theory is that the root conveys the most central topic at the time of reading. We build upon Fang (2019) criteria and model this property by selecting the node that presents the highest closeness centrality as the root. Such a root would facilitate reaching all nodes in the least amount of steps –in average–, a desired property during pruning.

Pruning Working Memory. Next, subroutine `memorySelect` (Line 10) receives as input memory capacity parameter `WM` and memory tree T, and proceeds to select at most `WM` nodes from T in the following manner. Starting from the root, T is traversed in topological order until reaching a leaf node, selecting each node visited along the way. At this point, if the amount of select nodes is less than `WM`, nodes are selected in breath-first traversing order (starting from the root) until capacity is reached or until all nodes are traversed. Finally, nodes not selected are pruned from T and moved to G.

Proposition Scoring. Following Eq. 4.2, reproduced here for convenience, the score of propositions is updated as

$$\text{PropScore}^{k}(t) = \text{PropScore}^{k-1}(t) + c(t,T), \forall t \in T,$$

in which subroutine `updateScore` (Line 11) defines the updating term $c(\cdot)$ as

$$c(t,T) = \frac{|T_t|}{|T|} \exp\left(\frac{1}{\text{depth}(t)}\right), \tag{4.6}$$

where $\text{depth}(t)$ is the depth of node t with respect to the root and $|T_t|$ is the size of the subtree rooted in t. In this way, nodes closer to the root as well as nodes holding more information in their subtree are scored higher.

Limitations. The presented system closely follows mechanisms of memory organization theorized by (Kintsch and van Dijk, 1978). As such, the system presents a number of processing limitations inherent to the KvD theory itself which we now elaborate on.

First, the constrained amount of content units in working memory at any given time poses a limitation to how much information the system has access to when updating the score of memory tree nodes. It is entirely possible that some propositions are pruned away and never recalled again, in which case their score will be zero.

Second, Kintsch and van Dijk (1978) define the recall mechanism as a routine capable of pulling an unlimited number of propositions from long-term memory. Additionally, propositions might not be recalled *verbatim* but simplified, given that the difficulty to recall specific details increases over time (Postman and Phillips, 1965). In system TreeKvD, we limit ourselves to recall previously read propositions verbatim and further limiting the maximum number of propositions to recall. This design choice limits the possibility of recalling important propositions back into working memory.

Third, attachment of an incoming proposition tree to the current memory tree is done by connecting one node in memory tree to one node in the incoming tree. Whilst this strategy guarantees that the resulting structure remains a tree, as KvD requires, many potentially useful connections are ignored. We address these limitations in the design of the next system.

4.3.4 GraphKvD

The second proposed system, GraphKvD, considers instead a single underlying structure for long-term memory and short-term memory. Working memory is modeled as a subgraph of long-term memory that preserves properties of KvD micro-structure, i.e. a tree with constrained size. Such modeling of memory modules allows for richer connections between incoming proposition trees and working memory, in addition to giving the system efficient access to nodes neighboring memory tree nodes, significantly increasing the coverage of content during scoring. We now proceed to elaborate on how GraphKvD instantiates Algorithm 4.1.

Extracting and Attaching Incoming Nodes. In the same fashion as in TreeKvD, procedure `getPropositionTree` extracts P_k from incoming sentence s_k (line 6). Then, procedure `attachPropositions` will first attempt to connect P_k to T directly, falling back to two cascaded strategies if unsuccessful.

In contrast with TreeKvD, all nodes in P_k are allowed to connect to T. Hence, for each $p \in V[P_k]$, its optimal place to be attached to T is node $t^* = \text{argmax}_{t \in V[T]} \phi(t,p)$, where $\phi(\cdot)$ is again the proposition overlap function defined in Equation 4.4. In case no node in P_k could be connected to any node in T, `attachPropositions` employs again two backup plans. Note that these plans are not triggered if at least one node in P_k was connected to T.

The first plan consists of a recall mechanism that retrieves paths from G connecting each node in P_k to each node in T. For each node $p \in V[P_k]$, its the optimal attachment place $t^* \in V[T]$ aided by path $\mathbf{f}^* = \langle f_1, ..., f_n \rangle$, $f_i \in V[G] \wedge n \leqslant R$, is defined as

$$(t^*, \mathbf{f}^*) = \underset{t \in V[T], \mathbf{f} \subset G}{\text{argmax}} \; \phi(f_1, t) + c(t, T) \left(\sum_{i=2}^{|\mathbf{f}|} \phi(f_{i-1}, \hat{f}_i) \right) \exp(-|\mathbf{f}|) + \phi(f_n, p).$$

Note that GraphKvD defines the optimal attachment place differently from TreeKvD in two respects. First, GraphKvD explicitly favours the attachment of recall paths to highly relevant nodes in T, i.e. high $c(\cdot)$ value. This encourages the memory tree to expand on information about relevant content rather than non-relevant ones. Second, GraphKvD includes an exponential decay length penalty ($\exp(-|\mathbf{f}|)$) to favour the retrieval of shorter recall paths. This penalty is inspired by recent research on how content is gradually forgotten ('decays') in human memory and becomes harder to retrieve (Berman et al., 2009), an idea also applied in the optimization of neural networks (Loshchilov and Hutter, 2019). In this way, we avoid populating T with long proposition chains that may contain only marginally relevant and potentially redundant information. Moreover, this approach aims to save memory capacity for other potentially informative attachments.

As second backup plan, procedure `attachPropositions` will replace T with P_k if $|V[P_k]| > |V[T]|$ and the closeness centrality of the root of P_k is greater than that of the root of T. T will also be replaced if the tree persistence has reached its allowed limit, $\psi = \Psi$. In case P_k is chosen, we *enrich* it by retrieving single nodes from G and connecting them to P, in a similar fashion to the *construction* stage in the Construction-Integration theory of comprehension (Kintsch, 1988). For each node $p \in V[P_k]$, we retrieve candidate nodes in the following order. First, nodes from the local context, i.e. from the current paragraph or article section, are retrieved. Then, nodes are retrieved in inverse order of processing recency, i.e. propositions from sentences processed at the beginning of the simulation are retrieved first. For each node, searching stops when the argument overlap score of a candidate is greater than zero.[2]

[2] Experimentally, increasing this threshold does not impact downstream performance significantly.

This particular retrieval order follows *free recall* accuracy in human subjects (Glanzer, 1972).[3] The tendency to accurately recall the first processed items is known as the *priming effect* (Harley, 1995), and is said to depend on long-term memory. Instead, the tendency to accurately recall the most recent items is called the *recency effect*, and it depends on short-term memory.

Updating Memory Structures. After attachment, long-term memory graph G is updated with nodes and edges in T. Note that after executing the attachment procedures described above, the updated memory graph T might no longer be a tree. However, as mentioned before, the KvD theory models that a valid working memory structure as a tree. Hence, we reduce T to its maximum spanning tree using the argument overlap score between propositions as edge weights. Similarly to TreeKvD, the node with maximum closeness score is chosen as new root. Then, T is pruned down to have at most `WM` nodes using the same strategy as in Section 4.3.3.

Proposition Scoring. The score of nodes in working memory T is updated according to Eq. 4.2 and Eq.4.6. However, GraphKvD will also update the score of nodes neighboring those in T. In this way, propositions that contribute to the understanding of nodes in T are reinforced, and the more a proposition is selected the more its connections are updated. For each node $t \in V[T]$, we define

$$N(t) = \{u; u \in V[G] \backslash V[T], \text{ s.t. } (u,v) \in E[G]\}$$

the set of nodes neighboring t located in G. Then, the updated score of neighbor node u is:

$$\text{PropScore}^{k}(u) = \text{PropScore}^{k-1}(u) + \beta \cdot c(t,T), \forall u \in N(t) \qquad (4.7)$$

where $\beta < 1$ is a decay factor. The consideration of neighboring nodes and a decayed scoring strategy follows the *integration* and *spreading* processing proposed in the Construction-Integration theory. The objective is to integrate peripheral or related concepts into the memory cycle and spread minimal attentional resources to them in the form of score value, where parameter β controls how much attention is leaked.

4.3.5 Simulation Example

Next, we illustrate the procedures outlined in previous sections with an example, showcased in Figure 4.4. The example takes two sentences from a scientific article and simulates two

[3] Free recall is a technique used in psycholingusitic studies of human memory in which a subject is presented with a string of items and is free to recall them in any order; in contrast, *serial recall* requires the subject to recall the items in order.

memory cycles with TreeKvD (left) and GraphKvD (right). The propositions involved (middle row) in the cycles are presented alongside the corresponding gold summary (bottom row). Propositions not directly mentioned in the simulation but necessary for content interpretation are showed in italic. First, we analyse the processes involved during attachment in a memory cycle, including how recall mechanism operates. Then, we relate the properties a memory tree should exhibit according to the KvD theory, and the properties of memory trees obtained with TreeKvD and GraphKvD.

Memory Cycles. In cycle k, both systems manage to attach the incoming proposition tree P directly to the current memory tree T_{k-1}, with such connections illustrated as red dotted lines in Figure 4.4. Notice that TreeKvD is allowed to make only one connection ($79 \mapsto 81$) so that the resulting structure, T', remains a tree. In contrast, GraphKvD is allowed to connect each node in P back to T_{k-1} (e.g. $84 \mapsto 79$, $85 \mapsto 71$), which results in structure G', an undirected weighted graph. After choosing the new root (node 81), the retention process (function `memorySelect`) selects the new memory tree T_k.

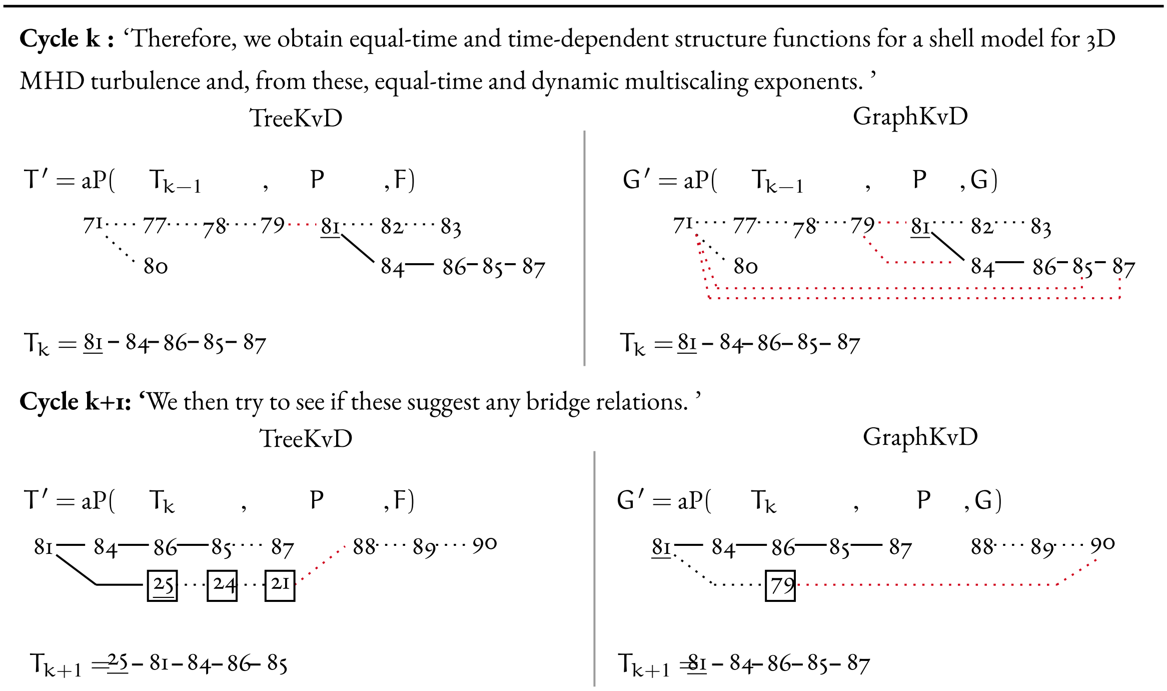
In the next cycle, $k + 1$, the incoming P cannot be attached directly to T_k and hence, the recalled mechanism is used. TreeKvD recalls a 3-node path to connect node 88 to 81, linking information about proposed models (*'models for turbulence'* in 81) to methodology (*'scaling methods'* in $25, 24, 21$) and hypothesis exploration (*'we try to see if these suggest'* in 88). In contrast, GraphKvD recalls a single node linking the studied phenomenon (*'MHD turbulence'* in 81) to its properties of interest (*'such relations'* in 79, making reference to information in 75) and to the specific property being studied (*'bridge relations'* in 90).

Properties of Memory Trees. Properties of memory structures at the micro level, as discussed in Section 2.2.4, have the potential to greatly influence the level of lexical cohesion and redundancy in output summaries, in addition to identifying relevant content to be included. We now elaborate on how this influence manifests in our example.

First, regarding lexical cohesion, a connected memory tree is evidence that content units currently held in memory are not a disjoint set of mutually exclusive concepts but a set that can be interpreted in a coherent manner. For instance, the content in T_{k-1} could be verbalized in the following manner:

> *We examine dynamic multiscaling...in a shell model for 3D MHD [71,72] and scalar turbulence [80]. Dynamic multiscaling exponents are related by linear bridge relations to equal-time multiscaling exponents [75]. We have not been able to find such relations for MHD turbulence so far [77,78,79].*

where the propositions used to verbalize each phrase or sentence are indicated inside square brackets. As can be seen, the text above reads smoothly and exhibits an acceptable level of



Cycle k : 'Therefore, we obtain equal-time and time-dependent structure functions for a shell model for 3D MHD turbulence and, from these, equal-time and dynamic multiscaling exponents. '

Cycle k+1: 'We then try to see if these suggest any bridge relations. '

Propositions

24: must be generalized(that, $21, $25)
21: the simple scaling($22)
22: see(we, at most critical points)
25: to multiscaling(in turbulence)
71: behooves(therefore, it, us, $72, $75, $77)
72: to examine first the dynamic multiscaling(of structure functions, $73)
73: in a shell model for MHD(three dimensional, 3D MHD)
75: are related(dynamic multiscaling exponents, by linear bridge relations to equal time multiscaling exponents)
77: have not been able(we, $78)
78: to find($79, so far)
79: such relations(for MHD turbulence)
80: and($71; scalar turbulence)
81: obtain(therefore, we, $82, $84)
82: and(equal time, $83)
83: time _dependent structure functions(for a shell model)
84: for 3D MHD turbulence from these(and, $86)
85: equal time (dynamic, $87)
86: and($85)
87: multiscaling(exponents)
88: try(then, we, $89)
89: to see($90)
90: suggest(if, these, any bridge relations)

Gold Summary

We present the first study of the multiscaling of time-dependent velocity and magnetic-field structure functions in homogeneous, isotropic Magnetohydrodynamic (MHD) turbulence in three dimensions . We generalize the formalism that has been developed for analogous studies of time-dependent structure functions in fluid turbulence to MHD. By carrying out detailed numerical studies of such time-dependent structure functions in a shell model for three-dimensional MHD turbulence, we obtain both equal-time and dynamic scaling exponents .

Figure 4.4: Simulation example showcasing input sentence, propositions, and memory tree, per cycle. Common content between propositions and the gold summary is shown in blue. Solid line: edge in final memory tree; dotted line: pruned edge; red dotted line: edge connecting T and P. Squared nodes: propositions recalled from long-term memory; underlined node: new root of tree. aP: `attachPropositions` (Alg. 4.1).

lexical cohesion and co-referential coherence. By updating the score of a set of propositions capable of forming a coherent text, a KvD system encourages the similar ranking of mutually coherent propositions. Hence, a content selector is also encouraged to select a set of sentences exhibiting a non-trivial level of lexical cohesion.

A similar reasoning can be applied to explain the influence of memory simulation over redundancy in output summaries. As claimed in Section 2.2.4, a memory tree constitutes a non-redundant set of propositions, with each proposition adding details of an entity or topic shared with the propositions it is connected to. For instance, node 81 adds information about 'MHD turbulence' to T_{k-1} when connected to node 79. Moreover, when the recall mechanism is used in cycle $k+1$, only one recall path is added to T_k $(25, 24, 21$ in TreeKvD and 79 in GraphKvD) instead of many potentially redundant recall paths. Hence, by updating the score of a minimally redundant set of propositions in each cycle, a KvD system encourages non-redundant content to be ranked closely and by extension, the content selector is encouraged to select sentences with an acceptable level of redundancy.

Finally, memory trees are capable of identifying and ranking relevant propositions, hence encouraging a selector to pick sentences with relevant content. In our example, we observe that both TreeKvD and GraphKvD retain propositions 81, 84, 85, 86 in T_k and T_{k+1}. These propositions cover information directly mentioned in the gold summary, coloured in blue in Figure 4.4.

4.4 Experimental Setup

In this section we present the experimental setup for assessing the trade-off between informativeness, redundancy, and cohesion, under the two control scenarios defined in previous sections, reward-guided and unsupervised. We evaluate our models on the task of extractive summarization of scientific articles and define appropriate automatic evaluation metrics to capture the analyzed summary properties. Moreover, we design two human evaluation campaigns aimed to quantify the perceived informativeness and cohesion of summaries produced by the proposed unsupervised systems, TreeKvD and GraphKvD. In the following, we elaborate on the datasets used and the preprocessing employed, the comparison systems, and the setup for automatic and human evaluation.

4.4.1 Datasets

We used PubMed and arXiv datasets (Cohan et al., 2018), consisting of scientific articles in English in the Biomedical and Computer Science, Physics domains, respectively. For each article, the source document is defined as the concatenation of all section texts, and the abstract is used as reference summary. We further preprocessed both datasets after noticing substantial sentence tokenization errors and pollution of latex code. Similar to the previous chapter, articles with abstracts with less than 50 tokens and more than 300 tokens were discarded, as well as articles with documents with less than 100 tokens. Sentences are capped to 200 tokens, and sentences with more than 3 latex code keywords (e.g. *usepackage, documentclass*) and less than 5 tokens are ignored. Following previous work (Xiao and Carenini, 2020; Gu et al., 2022), we use a budget of $\mathrm{B}=200$ tokens for both arXiv and PubMed.

4.4.2 Comparison Systems

In addition to the discussed and proposed models, we report results on a range of standard heuristic and unsupervised baseline systems. As heuristic baselines we include the following: extractive oracle, Ext-Oracle, which consists on greedily selecting a set of sentences that maximize the sum of ROUGE-1 and ROUGE-2 F_1 scores w.r.t. the reference summary; Lead, selecting the leading sentences of a document until the budget is met; and Random, randomly sampling sentences following a uniform distribution. Next, we elaborate on the training details and hyper-parameter configuration of our reward-based and unsupervised systems.

Supervised and Reinforcement Learning Systems. We report performance of E.LG as a reference for an informativeness-oriented baseline, and use the checkpoints provided by Xiao and Carenini (2020). For redundancy-oriented model E.LG-MMRSel+, we use the default hyper-parameter configuration (Xiao and Carenini, 2020) and set $\lambda_R = 0.6$, $\gamma_R = 0.99$. For local coherence-oriented model E.LG-CCL, we tune λ_{LC} over validation sets and set it to $\lambda_{LC} = 0.2$. Both models were trained using Adam optimizer (Loshchilov and Hutter, 2019), batch size of 32, learning rate of 10^{-7}, and trained for 20 epochs, with the best checkpoint selected based on the sum of ROUGE-1 and ROUGE-2 F_1 scores.

In addition, we compare against MemSum (Gu et al., 2022), a model that employs a multi-step episodic Markov decision process that samples a candidate summary sentence by sentence instead of sampling the complete summary via a single action (Narayan et al., 2018b; Dong et al., 2018). Crucially, MemSum incorporates an *extraction history* module that informs the agent about the information already selected and hence, minimize redundancy in the final summary. Although the model is trained to produce a *stop* action, we stop extraction once the budget is met in order to have a fairer comparison with other baselines in terms of summary length.

Finally, similarly to the previous chapter, we do not include supervised baselines that require the calculation of coreference chains or rhetorical structure trees over the input document, such as DiscoBERT (Xu et al., 2020), because of their limited applicability in out-of-domain scenarios and their inability to process documents of the length analyzed in this chapter.

Unsupervised Systems. For the proposed KvD systems, we perform hyper-parameter tuning over the validation sets, and set the maximum recall path length $R = 5$, maximum tree persistence $\Psi = 8$, working memory capacity $\texttt{WM} = 100$ for both TreeKvD and GraphKvD. For proposition scoring in GraphKvD, decay factor is set to $\beta = 0.01$. During proposition building, we use UDPipe 2.0 (Straka, 2018) to extract dependency trees.

Similarly to the experimental setup in Chapter 3, we differentiate between *completely* unsupervised systems and unsupervised systems that require some form of finetuning using data in the target knowledge domain, the latter being marked with (*). Regarding completely unsupervised systems, we compare against TextRank (Mihalcea and Tarau, 2004) and PacSum (Zheng and Lapata, 2019) with a SciBERT sentence embedder. These systems model a document as a graph of sentences and employ node centrality (eigen-vector and weighted degree centrality, respectively) as a proxy for informativeness. As a non-completely unsupervised system, we report results for PacSum-FT*, finetuned over a sample of 1000 documents following the procedure therein. For more details about these systems, please refer to § 3.3.

Moreover, we investigate the appropriateness of constraining the size of working memory during KvD simulation, and define baseline FullGraph, which simulates all steps of KvD reading in Alg. 4.1 except subroutine `memorySelect`. Similarly to PacSum, proposition connection is limited to those in the previous 200 sentences. Finally, we compared our proposed models against a previous implementation of the KvD theory (Fang, 2019), labeled as FangKvD. This system is equivalent to a reader configuration `Cnt-Cnt` in Chapter 3.

4.4.3 Automatic Evaluation

We evaluate the intrinsic performance of the analyzed models in terms of informativeness, redundancy, local coherence, and lexical cohesion. For more details on the following metrics, please refer to § 2.3.

Informativeness. We report F_1 ROUGE (Lin, 2004) for n-gram overlap-based relevance, and F_1 BertScore (Zhang et al., 2019) for semantic relevance. In all our experiments, we report scores using RoBERTa (Liu et al., 2019) as underlying model, and apply importance weighting to diminish the effect of non-content words, e.g. function words.[4]

Redundancy. We assess redundancy in summaries with Inverse Uniqueness (IUniq) and Sentence-wise ROUGE (RdRL). For IUniq, we report the mean among values for unigrams, bigrams, and trigrams. For RdRL, we use ROUGE-L F_1 score as base.

Cohesion. Lexical cohesion in summaries is measured using the Entity Grid (EEG) model (Barzilay and Lapata, 2008) and the Entity Graph (EGr) model (Guinaudeau and Strube, 2013). For EEG, we employ the implementation part of the Brown Coherence Toolkit[5], using the extended features setup (Elsner and Charniak, 2011) and train models over 50 000 samples uniformly chosen from each training set. For EGr, we use the spaCy wrapper over UDPipe [6] to perform POS tagging and ultimately to extract nouns.

Local Coherence. The local coherence of a summary is assessed using the CCL scorer (CCL; Steen and Markert 2022) using a window of 3 sentences taken with padding of one sentence. We train separate CCL models for each dataset analyzed.

Metric Reliability. The automatic metrics used in this chapter present the following limitations that might impact their reliability. For informativeness, as mentioned in § 3.3.4, ROUGE is impacted by the difference in length between reference summaries and candidate summaries. In this chapter, similarly to Chapter 3, this issue is mitigated by discarding dataset instances where the gold summary is too short or too long and by setting a hard budget for the summary length.

Regarding metrics of lexical cohesion, their reliability depends on the accuracy of noun extraction. EEG employs a co-reference resolution tool (Ng and Cardie, 2002) that uses lexical, grammatical, and semantic features, in order to extract and link nouns from sentences. This method –rather limited to modern NLP standards– is complemented by metric EGr, which instead employs strong neural taggers for noun extraction.

In the case of local coherence, reliability might be impacted by the length (in wordpieces) being scored at a time by the model (Steen and Markert, 2022). In this chapter, we train our CCL scorers using binary cross-entropy with positive and negative examples taken from different documents, hence mitigating the model bias for chunk length.

4.4.4 Human Evaluation

We elicit human judgments to assess informativeness and cohesion in two separate studies conducted on the Amazon Mechanical Turk platform. We sampled 30 documents from the

[4] IDF statistics were obtained from documents in the training set of each dataset.

[5] **https://web.archive.org/web/20200505174052/https://bitbucket.org/melsner/browncoherence**

[6] https://spacy.io/universe/project/spacy-udpipe

test set of PubMed and the respective summaries extracted by unsupervised systems optimizing for cohesion, i.e. TreeKvD, GraphKvD, and PacSum.

In order to ensure the quality of annotations, we put in place catch controls (reading confirmation and annotation time), i.e. annotations that did not pass the control were discarded. For more details on catch trials, instructions, and examples, please refer to Appendix B.2. We now elaborate on the details of each study.

Informativeness. In the first study, subjects were shown the abstract and the introduction of a scientific article along with two system summaries. Subjects were then asked to select the most informative summary among them with the possibility to select both in case of a tie, following previous work (Wu and Hu, 2018; Luo et al., 2019; Fabbri et al., 2021). In each system pair comparison, a system is assigned rank 1 if its summary was selected as most informative, and rank 2 otherwise. In case of a tie, both systems are assigned rank 1. Then, the score of a system is defined as its average ranking. We collected three annotations per system-pair comparison and made sure that the same annotator was not exposed to the same document twice. As an additional catch trial, we included in each annotation batch an extra instance with summaries extracted by the extractive oracle and the random baseline.

Cohesion. Lexical chains are sequences of semantically related words (Morris and Hirst, 1991), and the distribution of these chains across a text has been shown to be a strong indicator of cohesion (Barzilay and Elhadad, 1997; Galley and McKeown, 2003). We relax the concept of lexical chains and extend it to that of *chains of summary content units (SCUs)*, where all SCUs in a chain cover semantically related content.

In our second study, we aimed to capture cohesive ties between sentences in a system summary by asking participants to identify SCU chains. Following previous work on semi-automation of the pyramid method (Zhang and Bansal, 2021), we employ propositions –as extracted in Section 4.3.2– as surrogates for SCUs. Hence, a propositional chain is defined as a set of propositions that exhibit semantically related arguments.

Participants were shown a single system summary as a list of sentences where tokens that belonged to the same proposition were colored the same, as depicted in the example in Figure 4.5. Then, the task consists of selecting chains of colored text chunks that shared content among them. For instance, in our example proposition chain $\{0, 6, 7\}$ is connected through information about *the proposed method*, whereas chain $\{1, 3, 6\}$, through *optic nerve segmentation*. Chains were allowed to be non-exclusive, i.e. propositions can be selected in more than one group. Similarly to the previous study, we collected three annotations per system summary and include the gold summary of an extra system in the campaign.

Finally, based on annotations of propositional chains, we define the following measurements of lexical cohesion: (i) *chain spread*, defined as the average number of sentences between two consecutive propositions in a chain; (ii) *chain density*, the number of chains covering the same sentence[7]; and (iii) *sentence coverage*, the number of sentences covered by at least one chain. Intuitively, a text with less spread propositional chains exhibits cohesive ties that link sentences that are closer to each other, making the topic transition between sentences smoother (Hassan et al., 1976). Chain density can be interpreted as an indicator of the topic density in a sentence as well as how well a sentence connects to preceding and posterior sentences, e.g. by connecting to a preceding sentence through one chain and connecting to a posterior one though another chain. Finally, sentence coverage constitutes a straightforward measurement of how many sentences are connected through cohesive ties in a summary.

Agreement between human annotators is obtained by calculating the average text overlap between proposition chains, as follows. Given candidate summary $\hat{S}$, let C_A and C_B be sets of chains extracted from $\hat{S}$ by annotators A and B, respectively. Given chains $a \in C_A$ and $b \in C_B$, we define Precision, Recall, and F_1 score as follows,

$$\begin{aligned} P^{ov}(a,b) &= \frac{\sum_{p\in a} \max_{q\in b} |LCS(p,q)|}{\sum_{p\in a} |p|} \\ R^{ov}(a,b) &= \frac{\sum_{q\in b} \max_{p\in a} |LCS(p,q)|}{\sum_{q\in b} |q|} \\ F_1^{ov}(a,b) &= \frac{2\cdot P^{ov}\cdot R^{ov}}{P^{ov}+R^{ov}} \end{aligned}$$

where p and q are propositions included in chains a and b, respectively, $LCS(p,q)$ is the longest token sequence common to p and q, and $|p|$ indicates the number of tokens covered by p. Then, the overlap score between annotator A and B is defined as

$$\text{ChainOverlap}(A,B) = \frac{1}{|C_A|\cdot|C_B|} \sum_{a\in C_A, b\in C_B} F_1^{ov}(a,b). \tag{4.8}$$

Finally, we report the average overlap score over all pair of annotators, averaged over all system summaries.

4.5 Results and Discussion

In this section, we present results for our proposed systems, TreeKvD and GraphKvD, and comparison systems on the PubMed and arXiv datasets. First, we discuss the trade-offs systems incur when aiming to balance informativeness, redundancy, and lexical cohesion, under varying setups of training supervision. Then, we investigate how systems apply

[7] We say that a chain *covers* a sentence if at least one of the chain's proposition belongs to said sentence.

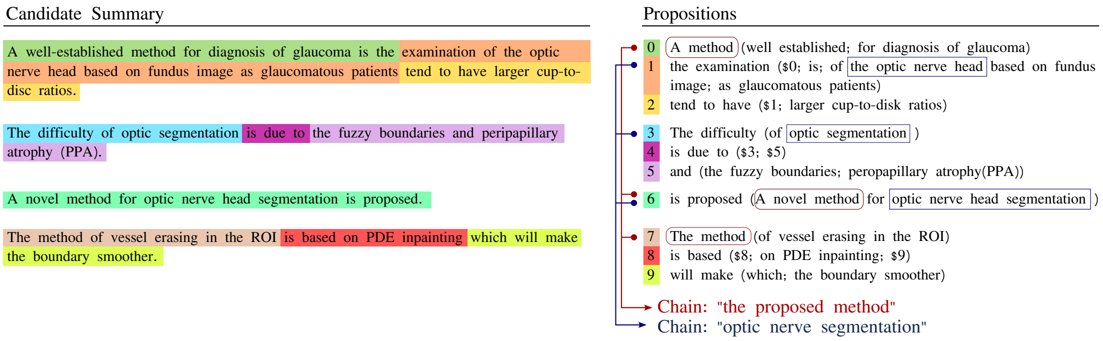

Candidate Summary
A well-established method for diagnosis of glaucoma is the examination of the optic nerve head based on fundus image as glaucomatous patients tend to have larger cup-to-disc ratios.
The difficulty of optic segmentation is due to the fuzzy boundaries and peripapillary atrophy (PPA).
A novel method for optic nerve head segmentation is proposed.
The method of vessel erasing in the ROI is based on PDE inpainting which will make the boundary smoother.
Propositions
0 A method (well established; for diagnosis of glaucoma)
1 the examination ($0; is; of the optic nerve head based on fundus image; as glaucomatous patients)
2 tend to have ($1; larger cup-to-disk ratios)
3 The difficulty (of optic segmentation)
4 is due to ($3; $5)
5 and (the fuzzy boundaries; peropapillary atrophy(PPA))
6 is proposed (A novel method for optic nerve head segmentation)
7 The method (of vessel erasing in the ROI)
8 is based ($8; on PDE inpainting; $9)
9 will make (which; the boundary smoother)
Chain: "the proposed method"
Chain: "optic nerve segmentation"

Figure 4.5: Example of proposition chain annotation in our cohesion evaluation campaign. Each coloured chunk in the candidate summary corresponds to a pre-extracted proposition. Users are tasked to group text chunks that share information by clicking on them. Best seen in colour.

these trade-offs across increasing levels of source document redundancy. Finally, we present a thorough analysis, both quantitative and qualitative, of how properties of simulated cognitive processes affect final summaries.

4.5.1 Informativeness, Redundancy, and Cohesion

We start by analyzing the performance of our models in terms of relevancy, redundancy, and cohesion. Results on informativeness are summarized in Table 4.1, whereas results on redundancy, cohesion, and local coherence metrics are presented in Table 4.2. Both tables are organized in three sections: heuristic systems (*Heur.*), supervised and reinforcement learning-based systems (*Sup., R.L.*), and unsupervised systems (*Unsup.*). Systems are color-coded according to which summary properties they aim to optimize, such as informativeness (I), redundancy (R), and cohesion (C). For completeness, we also report redundancy and cohesion of reference summaries (Gold, last row in Table 4.2) to have a reference point for a desirable level of redundancy and cohesion.

Statistical significance at the system level is tested pairwise using Bootstrap resampling (Davison and Hinkley, 1997) with a 95% confidence interval. For PubMed, we found no pairwise statistical difference between R1 scores of systems TreeKvD and GraphKvD; and between systems E.LG, E.LG-MMRSel+, and E.LG-CCL. For arXiv, no pairwise statistical difference in R1 scores was found between systems TreeKvD and GraphKvD; and between systems E.LG, E.LG-MMRSel+, MemSum, and E.LG-CCL. Analogously, Table 4.1 and 4.2 indicate system groups in which no pairwise difference was found, one group per marker, for each metric reported.

Aim	System	PubMed				arXiv			
		R1	R2	RL	BSc	R1	R2	RL	BSc
-	Ext-Oracle	59.62	35.14	54.52	88.22	58.66	30.28	52.28	87.12
-	Lead	37.07	12.73	33.28	82.94	36.46	9.78	32.02	82.58
-	Random	36.11	10.43	32.70	82.21	33.02	6.52	30.09	80.51
I	E.LG	47.34‡	21.04‡	42.42‡	85.17‡	46.38‡	18.66‡	40.77	85.01‡
I,R	E.LG-MMRSel+	47.55‡	21.20‡	42.70‡	85.21‡	46.52‡	18.69‡	**41.06**‡	85.00‡
I,R	MemSum	**48.02**	**22.06**	**43.16**	**85.63**	**46.69**‡	**19.50**	41.02‡	**85.13**
I,C	E.LG-CCL	47.42‡	21.21‡	42.57‡	85.34	46.35‡	18.74‡	40.80	85.05‡
I,C	*PacSum-FT* *	40.05	13.66	36.29	83.86	38.05	9.87	34.18	83.06
I	FullGraph	35.48	11.06	30.28	81.89	27.44	6.61	22.75	78.73
I	TextRank	**41.51**	**15.37**	**35.78**	**83.59**	**40.32**	**12.67**	**34.06**	**82.68**
I,C	PacSum	37.01	10.07	33.55	82.98	33.41	6.54	30.48	81.70
I,R,C	FangKvD	35.80	10.94	30.97	82.17	32.76	8.31	27.81	80.60
I,R,C	TreeKvD (ours)	37.22†	11.40†	32.37†	82.61†	34.90†	9.06†	29.85†	81.16†
I,R,C	GraphKvD (ours)	37.21†	11.42†	32.25†	82.57†	34.98†	9.19†	29.73†	81.14†

Table 4.1: Performance of systems over PubMed and arXiv test sets in terms of ROUGE F_1 (R1, R2, RL) and BERTScore (BSc). Optimization Aim (Aim) indicates whether a system was optimized for (I)nformativeness, (R)edundancy, Cohesion (C), or a combination of these, grouped by color. Best models in each section are **bolded**. (†,‡): no statistical difference between systems in the same section and column. (*): non-completely supervised system.

Heuristics. It is worth noting that the extractive oracle, Ext-Oracle, even though optimized for informativeness by design, can still be used as a good-enough reference for redundancy in an extractive summary, given that RdRL and IUniq scores remain tightly close to those of Gold. However, note that summaries extracted by Ext-Oracle need not be lexically cohesive, as indicated by its lower CCL scores than systems optimized for cohesion. Instead, Lead does obtain high EEG, EGr, and CCL scores, and low RdRL and IUniq scores, a trend also present in Gold. These measures indicate that such a trend is proper of cohesive text. Notice, however, that source documents in arXiv might showcase lower lexical cohesion than those in PubMed, as indicated by their EEG and EGr scores. Finally, it can be observed that the organization of information in scientific articles poses a challenge for trivial baselines, as evidenced by the low ROUGE scores of Lead and Random.

Supervised and Reinforcement Learning Systems. When optimizing one extra summary property besides informativeness in a reinforcement learning setup, the following in-

Aim	System	PubMed					arXiv				
		RdRL	IUniq	EEG	EGr	CCL	RdRL	IUniq	EEG	EGr	CCL
-	Ext-Oracle	14.07	18.72	0.76	0.84	0.58	14.98	18.78	0.71	0.72	0.40
-	Lead	12.75	18.25	0.72	0.78	0.76	13.95	19.32	0.68	0.96	0.77
-	Random	11.36	18.29	0.63	0.69	0.41‡	10.78	20.67	0.61	0.61	0.24
I	E.LG	16.19	21.60‡	0.75‡	1.03	0.18	16.71†	21.20‡	0.70	1.01	0.21‡
I,R	E.LG-MMRSel+	**15.03**	**20.69**	0.75‡	0.96	0.16	**14.58**	**20.66**	**0.71**†	0.91	0.21‡
I,R	MemSum	17.24	24.01	0.75	0.75	0.48	16.80†	21.89‡	0.69	1.03	0.44†
I,C	E.LG-CCL	16.92	21.21‡	0.75‡	**1.04**†	**0.51**	16.92†	21.21‡	0.70‡	**1.05**	**0.45**†
I,C	*PacSum-FT**	12.92	18.76	0.73	0.77	0.61	11.42‡	16.93	0.72	0.67‡	0.56
I	FullGraph	15.82	23.79	0.73	0.68	0.45	11.65‡	33.22	0.56	0.67‡	0.24
I	TextRank	22.08	26.76	**0.78**	**1.05**†	0.41‡	17.55	22.25	**0.72**†	**1.02**	0.26
I,C	PacSum	11.66	20.84†	0.64	0.71‡	**0.49**†	10.17	**19.27**	0.62	0.44	**0.40**
I,R,C	FangKvD	12.59	**20.45**†	0.74	0.70‡	**0.50**†	12.15	26.11	0.66	0.69	0.34
I,R,C	TreeKvD (ours)	13.06	**20.62**†	0.75†	0.83	**0.49**†	12.72	24.22†	0.70‡	0.83†	0.36
I,R,C	GraphKvD (ours)	**13.74**	21.00‡	0.75†	0.85	0.44	**13.46**	24.57†	**0.71**†	0.84†	0.31
	Gold	13.54	19.12	0.70	0.96	0.91	14.83	17.27	0.72	0.87	0.89

Table 4.2: Redundancy (RdRL, IUniq), cohesion (EEG, EGr), and local coherence (CCL) levels in candidate summaries over PUBMED and ARXIV test sets. See Table 1 for details on Optimization Aim (Aim) and color coding. Best models in each section are **bolded**, according to redundancy (those closest to GOLD), cohesion and coherence (the higher the better). (†,‡): no statistical difference between systems in the same section and column. (*): non-completely supervised system.

sights can be drawn. First, it is possible to reduce redundancy or improve lexical cohesion without losing informativeness: E.LG-MMRSEL+ and E.LG-CCL obtain comparable ROUGE scores to E.LG, a supervised system optimized only for informativeness. E.LG-MMRSEL+ obtains the lowest redundancy scores (RdRL and IUniq) and E.LG-CCL, the highest cohesion and local coherence scores in terms of EGr and CCL, respectively. However, optimizing for redundancy or informativeness alone incurs a huge sacrifice of in terms of cohesion, as indicated by the low CCL scores. On the other hand, optimizing for cohesion entails maintaining a non-trivial level of redundancy, as indicated by the RdRL and IUniq scores in E.LG-CCL, which are higher than those of E.LG and E.LG-MMRSEL+.

Second, we find that tackling redundancy in the model architecture itself, i.e. MEMSUM, works consistently better than using a redundancy-aware reward during training, i.e. E.LG-MMRSEL+. Not only does MEMSUM obtain higher ROUGE scores, but seems to better balance cohesion and redundancy. Even though MEMSUM's CCL scores are lower than E.LG-CCL in both datasets, they are significantly higher than those of E.LG-MMRSEL+. Once again, we observe the trade-off between cohesion and redundancy, as indicated by the higher redundancy scores in MEMSUM.

Unsupervised Systems. When comparing proxies for relevancy, we find that sentence centrality (as in TEXTRANK and PACSUM-FT) performs better than sentence scoring based on reading comprehension, such as in our proposed KvD systems. However, whilst TEXTRANK obtains the highest ROUGE-1 and 2 scores in both datasets, it also obtains the highest redundancy scores (in terms of RdRL) and low CCL scores (lowest in PUBMED and second to lowest in ARXIV). A similar trend can be observed for FULLGRAPH. Since both FULLGRAPH and TEXTRANK use PageRank to rank content, we can conclude that lexical overlap at the sentence level is more beneficial than overlap at the proposition argument level, as done by FULLGRAPH. Interestingly, EEG and EGr scores for TEXTRANK are surprisingly high in both datasets. Upon closer inspection, we found that EEG detects very few entity chains –most of the time a single one– with high probability. For EGr, this translates into having a sentence graph where edges are a result of co-occurrence of the same very few nouns. This phenomenon can be interpreted as a sign of poor content coverage and high redundancy.

Consider now systems PACSUM and PACSUM-FT. First, we notice that perhaps unsurprisingly, finetuning over in-domain data gives huge improvements in relevancy and a better cohesive-redundancy trade-off. Second, unlike the supervised scenario, we observe that adding a proxy for cohesion during training significantly hurts relevancy. This can be observed by the higher ROUGE-1 and 2 scores of TEXTRANK against PACSUM-FT. Notice, however, that fluency (ROUGE-L) and semantic relevancy (BertScore) do experiment an improvement. Moreover, PACSUM-FT obtains more cohesive summaries than EXT-ORACLE and even the supervised baseline optimized for local coherence, E.LG-CCL. We hypothesize that PACSUM and PACSUM-FT model a strong proxy for cohesion by encouraging strong connections between neighboring sentences.

When comparing KvD systems in terms of relevancy scores (ROUGE-1 and 2), we observe that GRAPHKVD and TREEKVD significantly outperform other unsupervised baselines, except TEXTRANK. Notice, once again, that PACSUM obtains better fluency (ROUGE-L) and semantic relevancy (BertScore). Whilst PACSUM aims to optimize local coherence, it does not explicitly encourage lexical cohesion, as indicated by its EEG and EGr scores, lower than KvD systems. In contrast, KvD systems improve lexical cohesion, which translates in higher EEG and EGr scores and in turn, slightly higher redundancy scores. The contrast is more defined when the source documents present low lexical cohesion, as is the case for arXiv.

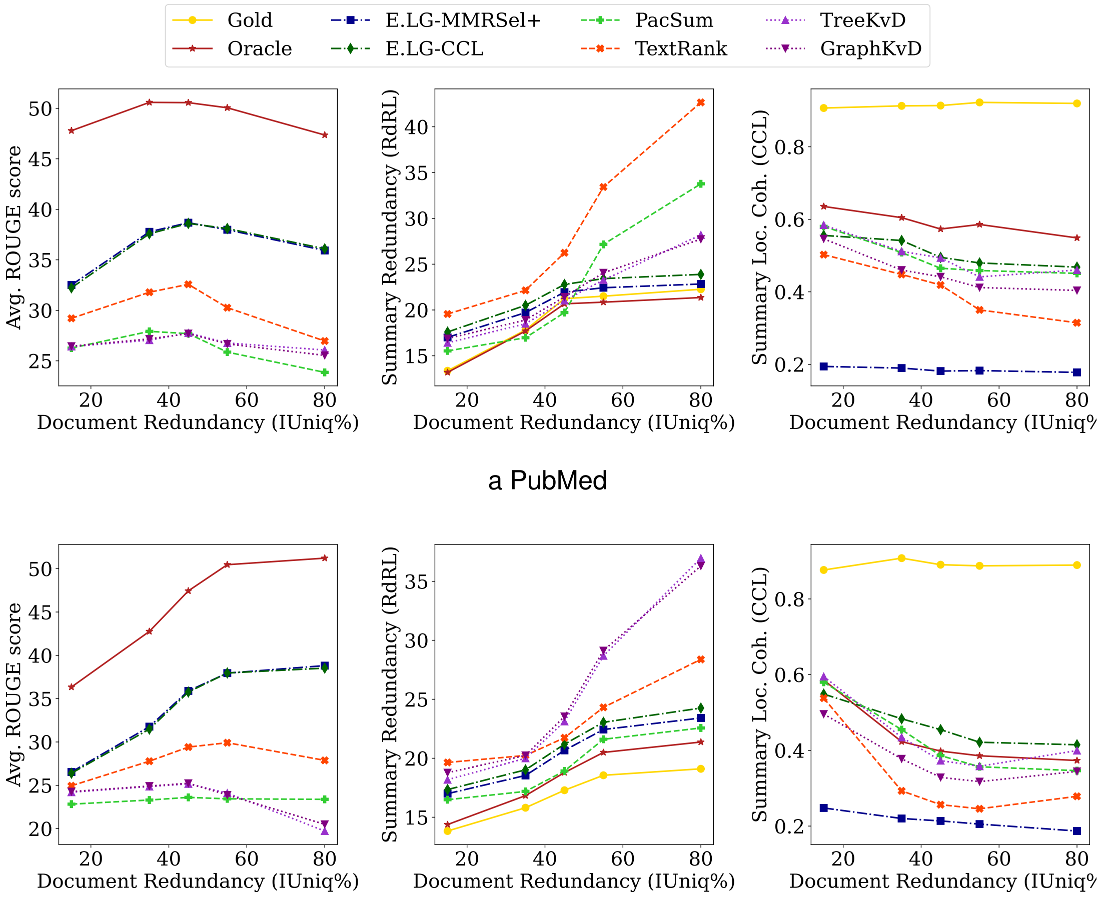
It is worth noting the advantage of the proposed KvD systems against a previous implementation of the KvD theory, FangKvD. We hypothesize of two reasons behind this result. First, FangKvD relies on external domain-dependant resources like WordNet, which makes it hard to apply in highly domain-specific applications such as the scientific domain. Second, GraphKvD and TreeKvD score propositions based on their position on the memory tree during simulation, whereas FangKvD only counts how many times a proposition has appeared in a memory cycle. Note also that our proposed KvD systems outperform FullGraph, highlighting the importance of constraining working memory in each cycle. In terms of cohesion-redundancy trade-off, we observe that TreeKvD obtains a comparable balance to FangKvD in PubMed but a better balance for arXiv. Notice that in both datasets, GraphKvD obtains redundancy scores closest to Gold w.r.t. RdRL but lower CCL scores than TreeKvD. In contrast, EEG and EGr scores indicate that GraphKvD maintains a comparable level of lexical cohesion to TreeKvD.

4.5.2 Effect of Document Redundancy

Next, we take a closer look at the redundancy and cohesion levels in summaries extracted from increasingly redundant documents. Figure 4.6 shows performance of summarization systems in terms of informativeness (average ROUGE score, (ROUGE-1 + ROUGE-2 + ROUGE-L)/3), redundancy (RdRL), and local coherence (CCL) across different levels of document redundancy (IUniq). Test sets were divided in bins according to their document redundancy score and the average metric value per bin is reported. For simplicity, we only plot performance of representative systems in each section.

Reinforcement Learning Systems. In general, we observe that performance in informativeness and redundancy degrades slightly but surely as redundancy increases in the source document. Most notably, E.LG-MMRSel+ and E.LG-CCL show comparable robustness in informativeness and redundancy, whilst E.LG-CCL shows significantly better robustness in local coherence, highlighting the importance of optimizing for cohesion instead of redundancy.

Unsupervised Systems. In PubMed, we observe that PacSum and TextRank are highly susceptible to document redundancy, showing quick degradation in informativeness and redundancy as document redundancy increases. Whilst PacSum remains robust in terms of cohesion, TextRank exhibits a significant drop. In contrast, TreeKvD and



a PubMed

b arXiv

Figure 4.6: Informativeness (left), summary redundancy (mid), and summary local coherence (right) across increasing levels of document redundancy. Metric values are averaged over each document redundancy range.

GraphKvD show more robustness w.r.t. informativeness, remain closer in redundancy to Gold, and show local coherence levels comparable to E.LG-CCL. Notably, our KvD systems show comparable redundancy to the RL-based baselines at low and mid levels of document redundancy. This indicates that our systems manage to successfully balance informativeness, redundancy and cohesion across increasing levels of document redundancy.

In arXiv, however, a few differences can be observed. First, PacSum shows notable robustness to document redundancy, and remains closer in redundancy to Gold than all other unsupervised systems. Our KvD systems exhibit a degradation in informativeness and redundancy, although robustly keeping high levels of cohesion. We hypothesize that KvD systems prioritize cohesion above informativeness and redundancy. In addition, we point out that arXiv is composed of noisier text than PubMed, exhibiting a number of preprocessing errors that might affect the quality of the proposition extraction.[8]

4.5.3 Human Evaluation

The results of our human evaluation campaigns are showcased in Table 4.3. In both studies, statistical significance between system scores was assessed by making pairwise comparisons between all systems using a one-way ANOVA ($p < 0.01$) with posthoc Tukey tests with 95% confidence interval.

Informativeness. After discarding annotations that failed the controls, we are left with 229 out of 270 instances (30 documents, 3 system pairs, and 3 annotations per pair). Inter-annotator agreement –Krippendorff's alpha (Krippendorff, 2011)– was found to be 0.73.

We found that humans showed significantly higher preference for TreeKvD summaries compared to PacSum summaries, highlighting the advantage of modeling informativeness using KvD reading simulation compared to using a sentence centrality proxy in an unsupervised setup. All other system pair differences are not statistically significant.

Lexical Cohesion. We obtained 343 out of 360 summary-level annotation instances (30 documents, 4 systems –including gold summaries–, and 3 annotations per summary) after applying the control filters. In average, annotators identified 2.71 groups per summary and 3.89 propositions per group. Chain overlap, as defined in Equation 4.8, was calculated at 0.97. Score differences between system pairs TreeKvD–PacSum and GraphKvD–PacSum were found to be statistically significant, for all the analysed measurements of cohesion. Similarly, gold summary scores are significantly different from all systems in chain spread and chain density, and different from PacSum in sentence coverage.

The following insights can be drawn from these results. First, gold summaries present chains that span sentences that are either adjacent to each other or separated by one other sentence, as indicated by its chain spread scores. Chains in GraphKvD summaries mostly span adjacent sentences, in stark contrast with PacSum chains which are separated by two sentences in average. Second, chain density scores indicate that sentences in gold summaries are covered by either one or two chains, whereas GraphKvD summary sentences are covered by two chains in average. On the one hand, this indicates that KvD summaries present a smooth topic transition by linking a summary sentence to the previous one through one chain and to the following sentence though another chain. On the other hand, we note that gold summaries show lower chain density than GraphKvD summaries in average. We hypothesize that the lower chain density in gold summaries is due to the high technicality

[8] Such errors include sentence tokenization errors, incomplete equations, bibliography text included in the document, among others. Even though we re-processed the dataset, many of these errors persisted.

Criteria	TreeKvD	GraphKvD	PacSum	Gold
(I) Ranking ↓	**1.44**	1.47	1.59	-
(C) Chain Spread ↓	1.15	**1.08**	2.14	1.59
(C) Chain Density ↑	1.89	**2.29**	0.95	1.63
(C) Sent. Coverage (%) ↑	72.10	**77.33**	54.64	73.82

Table 4.3: Informativeness ranking (I) and cohesion scores (C) as a function of propositional chain properties, according to human judgements.(↑,↓): higher, lower is better.

of the scientific domain, making it harder for annotators to identify cohesive ties of non-lexical nature. Nevertheless, sentence coverage scores indicate that chains in TreeKvD and GraphKvD cover a comparable amount of sentences as chains in gold summaries. In contrast, the low chain density and low sentence coverage scores of PacSum indicate that fewer sentences (around only 54% of them) in its summaries are connected through cohesive links, the rest being perceived as isolated.

In summary, explicitly modeling lexical cohesive links during reading allows our KvD systems to extract summaries that exhibit a smooth topic transition between adjacent or near-adjacent sentences, with cohesive links connecting significantly more sentences than PacSum summaries.

4.5.4 Qualitative Analysis

We performed a qualitative analysis of system summaries extracted by the compared systems (Figure 4.7 and 4.8) by annotating the lexical chains in them and analysing the spread of chains as well as their relevance and coverage. Each sample is accompanied by its gold summary, informativeness (average ROUGE score), redundancy (RdRL), and local coherence level (CCL).

Reinforcement Learning Systems. Consider the example in Figure 4.7, showing summaries extracted by E.LG-MMRSel+, E.LG-CCL, and MemSum from a document in PubMed. First, it can be observed that the gold summary covers 6 lexical chains (all colored differently) and that these chains can appear throughout the entire text but always spanning windows of three to four sentences at a time. Note that chains spanning more than sentence implies a non-trivial level of redundancy, as showed by RdRL> 0. These smooth transitions are detected by the local coherence classifier –which scores a text by sliding a window of 3 sentences– and assigns a high CCL score.

Second, we can observe how E.LG-MMRSel+ trades off informativeness for redundancy by noting that the candidate summary exhibits one dominant chain ({miRNA expression}), possibly regarded as most promising relevancy wise. Redundancy reduction is translated in poor coverage of other chains (e.g. {miRNA}, {analysis}), being also too spread out (e.g. {biomarkers}), which is reflected in the low cohesion score of the summary. In stark contrast, E.LG-CCL exhibits most chains spreading in spans of three sentences whilst still favouring a highly relevant chain ({miRNA expression}). Note that this improvement in cohesion implied an increment in redundancy, as showed by the higher CCL score and slightly higher RdRL score.

Finally, MemSum exhibits two dominant chains ({miRNA expression} and {CAD patients}) which are highly informative, justifying the high ROUGE score of the system. However, we observed lower cohesion score compared to E.LG-CCL, which can be explained by how the chains are spread out in the summary. Whilst some chains do span adjacent sentences (e.g. the two dominant chains), others spread further (e.g. {biomarkers}, {control}). In terms of redundancy, the higher levels can be explained by the fact that chains have items with longer n-grams. This could lead to higher RdRL scores since the metric calculates the longest common n-gram subsequence in two strings. Moreover, one particular chain ({miRNA}) contains a high number of items, increasing the chance of higher lexical overlap between the sentences this chain covers.

Unsupervised Systems. Consider the example in Figure 4.8, showing summaries extracted by TextRank, TreeKvD, and GraphKvD from a highly redundant document (IUniq $= 63.34\%$) in arXiv. As observed in the previous example, the gold summary exhibits abundant lexical chains, although with varying degrees of coverage. We notice two main chains spanning the entire summary, with the rest being mentioned only once or twice. This sign of seemingly low cohesion was observed to be a common property in arXiv articles, perhaps attributed to the rather mathematical formality in the writing style, as opposed to articles in PubMed. Nevertheless, our cohesion classifier is able to pick non-lexical cues and assign a high cohesion score.

Regarding TextRank, we observe that its centrality-based scoring steer the model to focus mainly on two chains, although only one of them ended up being informative ({frequencies}). The high ROUGE scores and extremely high redundancy score confirms that centrality is a strong proxy for relevancy but without any redundancy reduction mechanism, the system will degrade into selecting repeating content. High repetition, in turn, proves to affect local coherence negatively, as indicated by the low CCL score. Most critically, TextRank is susceptible to select sentences with high –if not complete– token overlap between them, e.g. *'monopole'*, *'frequency'*, and *'ground state'*. Upon closer inspection, we found that some documents present repeated sentences in different sections, e.g. repeating a claim or conclusion.

In contrast, TreeKvD shows noticeably less repetitions and a more balanced coverage of lexical chains, as indicated by the lower redundancy score and comparable ROUGE score. Most of the chains spread consistently across the entire summary, which translates into a perceived and measured improvement in cohesion. Moreover, the system manages to recover the same two main chains present in the gold summary, and even covers short chains not covered by TextRank ({Boson}, {Stringari's result}). Upon closer inspection, we found that groups of extracted sentences are never more than two sentences apart.

Finally, GraphKvD exhibits a decrease in the spreading of lexical chains, showing instead a clear and smooth transition across the summary. This translated into an increase in local coherence, as indicated by a higher CCL score, which also impacts the redundancy score. Similarly to MemSum, the higher redundancy score can be explained by the longer common n-grams between sentences.

4.5.5 How Simulated Cognitive Processes Affect Final Summaries

The KvD theory describes cognitive processes involved in short-term memory manipulation and constraints over memory structures. While it is well understood how these processes and constraints would influence reading comprehension in a simulated environment, it is less intuitive to establish how they influence summary properties through sentence scoring. In this section, we shed light on how final summaries are affected by the following KvD processes. First, we investigate the impact of capacity in working memory and the impact of the strategy of proposition scoring used. Then, the mechanisms in charge of recall and memory replacement (tree persistance) are discussed. Finally, we investigate what kind of argument overlap strategy is best leveraged by our KvD systems.

Working Memory Capacity. Intuitively, the more memory capacity a KvD system has, the more propositions it will be able to retain in memory, increasing the chances that relevant propositions are scored higher and are eventually selected for the final summary. This is evidenced by the consistent increase in ROUGE scores for increasing memory capacity, WM, as showed in Figure 4.9. However, we did observe an optimal capacity for redundancy and cohesion levels. This indicates that, as the memory capacity increases, maintaining non-redundant information in the memory tree becomes more challenging.

Moreover, as seen in Table 4.1, KvD systems with $\mathtt{WM} = 100$ obtain consistently higher relevancy scores than FullGraph, a system that does not simulate working memory and which scoring strategy has access to all the propositions in a document at all times. This indicates that constraining the size of the memory tree in each iteration encourages KvD systems to retain only information relevant to the current local context.

Another aspect greatly influenced by working memory capacity is that of how much information in the source document can be covered. As noted in Section 4.3.3, it is possible that some propositions are pruned away and never recalled again, in which case their final score will be zero. We say that a proposition is *covered* by a KvD system if such proposition appears at least once in a pruned memory tree during simulation. Furthermore, we define document coverage as the ratio of covered propositions over the total number of propositions in a document. Not surprisingly, we found that increasing working memory capacity increased document coverage in both TreeKvD and GraphKvD. When $\texttt{WM} = 5$, TreeKvD is able to cover 62% of all document propositions in the arXiv test set, and up to 96% when $\texttt{WM} = 100$. GraphKvD further improves coverage to 78% at $\texttt{WM} = 5$ and 97% at $\texttt{WM} = 100$. However, we found that FangKvD exhibits a much lower coverage: 22% when $\texttt{WM} = 5$ and up to 44% when $\texttt{WM} = 100$. We hypothesize that the drastic improvement in GraphKvD is due to the diffusion mechanism that updates scores of direct neighbours of memory tree nodes. Similar trends were observed in the PubMed dataset. These results lay down evidence that the proposed computational implementations of KvD theory are effective at covering most –if not all- content units in a document during simulation.

So far in our analysis we have considered memory capacity as a hyper-parameter of a KvD system, expected to remain fixed throughout the entire simulation and fixed for all documents in an evaluation set. The following question then arises when looking at each sample individually: what is the *right* capacity of working memory in order to produce a summary with the most relevant content? We attempt to answer this question by selecting for each sample in the validation set, the working memory size $\texttt{WM}$ that yields the highest sum of ROUGE-1 and ROUGE-2 scores. The results are encouraging: when using the best possible $\texttt{WM}$ per sample in arXiv, TreeKvD exhibits an increase in absolute points of 3.19 in ROUGE-1, 2.36 in ROUGE-2, and 2.86 in ROUGE-L. This is compared to the best performing configuration, i.e. when using $\texttt{WM} = 100$ for all samples. Most surprisingly, the distribution of best $\texttt{WM}$ per sample is rather balanced, with 26.5% of samples preferring a $\texttt{WM} = 100$, 26.7% a $\texttt{WM} = 50$, 24.11% a $\texttt{WM} = 20$, and 22.5% a $\texttt{WM} = 5$. GraphKvD exhibits a similar increase of 3.06, 2.33, 2.75 in ROUGE-1, ROUGE-2, and ROUGE-L, respectively. A similar trend was observed on the validation set of PubMed. However, it should be noted that we did not find any strong correlation between working memory capacity and ROUGE or BertScore scores, which indicates that the ability of a KvD system to produce relevant summaries is not influenced by its working memory capacity. Instead, we suspect that memory capacity might be an indicator of text difficulty or cognitive easiness, however the exploration of this hypothesis falls out of the scope of this work and we leave it to future investigations.

Working Memory as a Tree. Next, we investigated the impact of leveraging the position of a node in the memory tree structure during proposition scoring. We compared scoring function $c(\cdot)$ in Eq. 4.6, labeled as TREE, against two other strategies. The first one, denoted FREQ, consists of a frequency heuristic, $c(t,T) = 1, \forall t \in T$, which only counts how many memory cycles a proposition participates in. The second strategy, denoted EIGEN, scores nodes based on their eigen-vector centrality as:

$$c(t,T) = \frac{1}{\lambda} \sum_{\substack{v \\ \text{s.t. } (t,v) \in E[T]}} c(v,T)$$

where λ is the largest eigen-value of the adjacency matrix of T.[9]

Figure 4.9 shows the performance of our KvD systems over the validation set of PUBMED and ARXIV. Systems using scoring function $c(t,T)$ in Eq. 4.2 are labeled with TREE, e.g. TREEKVD[TREE]. First, we observe that TREE scoring significantly outperforms EIGEN and FREQ scoring, for all values of working memory capacity in both datasets. This results demonstrates the advantage of modeling memory as a tree structure and leveraging the position of a node for scoring, compared to just considering memory as a bag of content units (as FREQ does) or even using node centrality strategies, as done by EIGEN. However, it is worth noticing that for GRAPHKVD, the gap between TREE and EIGEN diminishes as WM increases, even performing comparably in PUBMED. This might indicate that GRAPHKVD is superior than TREEKVD at placing highly influential (i.e. relevant) nodes closer to the root, in which case the proposition ranking given by TREE and EIGEN is highly similar.

In conclusion, TREE scoring enables our implementations of KvD not only to better keep track of relevant information but also to better model cohesion in the memory tree, which translates to lower redundancy scores and higher cohesion scores in final summaries.

Recall Mechanism and Tree Persistence. Additionally, we investigated the effect of allowing our KvD systems to retrieve longer node paths during recalls, as well as the effect of allowing systems to persist memory trees for more cycles. Whilst (Kintsch and van Dijk, 1978) do not define a limit for how many propositions can be recalled, (Fang, 2019) limits recall to only one proposition for computational efficiency. In this experiment, we test TREEKVD and GRAPHKVD with $\texttt{WM} = 100$ and TREE scoring, and set the maximum al-

[9] We use the eigen-vector centrality implementation in the NetworkX Python library.

System	PubMed					arXiv				
	R1	R2	RL	IUniq	CCL	R1	R2	RL	IUniq	CCL
TreeKvD										
w/ Lex. Overlap	35.93	12.63	31.53	19.05	0.53	35.40	9.84	30.08	22.36	0.46
w/ XLNet	35.60	13.53	31.39	18.79	0.57	34.47	9.74	29.29	21.94	0.46
GraphKvD										
w/ Lex. Overlap	36.11	12.97	31.65	19.49	0.49	35.60	10.12	30.14	22.56	0.38
w/ XLNet	35.75	12.66	31.34	19.02	0.51	34.77	9.37	29.32	22.55	0.38
Gold	-	-	-	18.94	0.92	-	-	-	17.15	0.89

Table 4.4: Effect of using lexical overlap and semantic similarity in argument overlap calculation, as measured by ROUGE F_1 scores, redundancy (IUniq), and coherence (CCL), over the validation sets of PUBMED and ARXIV.

lowed number of recalled nodes to $R = [2, 5, 8, 10]$ and the maximum persistence parameter to $\Phi = [2, 5, 8, 10]$. When compared in the validation set of both datasets, no statistical difference was found within TREEKVD and GRAPHKVD varieties. Absolute differences in average ROUGE scores were at most 0.1, whereas differences in IUniq redundancy were at most 0.2 percentual points. These results indicate that our implementations of the KvD theory are robust to recall and memory replacement parameters, an encouraging result when planning to use these systems in other domains.

Lastly, it is worth pointing out an additional benefit of the tree persistence mechanism, observed empirically in Figure 4.9. Tree persistence can be seen as a mechanism that guarantees that the content in WM changes periodically, providing the model with robustness to the length of an article section in a scientific article, and adding evidence to its applicability to other domains. As mentioned in the previous chapter, sections in PUBMED articles are shorter than those in ARXIV (16.8 vs 28.8 on average). In PUBMED, performance converges at $\mathtt{WM} = 150$, at which point there is enough capacity to keep all propositions read in the section so far. However, contrary to the behavior of FANGKVD in the previous chapter, performance is not hurt at high capacity regimes, with the persistence mechanism refreshing WM periodically. In ARXIV, sections are long enough for high WM capacity to be a problem, at which point WM starts storing noisy information which eventually hurts performance.

Effect of Argument Overlap Strategy. Finally, we investigated the effect of employing more sophisticated strategies to calculate argument overlap in propositions. We compared our proposed strategy –based in lexical overlap– against a strategy using a pretrained Transformer-based encoder (Vaswani et al., 2017) to calculate semantic similarity. We replace the Jaccard similarity between two arguments in Eq. 4.4 by the maximum pairwise cosine similarity between wordpiece embeddings of said arguments. Each sentence is encoded independently using XLNet (Yang et al., 2019) with the previous three sentences as context. Recent work (Jeon and Strube, 2020, 2022) showed the advantage of using XLNet against other Transformer-based architectures when modeling local coherence in contexts a few sentences long.[10]

Table 4.4 presents the results for TreeKvD and GraphKvD. In both cases, we observe a reduction of relevancy and redundancy scores when using embedding-based similarity in argument overlap. In PubMed, both KvD systems obtain higher cohesion scores with XLNet, whilst cohesion remains unchanged in arXiv. These results indicate that employing semantic similarity in argument overlap hurts informativeness in greedily selected summaries, in line with similar findings by Fang (2019).

We hypothesize that employing embedding-based similarity allows to connect arguments that are not semantically related but might be close in embedding space, hence resulting in spurious proposition connections during attachment. Naturally, with memory trees polluted with irrelevant propositions, KvD systems struggle to keep track on truly relevant information and informativeness will be impacted.

In conclusion, this section laid evidence as to how simulated cognitive processes impact properties (informativeness, redundancy, and cohesion) of the final summary. First, we pointed out the importance of constraining memory capacity in covering relevant content and dealing with redundant information. Then, we highlighted the benefits of modeling working memory as a tree and how this affects the cohesion-redundancy trade-off. We demonstrated the robustness of the proposed systems to parameters controlling recall from long-term memory. Finally, the sensitivity of the systems to spurious connections between propositions was assessed, and demonstrated that limiting connections through selective lexical overlap provides the best conditions for our systems to better balance informativeness, redundancy, and lexical cohesion in summaries.

4.6 Summary

In this chapter, we studied the trade-off between redundancy and lexical cohesion in summaries produced by extractive systems, and how this trade-off impacts informativeness. We focused on the case when the input is a long document that exhibits information redundancy among the parts it is divided into. As a case study, we experimented with scientific articles

[10] Indeed, preliminary experiments using SciBERT (Beltagy et al., 2019) showed poor results.

for which the main body –divided into sections– is considered as the input document and the abstract is used as the reference summary.

Two optimization scenarios were investigated and compared, (i) when a summary property is optimized with a tailored reward in a reinforcement learning setup, and (ii) when a summary property is optimized through proxies inspired by a psycholinguistic model in an unsupervised setup. In the first scenario, the trade-off between informativeness and cohesion was modeled as a linear combination between a reward optimizing for ROUGE score w.r.t. the reference summary and a classifier-based reward optimizing for cohesion. We found that models that optimize cohesion are capable of better organizing content in summaries compared to systems that optimize redundancy, whilst maintaining –if not improving– informativeness and coverage.

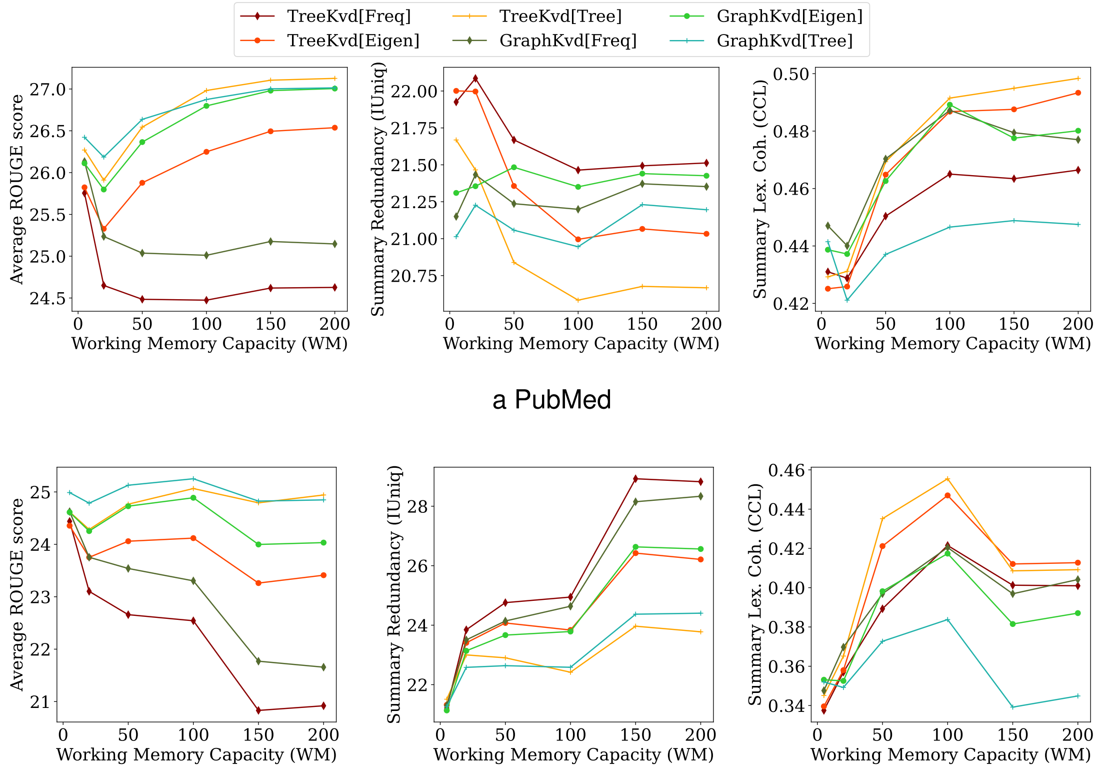
In the second scenario, we introduced two unsupervised summarization systems that implement explicit proxies that capture relevancy, non-redundancy, and lexical cohesion. The proposed systems closely simulate human memory during KvD reading. Extensive quantitative and qualitative analysis showed that our systems are able to extract summaries that are highly cohesive and as redundant as reference summaries, however at the expense of sacrificing informativeness. Finally, human evaluation campaigns revealed that KvD summaries exhibit a smooth topic transition between sentences as signaled by proposition chains –an extension to lexical chains–, with chains spanning adjacent or near-adjacent sentences, and each sentence being connected to a previous one with at least one chain and to the next sentence with another chain.

System	Avg. ROUGE	RdRL	CCL
Gold Summary Coronary artery disease (CAD) is the largest killer of males and females in the United States. There is a need to develop innovative diagnostic markers for this disease. MicroRNAs (miRNAs) are a class of noncoding RNAs that posttranscriptionally regulate the expression of genes involved in important cellular processes, and we hypothesized that the miRNA expression profile would be altered in whole blood samples of patients with CAD. We performed a microarray analysis on RNA from the blood of 5 male subjects with CAD and 5 healthy subjects (mean age 53 years). Subsequently, we performed qRT-PCR analysis of miRNA expression in whole blood of another 10 patients with CAD and 15 healthy subjects. We identified 11 miRNAs that were significantly downregulated in CAD subjects ($p < .05$). Furthermore, we found an association between ACEI/ARB use and downregulation of several miRNAs that was independent of the presence of significant CAD. In conclusion, we have identified a distinct miRNA signature in whole blood that discriminates CAD patients from healthy subjects. Importantly, medication use may significantly alter miRNA expression. These findings may have significant implications for identifying and managing individuals that either have CAD or are at risk of developing the disease.	-	12.6	0.86
E.LG-MMRSel+ We sought to compare miRNA expression in whole blood of patients with angiographically significant CAD to that of healthy aged-matched controls. We performed an initial exploratory microarray analysis in 5 cases and controls and then further examined the most highly expressed miRNAs in an additional 15 cases and controls. The present study provides insight into whole blood levels of miRNAs in patients with CAD compared to healthy subjects and demonstrates their potential utility as biomarkers for vascular disease. Thus, miRNA expression signatures in tissues and blood have a potential role in the diagnosis, prognosis, and assessment of therapy. Study participants were recruited as part of the Emory Cardiology Biobank, consisting of 3492 consecutive patients enrolled prior to undergoing elective or emergent cardiac catheterization across three Emory Healthcare sites, between 2003 and 2008. Validation of the changes in miRNA expression observed here in larger studies will be a necessary step to confirm their candidacy as biomarkers and therapeutic targets. Although we were able to detect some differences in whole blood miRNA levels between healthy subjects and CAD patients (miR-584, in particular), our microarray data suggest that, similar to other reports, levels of miRNAs in the blood are low and microarrays may lack the sensitivity to adequately identify miRNAs that might serve as vascular disease biomarkers.	31.60	16.13	0.20
E.LG-CCL Thus, miRNA expression signatures in tissues and blood have a potential role in the diagnosis, prognosis, and assessment of therapy. In this study, we sought to compare miRNA expression in whole blood of patients with angiographically significant CAD to that of healthy aged-matched controls. We performed an initial exploratory microarray analysis in 5 cases and controls and then further examined the most highly expressed miRNAs in an additional 15 cases and controls. Study participants were recruited as part of the Emory Cardiology Biobank, consisting of 3492 consecutive patients enrolled prior to undergoing elective or emergent cardiac catheterization across three Emory Healthcare sites, between 2003 and 2008. Although we were able to detect some differences in whole blood miRNA levels between healthy subjects and CAD patients (miR-584, in particular), our microarray data suggest that, similar to other reports, levels of miRNAs in the blood are low and microarrays may lack the sensitivity to adequately identify miRNAs that might serve as vascular disease biomarkers. The present study provides insight into whole blood levels of miRNAs in patients with CAD compared to healthy subjects and demonstrates their potential utility as biomarkers for vascular disease. Validation of the changes in miRNA expression observed here in larger studies will be a necessary step to confirm their candidacy as biomarkers and therapeutic targets.	31.82	18.90	0.60
MemSum We sought to compare miRNA expression in whole blood of patients with angiographically significant CAD to that of healthy age-matched controls. This analysis included miR-150, miR-584, miR-21, miR-24, miR-126, miR-92a, miR-34a, miR-19a, miR-145, miR-155, miR-222, miR-378, miR-29a, miR-30e-5p, miR-342, and miR-181d. Among these, we found that miR-19a, miR-584, miR-155, miR-222, miR-145, miR-29a, miR-378, miR-342, miR-181d, miR-150, and miR-30e-5p were significantly downregulated in the blood of patients with CAD compared to healthy subjects (Figure 2). Several recent studies have indicated that there is a potential role for circulating miRNA levels as valuable biomarkers for different disease processes, including cancer, cardiomyopathy, and acute myocardial infarction. In this study, we wanted to address the hypothesis that miRNA expression levels in blood could predict the presence of significant coronary artery disease in human subjects. We identified 11 miRNAs whose expression was significantly downregulated in patients with angiographic evidence of significant atherosclerosis compared to healthy subjects that were matched for age and gender. The present study provides insight into whole blood levels of miRNAs in patients with CAD compared to healthy subjects and demonstrates their potential utility as biomarkers for vascular disease.	34.16	22.34	0.33

Figure 4.7: Summaries extracted by reinforcement learning-based systems for a PubMed sample with informativeness (average ROUGE score), redundancy (RdRL), and local coherence (CCL) scores. Text is annotated with color-coded lexical chains, and was detokenized and truecased for ease of reading.

<table>
<tr><th>System</th><th>Avg. ROUGE</th><th>RdRL</th><th>CCL</th></tr>
<tr><td><b>Gold Summary</b></td><td>-</td><td>21.49</td><td>0.91</td></tr>
<tr><td colspan="4">We study the collective excitations of a neutral atomic Bose-Einstein condensate with gravity-like interatomic attraction induced by electromagnetic wave. Using the time-dependent variational approach, we derive an analytical spectrum for monopole and quadrupole mode frequencies of a gravity-like self-bound Bose condensed state at zero temperature. We also analyze the excitation frequencies of the Thomas-Fermi gravity (tf-g) and gravity (g) regimes. Our result agrees excellently with that of Giovanazzi et al., which is obtained within the sum-rule approach. We also consider the vortex state. We estimate the superfluid coherence length and the critical angular frequencies to create a vortex around the X axis. We find that the tf-g regime can exhibit the superfluid properties more prominently than the g regime . We find that the monopole mode frequency of the condensate decreases due to the presence of a vortex.</td></tr>
<tr><td><b>TextRank</b></td><td>38.99</td><td>45.02</td><td>0.23</td></tr>
<tr><td colspan="4">The gravity-like potential is balanced by the wave interaction strength. The ground state energy per particle varies as @xmath. The monopole and quadrupole frequencies obtained from the variational approach are similar to the exact numerical values. The trap potential and wave interaction can be neglected. The total ground state energy is @xmath. The ground state energy per particle varies as @xmath. One can use the time-dependent variational approach to describe the vortex state. The critical angular frequency vs. the dimensionless scattering parameter is shown in Fig.4. Tf-g regime: for large wave scattering length, kinetic energy can be neglected. The critical angular frequencies for @xmath and @xmath are @xmath and @xmath respectively. The monopole mode frequency for an ordinary atomic bec in the tf regime is independent of the vortex. The monopole mode frequency for @xmath is @xmath. The @xmath is also less than the monopole mode frequency in the vortex free condensate. In the tf regime of an ordinary atomic bec, the monopole and quadrupole mode frequencies are independent of the scattering length.</td></tr>
<tr><td><b>TreeKvD</b></td><td>39.87</td><td>14.62</td><td>0.36</td></tr>
<tr><td colspan="4">In this system, the gravity-like attraction balances the pressure due to the zero point kinetic energy and the short range interaction potential. The bec of charged Bosons confined in an ion trap can be described by the above mentioned Lagrangian if we set @xmath, where @xmath is the electronic charge. To calculate the excitations spectrum of an atomic bec with gravity-like interaction, we will use the time-dependent variational method. This technique has been first used to calculate the low-lying excitations spectrum of a harmonically trapped atomic bec in @xref. The result obtained from the variational method matches with Stringari’s result within the sum-rule approach. In @xref, it is shown that the oscillation frequencies obtained from the exact ground state and a Gaussian Ansatz are in good agreement. One can use the time-dependent variational approach to describe the vortex state. In these regimes, we have calculated the lower bound of the ground state energy, sound velocity, monopole and quadrupole mode frequencies.</td></tr>
<tr><td><b>GraphKvD</b></td><td>39.73</td><td>21.65</td><td>0.51</td></tr>
<tr><td colspan="4">Most of the properties of these dilute gas can be explained by considering only two-body short range interaction which is characterized by the S-wave scattering length. Therefore, we expand around the time dependent variational parameters around the equilibrium widths in the following way, and @xmath. The time evolution of the widths around the equilibrium points are @xmath is the first order fluctuations around the equilibrium points of @xmath. One can use the time-dependent variational approach to describe the vortex state. The vortex state play an important role in characterizing the superfluid properties of Bose system. The critical angular frequency required to produce a vortex state is where is the energy of a vortex states with vortex quantum number and is the energy with no vortex. In these regimes, we have calculated the lower bound of the ground state energy, sound velocity, monopole and quadrupole mode frequencies.</td></tr>
</table>

Figure 4.8: Summaries extracted by unsupervised systems for an ARXIV sample with informativeness (average ROUGE score), redundancy (RdRL), and local coherence (CCL) scores. Text is annotated with color-coded lexical chains, and was detokenized and truecased for ease of reading.



a PubMed

b arXiv

Figure 4.9: Effect of proposition scoring strategy (TREE, EIGEN, and FREQ) and working memory capacity (WM) on summary informativeness (average ROUGE scores; left), redundancy (IUniq; middle), and local coherence (CCL).

Chapter 5

Trade-off Control during Summary Extraction

Following on our work on trade-off control of summary properties, we now turn our attention to control during sentence selection. In this chapter, we aim to enforce cohesion whilst controlling for informativeness in summaries, in cases where the input exhibits high redundancy. The pipeline controls for content redundancy in the input as it is consumed, and balances informativeness and cohesion during sentence selection. Our sentence selector simulates human memory to keep track of topics –modeled as lexical chains– while building the summary, enforcing cohesive ties between noun phrases. Extensive experiments, both automatic and human, revealed that it is possible to extract highly cohesive summaries without sacrificing informativeness significantly, compared to summaries optimizing only for informativeness. The extracted summaries exhibit smooth topic transitions between sentences as signaled by lexical chains, with chains spanning adjacent or near-adjacent sentences.

5.1 Introduction

In previous chapters, we showcased the challenges of selecting the appropriate content units so that the summary covers relevant topics, or to control trade-offs between summary properties during document understanding. In this chapter, we focus on modeling and controlling summary properties during summary production, focusing on informativeness and cohesion. As discussed in Chapter 1, the modeling of summary coherence previously relied on capturing discourse patterns in nearby sentences (Barzilay and Lapata, 2008; Steen and Markert, 2022; Zhao et al., 2023). Cohesion, a special case of local coherence, relies on the explicit textualization of contextual connections called *cohesive ties*, making a text read as a unified whole (Hassan et al., 1976).

In this chapter, we introduce an extractive summarization methodology that implements two control mechanisms at different stages of processing: the first one to control redundancy during input understanding, and the second one to control the trade-off between informativeness and cohesion during summary extraction. When building extractive summaries by concatenating sentences, we argue that controlling for cohesion is a better-defined task than aiming to control coherence, especially if no sort of post-editing (e.g. replacing discourse markers) is applied (Zajic et al., 2007; West et al., 2019; Mallinson et al., 2020). A potential benefit of producing a more cohesive text is that it is easier to read and understand for humans, especially when the knowledge domain is highly technical, as reported by previous work in psycholinguistics (Kintsch, 1990) and automatic summarization (Barzilay and Elhadad, 2002).

In our pipeline, summary properties are controlled in the following way. On the one hand, summary redundancy is addressed by controlling the redundancy levels of the input text, following previous findings (Carbonell and Goldstein, 1998b; Xiao and Carenini, 2020). The pipeline consumes input text in a cascaded way: first splitting the input into contiguous passages, then consuming passages one at a time so as to minimize their semantic similarity with already selected passages.

On the other hand, informativeness and cohesion are directly modeled during summary extraction. Extraction is done in a sentence-by-sentence fashion, quantifying summary properties independently at each step. The objective is to select a highly cohesive sentence that is informative enough. We introduce a sentence selector that incrementally builds cohesive chains of noun phrases and models chain interaction. The selector, KvD-Select, keeps track of chains currently active by simulating KvD *production*, i.e. the cognitive processes involved in the handling of human memory during text production. Contrary to previous chapters, working memory is modeled as a limited-capacity buffer of lexical chains, forcing the model to keep only the most salient chains and send the rest to long-term memory.

We test our methodology on newswire multi-document summarization and single-long document summarization of scientific articles, patents, and government reports. Across domains, extensive experiments show that, first, our system is effective at incrementally building an input sequence with lower content redundancy, which translated to a significant reduction in summary redundancy. Second, the proposed sentence selector managed to maintain summaries informative while improving cohesion significantly: over 15% more noun phrases and over 20% more sentences were connected through cohesive ties w.r.t a greedy selector. Tailored human evaluation campaigns revealed that cohesion has a positive impact on perceived informativeness, and that our extracted summaries exhibit chains covering adjacent or near-adjacent sentences. Closer inspection showed that topics flow smoothly across extracted summaries with no abrupt change or jumps.

In summary, the contributions of this chapter are as follows:

- We propose a cascaded encoder capable of consuming arbitrary long textual input that controls the level of content redundancy the rest of the pipeline is exposed to.
- We propose a summary extraction method that models informativeness and cohesion independently and allows to control the balance between the two when building the summary.
- Automatic and human experiments show the effectiveness of our control mechanisms and how summary properties can be balanced according to user needs in a straightforward way.

5.2 Problem Setup

Continuing the formulation presented in previous chapters, we tackle the task of extractive summarization as a sentence-scoring step followed by a selection step. Figure 5.1 shows the pipeline of the system, in which sentences are scored in a cascaded fashion, as follows. First, the input is segmented into blocks of contiguous sentences to be selected based on their relevancy and their redundancy w.r.t. already selected blocks. Then, a local encoder obtains block-level representations for each sentence in the block. After all document blocks are processed, the encodings are concatenated into a single embedding sequence and passed to the global context encoder, which will obtain a document-aware representation of each sentence. Finally, a selection module will extract a subset of sentences and present them as the summary in the order they were extracted. The pipeline is designed to be capable of consuming documents of arbitrary length, offering further control over levels of information redundancy the sentence selector is exposed to. We now proceed to elaborate on each module of the proposed pipeline.

Document Segmentation and Block Selection. Processing starts by segmenting the input document(s) D into fixed-length overlapping blocks, each of which includes preceding and subsequent wordpieces in order to provide surrounding context. Then, blocks are selected iteratively until a predefined budget (e.g. total number of wordpieces) is met. At step m, the optimal block selection is defined as the trade-off between a block relevancy term and

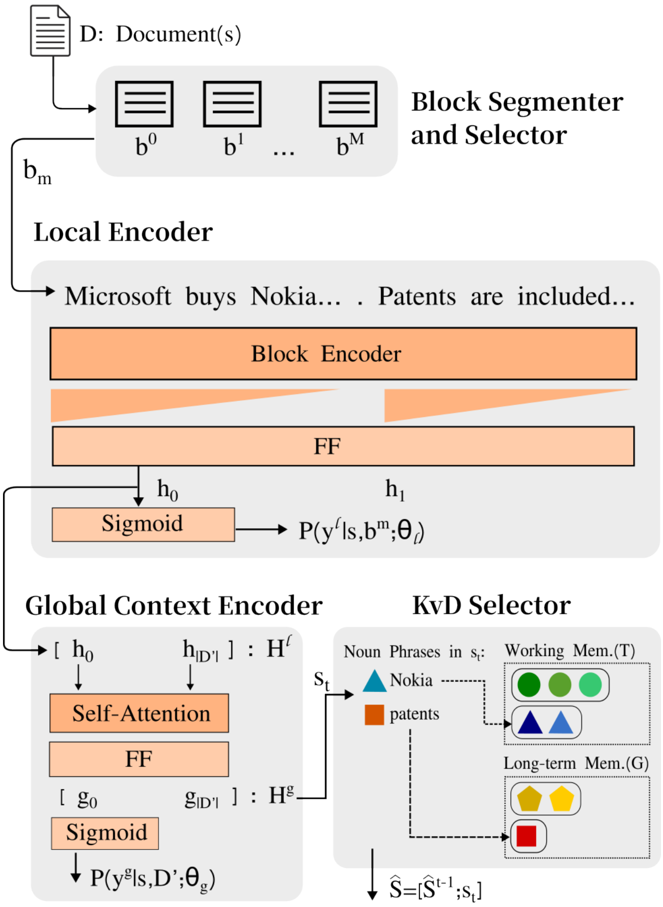


Figure 5.1: Extraction pipeline of the proposed system. Input D is consumed one block at a time. At block selection step m, the local encoder adds at most N local sentence embeddings from block b_m to the global sentence sequence D'. After the whole input has been consumed, the summary extractor module builds $\hat{S}$ one sentence at a time. For KvDSelector, the selector simulates one KvD memory cycle at each sentence selection step i.

a redundancy term,

$$b_m = \operatorname*{argmax}_{b \in B \setminus \hat{B}} [\lambda_b LR(b) - (1 - \lambda_b) \max_{b_j \in \hat{B}} Sim(b, b_j)] \tag{5.1}$$

where $\hat{B}$ is the set of blocks already selected, $Sim(x, y)$ is the cosine similarity between TF-IDF vectors of blocks x and y, and hyper-parameter λ_b allows to control the mix of both terms. Function $LR(b)$ represents the continuous LexRank score of block b (Erkan and Radev, 2004) obtained when modeling D as a complete graph in which each node is a block and edges quantify TF-IDF similarity between blocks, and calculating the centrality of each node with the PageRank algorithm (Page, 1998). Formally,

$$\mathrm{LR}(b) = \frac{d}{|B|} + (1-d) \sum_{v \in \mathrm{adj}[b]} \frac{\mathrm{Sim}(b,v)}{\sum_{z \in \mathrm{adj}[v]} \mathrm{Sim}(z,v)} \mathrm{LR}(v) \tag{5.2}$$

where d is the damping factor and $\mathrm{adj}(b)$ is the set of block nodes adjacent to b. In this way, this module provides a straightforward way to balance block relevancy (as proxied by centrality) and input redundancy by linearly combining quantifications of these properties. After an optimal block is selected, it is send to the local extraction module.

Local Encoder (LE). Given block b as a sequence of wordpieces spanning contiguous sentences, the local encoder will obtain representations for each of the sentences covered in b. This module is trained as a local extractive summarizer itself, in order to obtain sentence representations tailored for the task. We posit the local extraction task as a sequence labeling task where each sentence in the block is labeled as $y_i^\ell = \{0,1\}$ to indicate whether sentence s_i should be selected or not. Then, sentence representation h_i is defined the average embedding over wordpiece embeddings in s_i obtaining from a LongT5 encoder (Guo et al., 2022). Finally, the probability of s_i being selected is defined as $P(y_i^\ell | s_i, b; \theta_\ell) = \sigma(W^\ell \cdot h_i)$, and the module is trained using cross-entropy loss independently from the rest of the pipeline. During inference, the local encoder consumes one block, selects N sentences and adds them to D' –containing all locally selected sentences so far–, and their corresponding embeddings to H^ℓ.

Global Context Encoder (GCE). Given the sequence of local sentence embeddings H^ℓ, this module obtains the sequence of globally-aware representations H^g as follows. Each local embedding in H^ℓ is passed through a self-attention layer (Vaswani et al., 2017), i.e. $g_i = \mathrm{SelfAttn}(h_i, H^\ell), \forall h_i \in H^\ell$. Similarly to the local extraction module, the summary-worthiness of sentence s_i is modeled as $P(y_i^g \mid s_i, D'; \theta_g) = \sigma(W^g \cdot g_i)$, where $y_i^g \in \{0,1\}$ indicates whether s_i is selected or not for the final, global summary, and also trained using cross-entropy loss.

Summary Extractor. Finally, candidate summary $\hat{S}$ is built by selecting one sentence at a time from D', taking into account the informativeness and cohesiveness of each candidate sentence w.r.t. the already selected sentences. At selection step t, the optimal sentence is given by

$$s_t = \underset{s \in D' \setminus \hat{S}^{t-1}}{\mathrm{argmax}} \; \lambda_{\mathrm{sel}} f_I(s) + (1-\lambda_{\mathrm{sel}}) f_C(\hat{S}^t) \tag{5.3}$$

where function f_I estimates the informativeness of candidate sentence s, f_C estimates the cohesion of candidate summary $\hat{S}^t = [\hat{S}^{t-1}; s]$, and $\lambda_{\mathrm{sel}} \in [0,1]$ is a parameter that allows to control their trade-off. Following previous work (Xiao and Carenini, 2020), we take the probability of selecting s given by the global context encoder module as a proxy for informativeness, i.e. $f_I(s) = P(y^g \mid s, D'; \theta_g)$. In the next section, we elaborate on how f_C models and enforces cohesion during sentence selection.

5.3 KvD Select: Cohesion during Summary Extraction

Cohesion is a language mechanism that enables a sequence of sentences to function as a unified whole (Hassan et al., 1976). It does so by linking semantic units in a text through *cohesive ties*, regardless of the grammatical or discourse structure these units are part of. In particular, lexical cohesion links units with the same lexical form, synonyms, or units in the same semantic field. Furthermore, units tied cohesively can be grouped in chains by their semantic similarity. Whilst the mere presence of two or more chains does not guarantee a cohesive effect, their interaction can be a reliable proxy for cohesion (Morris and Hirst, 1991; Barzilay and Elhadad, 1997). In this chapter, we focus on modeling lexical cohesive ties between noun phrases in nearby sentences of a summary by controlling the interaction between lexical chains.

The proposed selector, KvD-Select, calculates cohesion score f_C by simulating the processes in working memory during text production according to the KvD theory. Similarly to previous chapters, we implement processes happening at the micro-level, which deal with the movement of content in and out of working memory.

Let T be working memory and G long-term memory (LTM), where both are separate sets of cohesive chains, and each chain as a set of noun phrases (NPs). At selection step t, the algorithm extracts NPs from s_t and connects them to the chains in T and G, constraining the number of active chains in T afterward. Cohesion score f_C then depends on the average similarity between units added to T and those added to G. We now elaborate on each step of the algorithm.

Extracting Noun Phrases. Given sentence $s_t \in D'$, we obtain P, the set of extracted nominal chunks, obtained by merging nominal nodes in dependency trees with their children. Specifically, given that node *u* is nominal dependent of a clausal predicate, *u* will have its child *v* merged if either*v* is a function word, a single-token modifier, or *u* and *v* form part of a multi-word expression.

Adding Content to Memory. Next, cohesive ties between s_t and $\hat{S}^{t-1}$ are enforced by adding each NP in P to the chain with the highest element-wise semantic similarity. Formally, the optimal chain to add $a \in P$ to is $C^* = \text{argmax}_{C \in T}\{\phi(p, C)\}$, where ϕ is the average BERTScore (Zhang et al., 2019) between a and each NP in C. In order to make sure that chains maintain an acceptable level of semantic similarity between elements, a is added to chain C only if $\phi(a, C) \geqslant \nu$, where ν is the minimum admissible similarity. This way the algorithm can control the similarity length between chain members, and avoid a single, long chain.

If similarity with chains in T is not strong enough, we look at chains in G, in which case the chosen chain is moved back to T. This step is analogous the the recall mechanisms implemented in previous chapters. If still no chain in G meets the similarity requirement, we proceed to create a brand new chain in T with a as its sole element. By searching for a good enough candidate chain first in T and then in G, we encourage cohesive ties between NPs in nearby sentences.

Updating Memory. After adding incoming NPs to chains in memory, T is updated to retain only the WM most recent chains, where *recency* of a chain is defined as the id of the selection step in which this chain was last retained in T. For instance, a chain currently in T is more recent (higher step id) than a chain in G discarded in an earlier step. This design choice mimics the *recency effect* behaviour during *free recall* tasks in human subjects (Glanzer, 1972), a behaviour attributed to short-term memory. Finally, discarded chains are moved to G, concluding the selection step.

Candidate Scoring. Next, we define cohesion score f_{coh} which will be used to discriminate amongst possible continuations to $\hat{S}^{t-1}$. The objective is to encourage NPs in P to be assigned to recent chains, in turn encouraging chains to cover nearby sentences in the final summary. In addition, we want to score down candidate sentences with NPs added to chains in long-term memory.

Let $A_T = \{a; a \in P, C_a \in T\}$, where C_a is the chain a was added to. Similarly, let $A_G = \{b; b \in P, C_b \in G\}$. Then, let $\mathrm{rec}(C)$ be the number of selection steps passed since the last time chain C was retained in T. Quantity $\mathrm{rec}(C)$ functions as a proxy for how spread chain C is, i.e. how far away two sentences covered by C are. Then,

$$f_{coh} = \frac{1}{|A_T|} \sum_{a \in A_T} \frac{\phi(a, C_a)}{\mathrm{rec}(C_a)} + \frac{\gamma_{rec}}{|A_G|} \sum_{b \in A_G} \frac{\phi(b, C_b)}{\mathrm{rec}(C_b)}. \tag{5.4}$$

Hence, the cohesive score depends on the contribution of each cohesive tie formed. For each chunk in A_T and A_G, its contribution depends directly on the strength of similarity to its assigned chain and inversely on the spread of said chain. The contribution of chunks in A_G is scaled down by hyper-parameter $\gamma_{rec} \in [0; 1]$ as to simulate the higher cognitive cost incurred when retrieving information from long-term memory.

5.4 Experimental Setup

In this section, we describe the datasets employed in our experiments, the hyper-parameters and training details of our pipeline, comparison systems, and evaluation methodology, both automatic and human-based.

5.4.1 Datasets

Our analysis includes datasets for single-document summarization of long, highly redundant documents as well as multi-document summarization in a variety of domains, as follows.

- **PubMed.** Scientific articles in the biomedical domain collected from PubMed (Cohan et al., 2018). As in Chapter 3(REF), we employ text from the '`sections`' field as input document We use text from all sections as the source document and the abstract as reference summary.
- **BigPatent.** Patents in several industry domains (Sharma et al., 2019). We restrict our analysis to the Chemistry and Metallurgy domain (subset 'C').
- **GovReport.** Long legislature reports (Huang et al., 2021) of U.S. bills summarized by experts.
- **MultiNews.** Consisting of collections of news articles in a topic paired with human-written summaries (Fabbri et al., 2019).

For all datasets, we homogenize the source-target length distributions by discarding samples with references that were too short (less than 3 sentences, not useful for our cohesion analysis) or too long (more than 500 tokens in all datasets except GovReport, for which this threshold is set to 1000). Similarly, samples with short input documents (less than 3 sentences or less than 30 tokens in total) were also discarded. Sentences were re-split using spaCy[1] and trimmed to 100 tokens, whilst sentences with less than 5 tokens were discarded. Table 5.1 presents the statistics of all dataset in terms of number of tokens.

5.4.2 Pipeline Parameters

Hyper-parameters were tuned over the validation sets of each dataset. See Table A.1 (Appendix A.1) for a comprehensive list of hyper-parameter values, including word budgets and architecture details.

[1] `https://spacy.io/`

Dataset	Input Length			Target Len.
	Avg.	Max.	Q90	Avg.
PubMed	3150	119875	5844	206
BigPatent.C	4534	72835	8655	119
GovReport	8840	206622	15752	580
MultiNews	2057	525348	3846	260

Table 5.1: Dataset statistics in terms of number of tokens showing average, maximum, and 90% quantile (Q90).

Document Segmentation and Block Selection. During document segmentation, we use a block size of $B = 2048$ and context size $C = 200$ pieces. During block selection, we set $\lambda_b = 0.2$ for both datasets after finetuning it over their respective validation set. Finally, we set a budget of 1000 sentences or 16 384 wordpieces in order to make our analysis comparable to previous work in long-document summarization (Guo et al., 2022; Beltagy et al., 2020).

Local Encoder and Global Context Encoder. The block encoder in LE is initialized with a pretrained checkpoint of LongT5 with transient-global attention (Guo et al., 2022),[2] and an output layer of size 200.

The LE module is trained independently from the GCE module, with LE being trained first, then GCE trained whilst LE remains frozen. In both cases, we used the Adam optimizer (Loshchilov and Hutter, 2019), a constant learning rate of $1e^{-6}$, effective batch size of 64, and 50k training steps. During inference, we extract a maximum of $N = 10$ local sentences per block and a maximum of 1000 sentences in total.

Summary Extractor. We set $\lambda_{sel} = 0.8$, working memory $\texttt{WM}{=}\ 6$, recall cost $\gamma_{rec} = 0.01$, and a minimum NP similarity of $\nu = 0.6$. Word budget is set to 200, 100, 650, 250 for PubMed, BigPatent.C, GovReport, MultiNews, respectively.

5.4.3 Comparison Systems

We compare against the standard extractive oracle, Ext-Oracle, obtained by greedily selecting sentences maximizing ROUGE-1 + ROUGE-2 F_1 against gold summaries until the word budget is met. For cohesion analysis, we also report metric values over the gold summaries, labeled as Gold.

[2] HuggingFace, `google/long-t5-tglobal-base`

The impact of cohesion modeling is assessed by employing a greedy selector over GCE scores, equivalent to set $f_C = 0$ in Eq. 5.3, dubbed LT5-Casc. Similarly, our cascaded extraction approach is contrasted against a LongT5 encoder and topped with a classification layer that consumes the whole input at once, dubbed LT5-Flat. Moreover, similarly to previous chapters, we do not include supervised baselines that require the calculation of coreference chains or rhetorical structure trees over the input document, such as DiscoBERT (Xu et al., 2020), because of their limited applicability in out-of-domain scenarios and their inability to process documents of the length analyzed in this chapter.

Finally, the complete LongT5 encoder-decoder architecture is reported as an abstractive baseline, dubbed LT5-Abs. In terms of sentence selectors, we compare against the following.

MMR-Select. (Xiao and Carenini, 2020) Reduces redundancy by selecting s_i (candidate sentence at selection step i) such that cosine similarity w.r.t. the partially extracted summary $\hat{S}$ is minimized. Informativeness and redundancy are balanced in the same way as in Eq. 5.3.

N-gram passing (NPass). Encourages repetition by allowing p percent of n-grams in s_i to overlap with $\hat{S}$. When $p = 0$, this method reduces to n-gram blocking, whereas when $p = 1.0$, to greedy selection. We report bi-gram passing with $p = 0.8$.

Semantic Similarity Distribution (KL-Dist). Models the intuition that noun phrases in s_i will be more semantically similar to some units in $\hat{S}$ whilst dissimilar to others (Taboada, 2004). Let $\hat{Q}_i$ be the similarity distribution obtained when comparing every NP in s_i against every NP in $\hat{S}$. Similarly, let Q be the distribution of similarity between NPs in different sentences in gold summaries. Then, $f_C = \exp(-D_{KL}(Q||\hat{Q}_i)) - 1$, where D_{KL} is the Kullback–Leibler divergence. Higher values of f_C indicate lower diverge, encouraging $\hat{S}$ to have a cosine similarity distribution similar to those seen in gold summaries. All distributions were discretized into 20 bins covering values from -1.0 to 1.0.

Shuffle Classifier (CCL-Select). Holistically quantifies local coherence using CCL (Steen and Markert, 2022), a scorer trained to distinguish shuffled from unshuffled text. We use RoBERTa (Liu et al., 2019) as underlying model and use a window of 3 consecutive sentences. A dedicated model is trained for each of the datasets analyzed.

5.4.4 Automatic Evaluation

We employ the following metrics to quantify summary quality in terms of informativeness, redundancy, cohesion, and local coherence.

Informativeness. Summaries are evaluated using ROUGE F_1 score (Lin, 2004), and semantic relevancy is assessed by means of BertScore F_1 (Zhang et al., 2019) with importance weighting (IDF). For underlying BERTScore models, we use RoBERTa (Liu et al., 2019) and DeBERTa v2 XLarge-MNLI (He et al., 2021), with HuggingFace checkpoint names `roberta-large` and `deberta-xxlarge-mnli`, respectively. Previous work found that DeBERTa obtained higher correlation with human scores than RoBERTa. [3]

Redundancy. We report content redundancy scores according to sentence-wise ROUGE (RdRL) and inverse Uniqueness (IUniq). Each metric considered computes a value in the range of $[0, 1]$, the higher it is the more redundant a text will be.

Cohesion. Cohesion is evaluated with the followed metrics: *CoRL*, the average ROUGE-L F_1 between consecutive sentences; and *Entity Graph (EGr)* (Guinaudeau and Strube, 2013), which models a text as a sentence graph with edges between sentences with nouns in common, using the average edge weight as a proxy for cohesion.

Local Coherence. Finally, we report local coherence scores using our CCL scorers (Steen and Markert, 2022).

5.4.5 Human Evaluation

We elicit human judgments to assess overall quality, informativeness, and cohesion in two separate studies following the methodology in Chapter 4. We sampled 30 documents from the test set of PubMed and compare systems LT5-Casc, MMR-Select, and KvD-Select. We now elaborate on the details of each evaluation campaign.

Ranking Campaign. Following a ranking setup (Wu and Hu, 2018; Luo et al., 2019), subjects were shown the abstract and the introduction of a scientific article along with two system summaries, and then then asked to select the best summary (or select both in case of tie) according to three criteria: (i) overall quality, (ii) informativeness, and (iii) cohesion. In this setup, cohesion is evaluated as a holistic property of the text, as perceived by a reader. For more details on catch controls, instructions, and examples, please see Appendix B.3.

Chaining Campaign. We employ the same evaluation setup outlined in § 4.4.4, in which subjects were shown a single summary and were asked to annotate chains of summary content units (SCUs). As chain metrics, we report chain spread, chain density, and sentence coverage.

Agreement between human annotators is obtained by calculating the average lexical overlap between chains, expressed in F_1 score. We report the average overlap score over all pair of annotators, averaged over all system summaries. For this campaign, we include reference

[3] https://github.com/microsoft/DeBERTa

summaries as one more analysis system in order to obtain a point of reference in terms of cohesive measurements across domains.

5.5 Results and Discussion

Next, we discuss the results of our analyses, both quantitative and qualitative, and the outcome of the human evaluation campaigns.

5.5.1 Reducing Redundancy in Input Blocks

We start with the first control mechanism in the pipeline, the block selection module, and analyze its effectiveness in reducing content redundancy in the input. The following block selection strategies were compared: (i) *Original*, consisting of selecting blocks in their original order in the source document;[4] (ii) *Oracle Selection*, which selects the block that maximizes ROUGE F_1 scores (mean of ROUGE-1 and ROUGE-2) w.r.t. the reference summary; (iii) *Max. Redundancy*, which selects the most similar block possible (by flipping the sign in Eq. 5.1); and finally, (iv) BlockSelect, the proposed strategy.

The analysis, showcased in Figure 5.2, evaluates input redundancy at each block selection step, as well as informativeness and redundancy of summaries extracted from the blocks available at each step, using a greedy selector. The results indicate that the strategy used to select input blocks has a direct impact not only on input redundancy –as intended– but also on summary redundancy. This insight complements previous findings (Xiao and Carenini, 2020) that greedy selectors are highly sensitive to input redundancy.

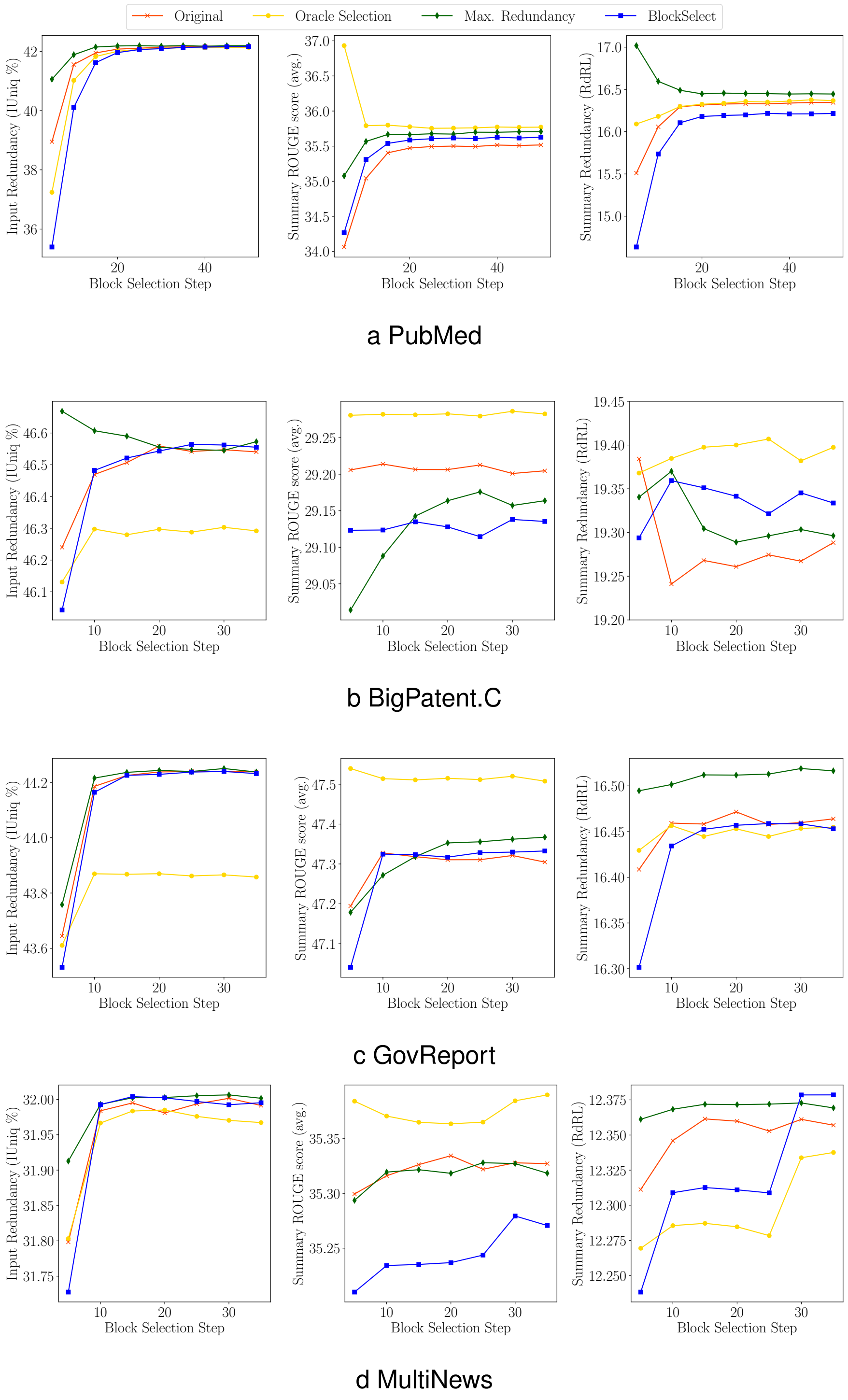
Notably, BlockSelect is effective at incrementally building an input sequence with lower content redundancy. Compared to the other strategies, ours has a clear impact on summary redundancy, enabling the pipeline to consistently extract summaries that are significantly less redundant.

5.5.2 Trading off Informativeness and Cohesion

Next, we turn to the summary extraction module. Tables 5.2 and 5.3 present the performance of all compared system in terms of informativeness, whereas Tables 5.4 and 5.5, for redundancy and cohesion, respectively.

In all our experiments, statistical significance at the 95% confidence level is estimated using Mann–Whitney U tests ($p < 0.05$). For all datasets, we found no pairwise statistical

[4] For multi-document datasets, we use the order provided in the dataset release.



a PubMed

b BigPatent.C

c GovReport

d MultiNews

Figure 5.2: Effect of block selection strategy over input redundancy (left), summary informativeness (mid), and summary redundancy (right), as block selection proceeds for PubMed, BigPatent.C, GovReport, and MultiNews.

difference between R1 scores of systems LT5-Casc, +MMR-Select, +NPass, and +CCL-Select. Analogously, Tables 5.2, 5.3, 5.4, 5.5 indicate system groups in which no pairwise difference was found for each metric reported.

First, note the impact on cohesion when controlling for redundancy. MMR-Select indeed manages to obtain comparable informativeness levels to LT5-Casc, being most effective for BigPatent.C. However, minimizing sentence similarity comes at the expense of a significant decrease in cohesion (CoRL) and local coherence (CCL). Second, we find that NPass is the only one capable of obtaining comparable or better ROUGE scores but CoRL and EGr scores indicate that lexical passing is not enough to improve cohesion. Next, note that KL-Dist employs a seemingly more aggressive trade-off between ROUGE and CoRL in all datasets except PubMed. We hypothesize that its cohesion term, f_C, saturates the final candidate score during trade-off, which prompts the selector to pick candidates with lower informative scores.

When guiding selection with a holistic shuffle scorer, as expected, CCL-Select obtains remarkably high CCL scores, closing the gap w.r.t. Ext-Oracle in most datasets and even surpassing it for BigPatent.C. However, note that this selector does show a significant reduction in CoRL and EGr scores w.r.t. LT5-Casc, indicating that CCL is measuring also discourse organization, possibly in the form of rhetorical role ordering –first background, then method, and so on. Hence, it can be said that summaries in CCL-Select are better organized in terms of rhetorical roles but exhibit lower cohesion than greedily selected summaries.

Finally, KvD-Select manages to strike an even more aggressive trade-off between informativeness and cohesion. Across datasets, the selector exhibits lower ROUGE scores but the best CoRL, EGr scores (except for PubMed), and second highest CCL score after CCL-Select.

Complementary Cohesion Measures. At this point in the analysis, it is important to note the limitations of the reported measures of cohesion so far. CoRL is capable of capturing cohesive ties in consecutive sentences, potentially missing ties between sentences farther apart; whereas EGr relies on lexical repetition of nouns to connect sentences, potentially missing ties between lexically different but semantically similar nouns. For this reason, we report complementary results for Lexical Graph (LGr) (Mesgar and Strube, 2016) and the noun variants of DiscoScore (DS) (Zhao et al., 2023), DS-Focus[NN] and DS-Sent[NN].

Table 5.6 presents results for the additional metrics in all datasets and systems. Similarly to the previous section, Table 5.6 indicates system groups, one marker per group, in which no pairwise difference was found for each metric reported.

In terms of LGr scores, we find that KvD-Select outperforms all systems in all datasets except PubMed, indicating that consecutive sentences in summaries extracted by our sys-

System	PubMed			BigPatent.C			GovReport			MultiNews		
	R1	**R2**	**RL**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
Ext-Oracle	65.10	37.99	60.76	53.85	23.20	46.90	72.66	40.90	69.36	62.66	33.73	57.93
LT5-Abs	46.27	20.92	42.40	37.63	15.67	32.84	51.72	24.79	49.03	45.72	17.70	41.86
LT5-Flat	**48.15**	**21.45**	**44.49**	39.54	13.25†	**34.30**†	59.33	25.94	56.29	**47.07**	**17.54**	**42.96**
LT5-Casc	46.16†	19.74†	42.49†	39.57†	13.25†	34.26†	59.73†	26.21†	56.50†	46.80†	17.21†	42.66†
+MMR-Select	46.14†	19.63†	42.47†	**39.59**†	**13.29**†	**34.30**†	**59.79**†	**26.30**†	**56.56**†	46.76†	17.13†	42.59†
+NPass	46.38†	19.92†	42.74†	**39.59**†	13.26†	34.29†	**59.79**†	26.25†	**56.56**†	46.91†	17.27†	42.78†
+KL-Dist	46.00†	19.62†	42.32†	39.25†	13.07†	33.89†	59.46†	25.85†	56.15†	46.63†	16.97†	42.45
+CCL-Select	45.91†	19.60†	42.45†	39.16†	12.95	33.92†	59.72†	26.24†	56.50†	46.85†	17.29†	42.71†
+KvD-Select	44.90	18.47	41.27	38.37	12.41	33.13	57.88	23.66	54.57	45.85	16.13	41.62

Table 5.2: Informativeness in terms of ROUGE scores (R1, R2, RL). †: no stat. difference between systems in the same column. Best systems are **bolded**; systems better than LT5-Casc shown in blue and worse, in red.

tem contain more cohesive ties connecting highly semantically similar nouns. Differences between systems in terms of DS-Foc and DS-Sen scores are much less clearer, with DS-Foc seemingly being more indicative of content coverage and DS-Sen more indicative of cohesion. Nevertheless, KvD-Select is consistently competitive across datasets, even obtaining the highest DS-Sen score for MultiNews.

Effect of Parameter λ_{sel}. Next, we analyze how summary properties vary across increasing levels of λ_{sel}, showcased in Figure 5.3 for all datasets. Note that when $\lambda_{\text{sel}} = 0$ selectors depend entirely on f_C, and $\lambda_{\text{sel}} = 1.0$ is equivalent to a greedy selector. As expected, informativeness is higher as f_I is weighted up (higher λ_{sel}) with all selectors except MMR-Select. This indicates that it is possible to increase cohesion without incurring a significant loss in informativeness. Interestingly, KvD-Select seems robust to λ_{sel} in terms of CoRL and RdRL. We hypothesize that KvD-Select benefits from a signal indicating which cohesive ties are informative and worth enforcing.

Impact of Cascaded Processing. Next, we investigated the impact of processing input blocks in a cascaded fashion vs concatenating them into a flat sequence and processing all of them at once. The results on informativeness, redundancy, and cohesion are presented in Tables 5.7, 5.8, and 5.9, respectively. Similarly to previous analyses, we indicate system groups, one marker per group, in which no pairwise difference was found for each metric reported.

We found that cascaded processing exhibits lower ROUGE scores than flat processing in PubMed and MultiNews, and comparable performance for BigPatent.C and GovReport. However, LT5-Casc shows slightly higher CoRL scores in all datasets. This indicates

System	PubMed		BigPatent.C		GovReport		MultiNews	
	RoB	DeB	RoB	DeB	RoB	DeB	RoB	DeB
Ext-Oracle	88.44	80.20	85.83	73.80	88.30	80.06	88.69	80.04
LT5-Abs	**85.71**	73.94	**84.09**	**70.16**	**86.49**	**76.52**	85.13	74.22
LT5-Flat	**85.71**	**74.16**	83.77	69.81	86.44	75.95	**86.03**	**74.34**
LT5-Casc	85.05	73.08	83.65†	69.75†	86.46†	76.06†	85.97	74.10
+MMR-Select	85.05	73.07†	83.66†	69.78†	**86.49**	76.08	85.93†	74.04
+NPass	85.13	73.21	83.67†	69.76†	86.47†	76.08	86.01	74.17†
+KL-Dist	85.02†	73.02†	83.52	69.57	86.30	75.84	85.94†	74.07
+CCL-Select	84.99†	72.98	83.63	69.62	86.47†	76.05†	85.91†	74.18†
+KvD-Select	84.76	72.43	83.34	69.15	85.99	75.17	85.72	73.67

Table 5.3: Semantic relevance of system summaries in terms of BERTScore F_1 using RoBERTa (RoB) and DeBERTa (DeB) as base models. Best systems are **bolded**. See Table 5.2 for formatting details.

System	PubMed		BigPatent.C		GovReport		MultiNews	
	RdRL	IUniq	RdRL	IUniq	RdRL	IUniq	RdRL	IUniq
Ext-Oracle	13.91	20.36	14.70	19.51	14.20	29.14	10.08	16.98
LT5-Abs	16.15	21.24	38.04	39.40	15.89	26.15	12.60	20.00
LT5-Flat	16.49	23.43	19.76	21.32	**15.78**	32.46	12.24	20.63
LT5-Casc	17.08†	22.94†	20.15†	21.46†	16.34	31.68	12.26	20.59
+MMR-Select	16.99†	22.85†	19.17†	21.09†	16.16	31.53	12.05	20.50
+NPass	16.39	21.66	19.79†	21.18†	16.24	31.42	12.03	19.92†
+KL-Dist	16.83	22.08	20.30†	21.44†	16.49†	31.35	12.57	20.22
+CCL-Select	16.63	22.42	**18.97**	**20.87**	16.31	31.65	**11.93**	20.29
+KvD-Select	**16.24**	**21.53**	21.09	21.65†	16.69†	**30.97**	12.97	**19.97**†
Gold	13.54	19.12	18.11	20.85	13.37	28.78	9.72	16.35

Table 5.4: Summary redundancy in terms of sentence-wise ROUGE (RdRL) and inverse uniqueness (IUniq). †: no stat. difference between systems in the same column. Best systems are **bolded**. For all metrics, lower is better. See Table 5.2 for formatting details.

that cascaded processing puts a greedy selector in a better position to extract more cohesive summaries at the expense of a slight decrease in informativeness.

Finally, we assessed the impact of architectural choice for the Local Encoder module

Systems	PubMed			BigPatent.C			GovReport			MultiNews		
	CoRL	EGr	CCL	CoRL	EGr	CCL	CoRL	EGr	CCL	CoRL	EGr	CCL
Ext-Oracle	14.68	0.99	0.43	15.94	0.68	0.34	16.54	1.92	0.58	10.85	0.68	0.51
LT5-Abs	19.17	0.96	0.86	42.58	0.82	0.81	18.54	1.60	0.81	14.25	0.78	0.78
LT5-Flat	16.59	**1.10**	0.24	19.76	0.75	0.32	16.06	2.01	0.31	12.25	0.91	0.25
LT5-Casc	**17.47**†	1.07†	0.25†	20.27†	0.73	0.37†	16.45†	2.04†	0.30†	12.51	0.90	0.24
+MMR-Select	16.89	1.07	0.23	18.83	0.73	0.33†	15.88	2.03†	0.30	11.83	0.88†	0.22
+NPass	16.66	1.07	0.25	19.92†	0.73	0.37†	16.38†	2.04	0.31†	12.18	0.89†	0.24
+KL-Dist	17.31†	1.08†	0.26†	20.54†	0.73	0.39†	16.87	2.05	0.31†	12.82	0.95	0.24
+CCL-Select	17.28†	1.06†	**0.66**	19.42	0.71	**0.88**	16.73†	2.04†	**0.65**	11.94	0.86	**0.63**
+KvD-Select	17.33†	1.05†	0.27	**22.20**	**0.78**	0.40†	**18.88**	**2.15**	0.32	**14.22**	**0.99**	0.28
Gold	14.45	0.96	0.91	19.20	0.78	0.92	16.20	1.95	0.87	10.45	0.71	0.90

Table 5.5: Summary cohesion in terms of consecutive ROUGE-L score (CoRL) and EntityGraph (EGr), as well as coherence (CCL). For all metrics, higher is better. See Table 5.2 for formatting details.

in our pipeline by comparing MemSum (Gu et al., 2022), and LLaMA with 7B parameters (Touvron et al., 2023a). Using LLaMA as local encoder allows our system to select –greedily– sentences that have little lexical overlap between them, prompting low summary redundancy scores and in turn lowering cohesion scores. Moreover, the coverage is severely impacted as seen by the low ROUGE scores. These results might indicate that finetuning a large pretrained model like LLaMA does not necessarily translate to better informativeness, performing much lower than a smaller model pretrained on the summarization task. Perhaps unsurprisingly, task-specific, smaller models can be competitive to massive foundation models trained on 1000x more data.

Using MemSum as the local encoder had a similar outcome, although not as severe as when using LLaMa. Summaries in Gu et al. (2022) were obtained in a scenario where only up to 500 sentences were consumed in the order they appear in the document. In contrast, in our setup, the compared systems consume up to 16 384 pieces of input text in the order the block selector module retrieves. The performance gap between the results in Gu et al. (2022) and the ones we report can then be explained by input length and ordering conditions.

Systems	PubMed			BigPatent.C			GovReport			MultiNews		
	LGr	DS-Foc	DS-Sen	LGr	DS-Foc	DS-Sen	LGr	DS-Foc	DS-Sen	LGr	DS-Foc	DS-Sen
Ext-Oracle	1.1463	0.5185	0.9692	0.8022	0.5642	0.9499	1.9735	1.7069	0.9973	0.7488	0.3684	0.9789
LT5-Abs	1.0953	1.2290	0.9262	0.8411	5.0947	0.9106	1.6333	7.9962	0.9896	0.8492	0.7973	0.9602
LT5-Flat	**1.2467**	1.0925	**0.9350**	0.8084	1.0244†	**0.9256**†	2.0583	3.8380	0.9949	0.9687†	0.7623	0.9642
LT5-Casc	1.2204†	1.0219†	0.9303†	0.7858†	1.0291†	0.9228†	2.0838†	3.4428†	0.9947†	0.9628†	0.7353†	0.9629†
+MMR-Select	1.2203†	1.0117†	0.9300†	0.7803	1.0108	0.9239†	2.0797	3.4391	0.9948†	0.9464	0.7348†	0.9632†
+NPass	1.2153‡	0.9833	0.9306†	0.7837†	1.0246†	0.9229†	2.0823†	3.4129	0.9946	0.9595	0.7183‡	0.9630†
+KL-Dist	1.2196‡	1.0048†	0.9296‡	0.7773	1.0218†	0.9204‡	2.0949	3.3835	0.9945‡	1.0219	0.7361†	0.9617
+CCL-Select	1.2075§	0.9970†	0.9305†	0.7623	0.9869	0.9233†	2.0867†	3.4481†	0.9948†	0.9323	0.7150‡	0.9638
+KvD-Select	1.1933§	0.9948†	0.9298‡	**0.8207**	1.0644	0.9210‡	**2.1824**	3.5580	0.9945‡	**1.0536**	0.7235	**0.9651**
Gold	1.1088	0	1.0000	0.8368	-	1.0000	1.9990	-	1.0000	0.7756	-	1.0000

Table 5.6: Summary cohesion in terms of Lexical Graph (LGr; Mesgar and Strube (2016)) and DiscoScore's (Zhao et al., 2023) DS-Focus[NN] (DS-Foc) and DS-Sent[NN] (DS-Sen). For all metrics, higher value is better except for DS-Foc. (†,‡,§): no stat. difference between systems in the same column. Best extractive systems are **bolded**; systems better than LT5-Casc shown in blue and worse, in red.

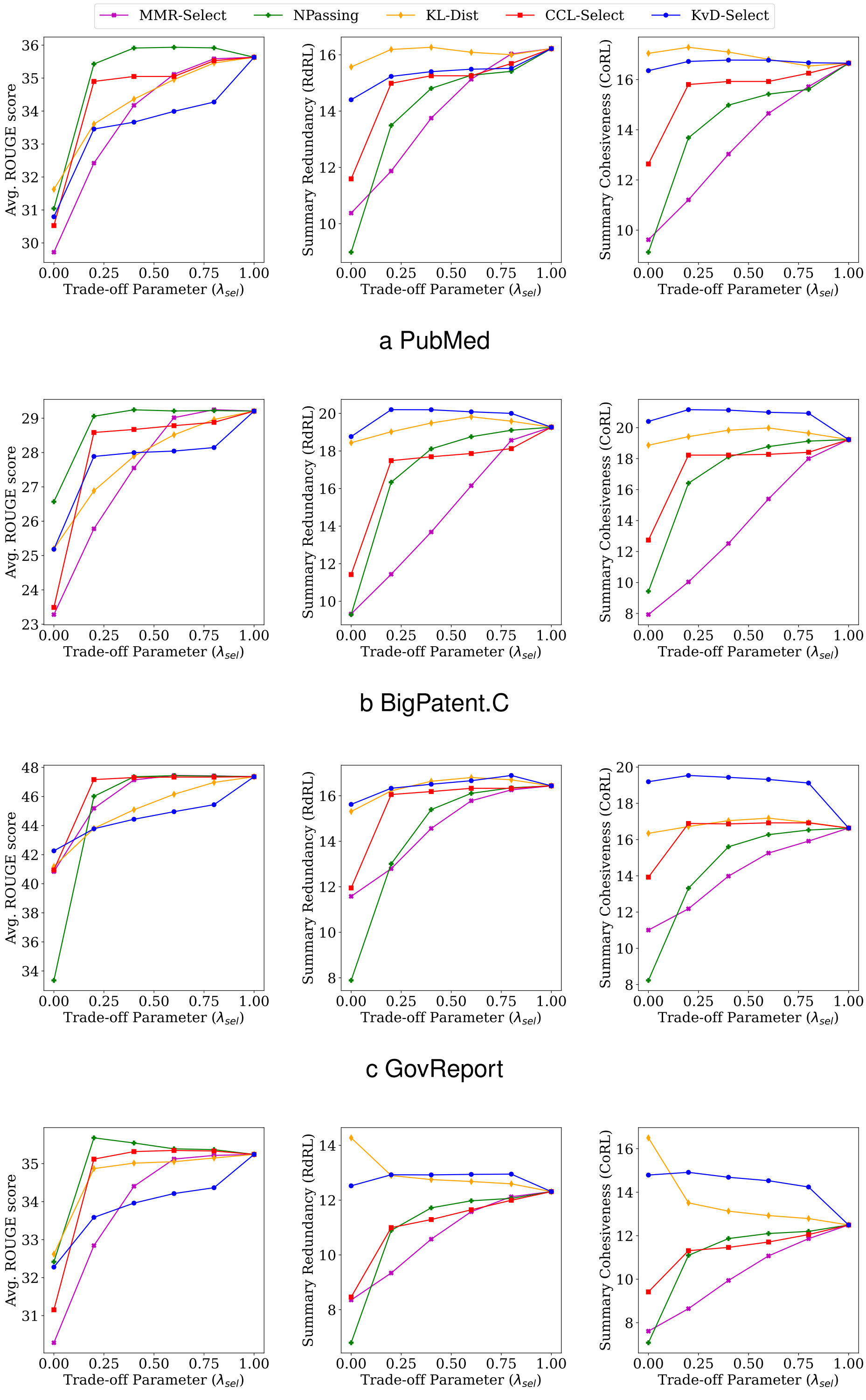
System	PubMed			BigPatent.C			GovReport			MultiNews		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
LED-Abs	45.31	20.73	41.82	35.89	14.66	31.45	54.34	24.78	51.48	46.73	18.93	43.11
LT5-Abs	46.27	20.92	42.40	37.63	15.67	32.84	51.72	24.79	49.03	45.72	17.70	41.86
LT5-Flat	**48.15**	**21.45**	**44.49**	**39.54**†	**13.25**†	**34.30**†	**59.33**†	**25.94**†	**56.29**†	**47.07**†	**17.54**	**42.96**
LED-Flat	40.20	13.87	36.85	36.65	11.07	31.94	57.59	23.40	54.56	45.28	15.78	41.35
MemSum-Casc	40.29	14.85	37.09	36.07	10.79	30.97	54.91	19.66	51.75	44.47	15.28	40.32
LLaMA-Casc	37.60	11.86	34.51	36.82	11.24	32.00	54.20	19.02	50.90	45.02	15.48	41.00
LT5-Casc	46.16	19.74	42.49	**39.57**†	**13.25**†	**34.26**†	**59.73**†	**26.21**†	**56.50**†	**46.80**†	17.21	42.66

Table 5.7: Informativeness in terms of ROUGE F_1 scores (R1, R2, RL), for Flat and Cascaded block-processing systems. Best extractive systems are **bolded**. †: no stat. difference between systems in the same column.

5.5.3 Human Evaluation

In both studies, statistical significance between system scores was assessed using a one-way ANOVA with posthoc Tukey tests with 95% confidence interval ($p < 0.01$). Results are presented in Table 5.10.

Ranking. Krippendorff's α (Krippendorff, 2011) showed an inter-annotator agreement of 0.68. For overall quality, subjects showed a significant preference for KvD-Select over LT5-Casc. For cohesion, KvD-Select was perceived as more cohesive compared to LT5-



a PubMed

b BigPatent.C

c GovReport

d MultiNews

Figure 5.3: Informativeness (left), redundancy (mid), and cohesion (right) across different values of the trade-off parameter λ_{sel} for all datasets.

System	PubMed		BigPatent.C		GovReport		MultiNews	
	RdRL	IUniq	RdRL	IUniq	RdRL	IUniq	RdRL	IUniq
LED-Abs	15.00	20.83	38.23	44.82	16.55	31.33	8.37	23.19
LT5-Abs	16.15	21.24	38.04	39.40	15.89	26.15	12.60	20.00
LED-Flat	14.70	21.86	17.62	20.07	14.94	31.01	11.25	**19.06**
LT5-Flat	16.49	23.43	19.76†	21.32†	15.78	32.46	12.24†	20.63
MemSum-Casc	12.58	**19.39**	19.41	21.23	13.77	27.47	12.29†	19.28
LLaMA-Casc	**11.61**	19.40	**17.51**	**18.96**	**12.43**	**26.64**	**10.87**	19.46
LT5-Casc	17.08	22.94	20.15†	21.46†	16.34	31.68	12.26†	20.59

Table 5.8: Summary redundancy in terms of sentence-wise ROUGE (RdRL) and inverse uniqueness (IUniq), for Flat and Cascaded block-processing systems. For all metrics, lower is better. Best systems are **bolded**. †: no stat. difference between systems in the same column.

Systems	PubMed			BigPatent.C			GovReport			MultiNews		
	CoRL	EGr	CCL	CoRL	EGr	CCL	CoRL	EGr	CCL	CoRL	EGr	CCL
LED-Abs	17.86	0.93	0.85	42.24	0.94	0.77	19.81	1.75	0.75	8.47	0.53	0.83
LT5-Abs	19.17	0.96	0.86	42.58	0.82	0.81	18.54	1.60	0.81	14.25	0.78	0.78
LED-Flat	15.18	1.00	**0.36**	17.67	0.69	0.35†	15.06	1.91	0.30	11.31	0.81	**0.30**
LT5-Flat	16.60	**1.10**	0.26	19.76†	**0.75**†	0.37†	16.06	2.00	0.28†	12.25†	**0.91**	0.26†
MemSum-Casc	12.87	0.75	0.25	**20.56**	0.62	0.34	13.93	1.72	**0.29**	**13.16**	0.84	0.26†
LLaMA-Casc	12.18	0.70	0.27	17.54	0.70	**0.39**	12.53	1.58	0.28†	11.15	0.77	0.25
LT5-Casc	**17.47**	1.07	0.26	20.26†	0.73†	**0.39**	**16.46**	**2.04**	0.27†	12.51†	0.90	0.26†

Table 5.9: Cohesion of extracted summaries in terms of consecutive ROUGE-L score (CoRL) and EntityGraph (E.Gr.), as well as coherence (CCL), for Flat and Cascaded block-processing systems. For all metrics, higher is better. Best systems are **bolded**. †: no stat. difference between systems in the same column.

Casc, and LT5-Casc was more cohesive than MMR-Select.

Chaining. Chain overlap was calculated at 0.90. Differences between LT5-Casc and all other systems, as well as MMR-Select–Gold and KvD-Select–LT5-Casc were found to be significant, for all measurements of cohesion. Moreover, the number of NPs annotated per chain was 2.30, 2.33, 2.80, and 2.55, for systems LT5-Casc, MMR-Select, KvD-Select, and Gold, respectively.

We found that KvD-Select summaries exhibit more active and denser chains and better-

System	**Ranking**			**Chaining**		
	Ov↓	**I↓**	**C↓**	**Spr↓**	**Den↑**	**Cov↑**
LT5-Casc	1.59	1.56	1.59	**1.93**	1.29	57.12
+MMR-Select	1.50	1.48	1.47	2.36	1.28	53.21
+KvD-Select	**1.41**	**1.46**	**1.44**	2.05	**1.40**	**68.78**
Gold	-	-	-	1.91	1.36	69.65

Table 5.10: Ranking (left) w.r.t. (Ov)erall quality, (I)nformativeness, and (C)ohesion; and properties of annotated chains (right): spread (Spr), density (Den), and sentence coverage (Cov,%). Best systems are **bolded**. (↑,↓): higher, lower is better.

covered sentences than the baselines. Note that LT5-Casc obtains the lowest chain spread but also low coverage, indicating that its summaries exhibit very few chains that happen to be close to each other. In contrast, MMR-Select obtains the highest chain spread and low number of chains, indicating content with low diversity and sparsely presented.

5.5.4 Qualitative Analysis

Finally, the output of systems Gold, MMR-Select, and KvD-Select is qualitatively analyzed in more detail. Table 5.11 showcases a reference summary along with candidate summaries extracted from a MultiNews sample. For each system summary, we report its informativeness, redundancy, and cohesion level, quantified by automatic metrics. Moreover, each summary is manually annotated with lexical chains, and each sentence is presented with the IDs of the chains covering them. We now proceed to analyze each system summary in turn.

Starting with the gold summary, it exhibits six chains (1,2,3,4,5,11) that always cover adjacent or near-adjacent sentences. Whilst most chains span windows of two to four sentences, we do observe one dominant chain, {company}, covering almost the entire summary. However, chains 2 ({phone business}) and 3 ({reporter}) also show high predominance.

Next, the MMR-Select summary exhibits seven chains (1,2,4,6,7,8,10), however showing extensive coverage gaps. Chains 1 and 2 are again dominant, evidence that the system is able to capture relevant content. Nevertheless, irrelevant content is still selected (chain 6, {crawl}), possibly due to the redundancy-reduction term in Eq.X encouraging semantically dissimilar sentences to be selected.

In contrast, the KvD-Select summary exhibits five chains (1,2,8,9,10), all of them covering adjacent or near-adjacent sentences and spanning windows of two to three sentences. Interestingly, although chains 1 and 2 are the dominant chains, chain coverage is rather uniform across the summary. As a consequence, topics flow smoothly across the summary, with no abrupt change in chain presence as seen in MMR-Select.

Finally, we observe the following from the automatic metrics. The presence of highly informative chains (1 and 2) in KvD-Select and MMR-Select summaries give them a similar average ROUGE score. Nevertheless, MMR-Select's relevancy is slightly higher, possibly due to the presence of chain 5 ({Stephen Elop}), also present in Gold. When looking at the cohesion score, Gold shows a relatively low CoRL score. This is to be expected given that CoRL is based on n-gram overlap and it is not equipped to capture cohesive ties other than lexical reiteration. However, CoRL is able to capture the ties presented by KvD-Select which scores higher than MMR-Select, demonstrating the efficacy of modeling cohesion of the former.

5.6 Summary

In this chapter, we presented an extractive summarization algorithm that controls each summary quality independently, in scenarios where the input is highly redundant. Redundancy is controlled as the input is consumed, and informativeness and cohesion are balanced during sentence selection.

Results show that our input processing strategy is effective at retrieving non-redundant yet relevant passages, reducing the redundancy levels the rest of the pipeline is exposed to. We found that even a greedy sentence selector benefits greatly from such control, obtaining encouraging improvements in summary informativeness and summary redundancy when compared against strong block retrieval strategies. In addition, our sentence selector emulates human memory to keep track of cohesive chains while building the summary, enforcing ties between noun phrases directly. Extensive automatic and human experiments revealed that it is possible to extract highly cohesive summaries that are as informative as summaries optimizing only for informativeness. Interestingly, the scorers for informativeness and cohesion –independently modeled– seem to complement each other. We hypothesize that our selector benefits from a signal indicating which cohesive ties are more informative and worth enforcing.

System Summary	Chain IDs
Gold (Avg. ROUGE=-; RdRL=8.3; CoRL=6.63)	
Why did Microsoft buy Nokia's phone business?	1,2
We now know Microsoft's answer: the computing giant released a 30-slide presentation today arguing that the move will improve Microsoft's margins on Windows phones, which will allow it to invest more in the platform, which will accelerate sales and market share growth, The Washington Post reports.	1,2,3,5
But John Herrman at BuzzFeed has another explanation: "fear of dying alone."	3
Here's what he and other pundits are saying: the presentation "manages to sound both insane and uninspiring, outlining modest goals that still sound unrealistic," Herrman argues - like capturing a whole 15% of the smartphone market.	2,3,5
"It's a fitting end for the close of Microsoft's Ballmer era, during which the company...missed out on the most important change in consumer electronics in decades" while remaining profitable in unglamorous ways.	1,2,4
Like everyone, Microsoft is trying to ape the Apple model, MobileOpportunity observes.	1,3
But it's not so sure that's a good idea.	3
"There already is an Apple," the blog points out, and other software/hardware hybrid companies, like Palm and BlackBerry, have been crushed under its heel.	1,3
Maybe Microsoft should have tried to patch up its tried-and-true strategy of licensing its OS.	1,2
The move risks complicating Microsoft's crucial relationships with other PC and device manufacturers, one analyst tells ZDNet.	1,2,3
But he adds that "Microsoft needed to make a bold move" or face "certain terminal decline," and that the price it paid for Nokia "seems extremely reasonable."	1,3
Meanwhile, Matthew Yglesias at Slate digs up a fairly interesting memo from Nokia CEO (and, perhaps, Microsoft heir apparent) Stephen Elop, in which he uses the story of a Deepwater Horizon worker leaping from the burning oil platform - a seemingly desperate, yet necessary move - to explain the company's shift from its own failed OS to Windows Phone.	1,2,3,4,11
Of course, Yglesias notes, that move "was basically a total failure."	3,11

Table 5.11: Reference summary, along with summaries extracted by MMR-Select and KvD-Select for a MultiNews sample with informativeness (average ROUGE score), redundancy (RdRL), and cohesion (CoRL) scores. Each sentence is annotated with lexical chains, color-coded in the text and IDs shown to the right. Text was detokenized and truecased for ease of reading.(*continues*)

System Summary	Chain IDs
MMR-Select (Avg. ROUGE=28.25; RdRL=8.31; CoRL=11.98)	
Summary: Microsoft's acquisition of Nokia is aimed at building a devices and services strategy, but the joint company won't take the same form as Apple.	1,2,10
This crawl was run at level 1 (URLs, including their embeds, plus the URLs of all outbound links, including their embeds).	6
Today's sale price, which includes 1.65 billion euros in patents, is just 5.44 billion euros.	2,7
It's been a rough decade.	-
Microsoft is buying Nokia's cell phone business and licensing its patent portfolio, according to both companies.	1,2
In 2003, Nokia's cell phone market share exceeded 35%.	1,2
That same year, its phone business alone posted an operating profit of 5.48 billion euros.	2,7
Nokia lashed itself to Microsoft's mast after losing out to iOS and Android in the smartphone market share stakes and with the limited success of the Lumia range so far, enough to keep interest in Windows Phone alive, most analysts are seeing a certain amount of inevitability to the acquisition, even if they are split on what its biggest implications are.	1,2,8,10
The seed for this crawl was a list of every host in the Wayback Machine.	6
The WARC files associated with this crawl are not currently available to the general public.	6
Five years ago was the year the App Store first opened.	2
Windows Phone has barely dented the now much larger smartphone market.	2
Many at the time wondered if Stephen Elop's time at Nokia would be spent grooming the company for purchase —a foreigner in all possible ways, he began his time at the company with a memo rightly but offensively declaring Nokia's proud platform a failure and quickly pledged the company's commitment to the still-tiny Windows Phone.	1,2,4

Table 5.11: (*continued*)

System Summary	Chain IDs
KvD-Select (Avg. ROUGE=26.33; RdRL=12.48; CoRL=14.41)	
Summary: Microsoft's acquisition of Nokia is aimed at building a devices and services strategy, but the joint company won't take the same form as Apple.	1,2,10
Microsoft has been working on its evolution into a devices and services company, moving away from the services business it has traditionally been, for several years now with limited success.	1,2,8
Nokia lashed itself to Microsoft's mast after losing out to iOS and Android in the smartphone market share stakes and with the limited success of the Lumia range so far, enough to keep interest in Windows Phone alive, most analysts are seeing a certain amount of inevitability to the acquisition, even if they are split on what its biggest implications are.	1,2,8,10
Owning the desktop (via Windows) and building additional services on top, like Office or Search, has been vital for Microsoft's strategy until now, so, as our interest shifts from the desktop to the tablet or smartphone, it's essential to Microsoft's broader business (even Azure) that it can retain that connection in some form.	1,2,9
But he said Microsoft's challenge remains how to unite the myriad services and brands - Windows, Nokia, Live, Surface, Xbox, Bing, and more - into a cohesive experience that will command and cement customer loyalty.	1,2,9
It felt like a radical about-face, but no matter: Nokia and Microsoft were going to save each other.	1,9

Table 5.11: (*continued*)

Chapter 6

Conclusions

The escalating demand for performant automatic summarization systems has spurred significant advancements in the field in order to cater to the diverse information needs of end-users. Specifically, when considering human end-users, summarizers are required not only to deliver informative content but also to ensure that the resulting text is easily comprehensible, i.e. to be coherent and cohesive. This requirement poses substantial challenges for extractive summarizers, where the detection and selection of content units are critical, and their presentation demands coherence and cohesion.

The Micro-Macro theory of text comprehension and production serves as a comprehensive framework, providing detailed operationalizations of cognitive processes that model interactions among content units, both in close proximity and over extended spans. Throughout this thesis, we presented evidence of the potency of these simulated processes as effective mechanisms for controlling summary properties.

In the rest of this chapter we elaborate on the main conclusions drawn from our efforts in this area, discuss the limitations of our approaches, and delineate potential avenues for future work.

6.1 Conclusions

In this thesis, we investigated whether summary properties could be controlled in a principled way to better fit the information needs of human end-users. To this end, we developed generic, extractive summarization systems equipped with mechanisms to control the text properties of the produced summary. The proposed mechanisms are inspired by cognitive processes that model the interactions among content units in human memory –both short-term and long-term– according to the Micro-Macro theory, KvD. As a case study, we investigated the scenario in which the input document(s) is long and contains redundant content. The resulting summarization systems allow the trade-off among informativeness, content coverage, redundancy, and cohesion in summaries through interpretable parameters. Furthermore, our models are interpretable, offering the option to researchers (or users) to track which concepts are considered relevant, cohesive among each other, or redundant.

First, we focused on the problem of content selection during unsupervised summarization of highly technical long documents, such as scientific articles. In Chapter 3, we reproduced a robust KvD reader, a system simulating KvD processes during document understanding, and equipped it with mechanisms designed to exploit the properties of memory trees during simulation. Our results demonstrated that the KvD reader configurations using these mechanisms perform comparably or better than strong unsupervised baselines on ranking highly relevant content units. Additionally, through comprehensive human evaluation and analysis, we observed that our system preferred to provide less specific yet relevant content rather than content not relevant at all. Moreover, we implemented a control mechanism capable of maintaining summary lengths close to a predefined budget, in order to ensure a fair comparison at the system level. The resulting summaries exhibit a length distribution with significantly low variance, centered around the budget, in contrast to a greedy sentence selector.

Next, we shifted our focus to mechanisms to balance redundancy and cohesion during document understanding, and the impact of these on informativeness in both unsupervised and reinforcement learning scenarios. In Chapter 4, we introduced two novel computational implementations of unsupervised KvD reading, addressing many limitations of previous implementations, including reliance on external NLP tools. Deep analyses revealed how the implemented cognitive processes trade-off informativeness for improved cohesion while still maintaining acceptable levels of repetitiveness. Notably, the proposed KvD systems excel at extracting highly cohesive summaries even at increasing levels of document redundancy, with humans perceiving the extracted summaries as more informative and more cohesive than strong unsupervised baselines. In a reinforcement learning scenario, we found that strong neural baselines are able to effectively optimize for informativeness and cohesion, obtaining improvements for both properties and showcasing their complementary nature.

Finally, we focused on controlling summary qualities at various stages of the summarization pipeline simultaneously, presenting two mechanisms for achieving this in Chapter 5. The first mechanism addresses input redundancy in a cascaded way, similar to information retrieval pipelines, indirectly reducing redundancy in the final summary. The second control mechanism consists of a summary extractor that quantifies informativeness and cohesion independently, employing a linear combination approach to balance them. The proposed extractor models informativeness using strong neural encoders, while cohesion is modeled by a novel KvD production simulator – a module simulating working memory during summary sentence selection that enforces cohesive ties between candidate sentences. When tested on both single and multi-document scenarios, the proposed control mechanisms are effective at extracting highly cohesive summaries, albeit at the expense of informativeness in terms of automatic metrics. However, the extracted summaries were still perceived as informative as baseline summaries by human evaluators, highlighting the limitation of automatic metrics in evaluating equally useful summaries with varying wordings.

In conclusion, this thesis offers a compelling option for summarization of long documents using extractive techniques that proved itself relevant in a highly changing generative landscape where abstractive techniques are dominant. The extractive systems proposed in this thesis have the advantage of producing more faithful summaries in terms of factuality and writing style w.r.t. the input document. Additionally, the decisions our systems make can be easily interpreted by inspecting the contents in the memory structures, providing invaluable insights during troubleshooting or when aiming to adapt our models to other usages or domains.

6.2 Limitations and Future Work

The research discussed in this thesis presents the following noteworthy limitations. In first instance, the proposed system for content selection, described in Chapter 3, still relies on external NLP tools, such as constituency and dependency parsers, to build semantic propositions. This reliance introduces a limitation, particularly when processing text in domains significantly distant from those on which the tools were originally trained. Throughout this thesis, we gradually reduced this dependence on external tools. Potential avenues for improvement in this area could be directed to simulate KvD reading over text spans, single wordpieces, or clusters of them. Recent work has yielded promising results on enforcing foci, modeled as wordpieces, in nearby sentences to be close in the embedding space (Jeon and Strube, 2022; Zhao et al., 2023).

Another limitation pertains to the high technicality of the analyzed domain, i.e. scientific articles and government patents and reports. In closed-book scenarios, where only the source document is used for inference without any access to external knowledge, the domain specificity hampers the capacity of automated systems to detect relevant content units or ascertain the equivalence of two concepts. An exciting avenue for future work in this context involves extending the summarization task to an open-book scenario, wherein rich knowledge sources are available at any stage of the summarization process. Recent work on lay summarization of biomedical articles (Goldsack et al., 2023) has shown promising results by leveraging external knowledge graphs to associate domain-specific concepts with their corresponding descriptions.

The results and conclusions derived throughout this thesis are also subject to the inherent limitations of the current human evaluation methodologies. Notably, concerns raised by Gillick and Liu (2010) regarding the quality differences between crowd-sourced and expert annotations have been acknowledged. Later, Fabbri et al. (2021) confirmed this issue while providing evidence that results from both levels of expertise can lead to the same meaningful conclusions. These insights were considered throughout our evaluation efforts, prompting us to adopt a middle-ground approach by recruiting crowd-sourced subjects with basic knowledge in the target domain, e.g. to have worked in the healthcare or medical sector before.

Finally, regarding the presentation format and usefulness of extractive summaries produced in this thesis, the following can be stated. First, the proposed system extracts complete sentences and concatenates them to form the final summary. We do not perform any kind of post-editing of discourse markers that might break coherence in the summary. However, as we saw in Chapters 4 and 5, our results show that the extracted summaries are still perceived as cohesive by humans. Nevertheless, post-editing is an interesting focus for future work. Second, we argue about the usefulness of an extractive system in a generative landscape where large language models are predominant. Recent large language models have shown impressive capabilities at producing coherent, assertive text, some even capable of consuming long sequences of tokens. However, hallucinations are a pervasive problem in these systems, especially in highly technical domains like the ones considered in this thesis. In this scenario, an extractive summary has the advantage of presenting information from the source verbatim and hence, with reduced –albeit still present– hallucination (Zhang et al., 2023). Moreover, extracted summaries preserve the writing style of the input as well as technical, domain-specific terms, avoiding altogether the problems of over-simplification and style drifting.

Appendix A

Optimization and Implementation Details

In this appendix, we elaborate on the training and optimization details of the models described in this thesis.

A.1 Trade-off Control during Summary Extraction

In this section, we provide pipeline details and complementary results for models in Chapter 5. The local encoders in the cascaded retrieval module were trained in the following manner.

Models based in LongT5 were finetuned from pretrained Huggingface's checkpoint `google/long-t5-tglobal-base` using one NVIDA A100 (80GB of GPU memory) Similarly, LLama-based baselines were finetuned using 4 A100s from the official weights.[1] The global context encoder is trained from scratch. Table A.1 provides a comprehensive account of hyperparameter values used for training and inference in our experiments, for all datasets.

Regarding the abstractive baselines, training of LongT5-Abs was done for 10k steps with batch size of 128, AdamW optimizers, and constant learning rate of 1E-3, using the Huggingface's checkpoints `google/long-t5-tglobal-base`. Inference was done using a beam size of 5 and length penalty of 0.5, 0.5, 2.0, and 1.0 for PubMed, BigPatent.C, GovReport, and MultiNews, respectively.

[1] https://github.com/facebookresearch/llama/tree/llama_v1

Parameter	**Value**
Block Selection	
Block length in tokens	2048
Overlapping context size in tokens	200
Damping factor (d)	0.85
Trade-off param. (λ_b)	0.2
Local Context Extractor	
Optimizer	Adam
Learning rate	1E-06
Learning rate scheduler	Const.
Batch size	64
Max. gradient norm	2
Training steps	100 000
Max. input length in tokens	2048
Max. # of sentences extracted	10
Global Contetext Encoder	
# Attention heads	8
# Layers	1
Output layer size	200
Dropout	0.1
Optimizer	Adam
Learning rate	1E-06
Learning rate scheduler	Const.
Max. input length in tokens	16 384
Max. input length in sentences	1000
Batch size	64
Max. gradient norm	1
Training steps	50 000
Sentence Selector	
All selectors. Trade-off param. (λ_{sel})	0.8
Summary budget in number of tokens	
PubMed	200
BigPatent.C	100
GovReport	650
MultiNews	250
KL-Dist. # of histogram bins	40
KvD-Selector.	
Working memory ($\texttt{WM}$)	6
Min. NP cos. similarity (ν)	0.6
Recall cost (γ_{rec})	0.01

Table A.1: Hyper-parameter values for all modules in the summarization pipeline described in Chapter 5.

Appendix B

Human Evaluation Campaigns

In this appendix, we elaborate on the human evaluation campaigns run to asses content selection (Chapter 3), overall quality and cohesion in an unsupervised scenario (Chapter 4), and informativeness and cohesion in a supervised scenario (Chapter 5).

B.1 Assessing Content Selection

In this section, we provide details about the content selection campaign described in Chapter 3.

B.1.1 Campaign Interface

We use Amazon Mechanical Turk to ask human subjects if a specific key content is present in a system summary. We employ a question-answering (QA) paradigm (Clarke and Lapata, 2010; Narayan et al., 2018b, 2019) with Cloze style queries instead of factoid questions (Hermann et al., 2015). In each Human Intelligence Task (HIT), subjects are asked to read a system summary and a query, and write down the answer to said query. Queries are constructed by replacing one factual detail from the reference (gold) summary with an 'X', as can be seen in the example in Table B.1.

In regards to qualification criteria, we required annotators to have an HIT approval rate higher than 99%, a minimum of 10 000 approved HITs, be proficient in the English language, and have worked in the healthcare or medical sector before. The payment is set to $\$15/\text{hour}$ which translates to \$0.50 per task at an average of 2 minutes per task. This timing was determined through a pilot internal run.

In regards of the content of the HIT's, we randomly sampled 50 documents from the test set and manually constructed three queries per document, blurring only one piece of information per query. Each `document-system-query` combination was answered by three subjects, for systems ORACLE, SUB-EXP (tree size 20), NOTREE, and PACSUM, a total of 1800 HIT's. We deployed the task items in batches (one `system-query` combination at a time) to ensure that any single participant is not exposed to system summaries of the same document or queries built from the same reference summary.

The answers obtained from subjects, as well as the gold answers, were cleaned by stemming and removal of stopwords and whitespaces.

B.1.2 Answer Categories

Table B.2 presents a complete break-down of number of answers per category, as defined in § 3.3.6, for all systems.

B.2 Assessing Informativeness and Cohesion in an Unsupervised Scenario

In this section, we provide further details of the evaluation campaigns described in Chapter 4. The first one evaluated informativeness using a ranking approach, whereas the second, cohesion through annotation of content unit chains.

B.2.1 Catch Controls

The two studies were deployed in the Amazon Mechanical Turker (AMT) platform. Annotators were awarded \$1 per Human Intelligence Task (HIT), translating to more than \$15 per hour. These rates were calculated by measuring the average annotation time per HIT in a pilot study. Similar to the study described in the previous section, we ensured the quality of annotations by requiring annotators to have an HIT approval rate higher than 99%, a minimum of 10 000 approved HITs, be proficient in the English language, and have worked in the healthcare or medical sector before. Furthermore, we implemented the following catch controls: (i) we asked participants to check checkboxes confirming they had read the instructions and examples provided, and (ii) we discard HITs that were annotated in less than 5 minutes.[1] Annotations that failed the controls were discarded in order to maximize the quality.

[1] Time threshold obtained from pilot study measurements.

Task Description

You will be given the abstract of a scientific article along with a short text. The abstract is missing one key information bit (replaced by "X"). Find a contiguous span that contains X and write it (or copy-paste it) in the text entry

Example

Article Excerpt

The higher prevalence in the series of Barba et al. compared to ours may either reflect a difference in the definition of PLA or a selection bias since all patients in their study had undergone intracranial electrode implantation.

Sampled the insula in 50 consecutive patients with TLE on the basis of ICTAL symptoms or SCALP VEEG data suggesting an early spread of seizures either to the suprasylvian opercular cortex (e.g., lip and face paresthesiae, tonic-clonic movements of the face, dysarthria, motor aphasia, gustatory illusions, hypersalivation, and postictal facial paresis) or the infrasylvian opercular cortex (e.g., auditory hallucinations, early sensory aphasia).

The retrospective nature of this study may be associated with a recall bias for the incidence and characteristics of SSA/PLA.

Finally, as mentioned previously, our data does not allow drawing conclusions about the prognosis of SSA/PLA in non-lesional temporal lobe-like epilepsy as numbers are too small.

Most patients with pharmacoresistant lesional TLE appear to have a favorable outcome following temporal lobectomy, even in the presence of SSA and PLA.

Abstract

Purpose. Somatosensory (SSA) and pharyngolaryngeal auras (PLA) may suggest an extratemporal onset (e.g., insula, second somatosensory area).

We sought to determine the prognostic significance of SSA and PLA in **X** patients undergoing epilepsy surgery.

Methods. Retrospective review of all patients operated for refractory **X** at our institution between January 1980 and July 2007 comparing outcome between patients with SSA/PLA to those without.

Results. 158 patients underwent surgery for pharmacoresistant **X** in our institution.

Eleven (7%) experienced SSA/PLA as part of their seizures.

All but one had lesional (including hippocampal atrophy/sclerosis) **X**.

Compared to patients without SSA or PLA, these patients were older ($p = 0.049$), had a higher prevalence of early ICTAL motor symptoms ($p = 0.022$) and prior CNS infection ($p = 0.022$), and were less likely to have a localizing spect study ($p = 0.025$).

A favorable outcome was achieved in 81.8% of patients with SSA and/or PLA and 90.4% of those without SSA or PLA ($p > 0.05$).

Conclusion . Most patients with pharmacoresistant lesional **X** appear to have a favorable outcome following temporal lobectomy, even in the presence of SSA and PLA.

Content X: ['answer here']

Table B.1: Example task from the human evaluation campaign on content selection, described in Chapter 3. 'Article Excerpt' is a system summary and 'Abstract' is the gold summary modified as query.

B.2.2 Campaign Interface

Figure B.1 depicts the instructions given to annotators for each campaign, whereas Figure B.2 and B.3 present example HITs.

Category	Model				Total
	ORACLE	Sub-Exp	NOTREE	PACSUM	
Exact match	286	237	237	216	976
Synonymity	29	29	25	27	110
Specificity	7	33	18	16	74
Incompleteness	47	41	46	44	178
Incorrectness	68	89	106	44	307
Not found	13	21	18	103	155

Table B.2: Break-down of human answers by category, for summarization systems analyzed in the human campaign on content selection over the PUBMED dataset.

B.3 Assessment of Summary Qualities in a Supervised Scenario

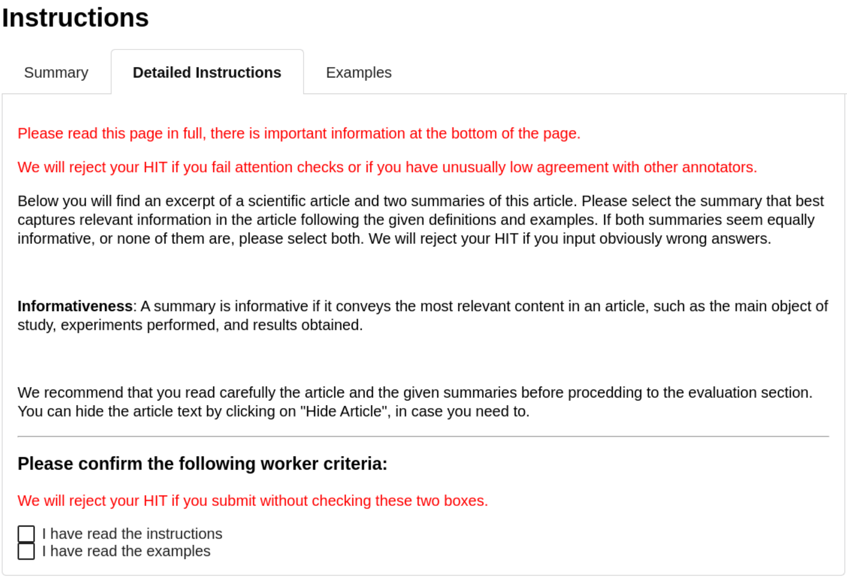
In this section, we further elaborate on the ranking evaluation campaign described in Chapter 5, which aimed at evaluating summary qualities –overall quality, informativeness, and cohesion– in a holistic manner. Moreover, we provide details about the chaining campaign.

B.3.1 Catch Controls

Similarly to previous sections, both campaigns were run on AMT, where Turkers were required to have a HIT approval rate higher than 99%, a minimum of 10 000 approved HITs, be proficient in the English language, and have worked in the healthcare or medical sector before. Annotators were awarded $1 per HIT, translating to more than $15 per hour. These rates were calculated by measuring the average annotation time per HIT in a pilot study. Furthermore, we implemented the following catch controls: (i) we asked participants to check checkboxes confirming they had read the instructions and examples provided, and (ii) we discard HITs that were annotated in less than 5 minutes, with the time threshold obtained from pilot study measurements. Annotations that failed the controls were discarded in order to maximize the quality.

B.3.2 Ranking Campaign

We collected three annotations per system-pair comparison and made sure that the same annotator was not exposed to the same document twice. As an additional catch trial, we included in each annotation batch an extra instance with summaries extracted by the extractive



Instructions

Please read this page in full, there is important information at the bottom of the page.

We will reject your HIT if you fail attention checks or if you have unusually low agreement with other annotators.

Below you will find an excerpt of a scientific article and two summaries of this article. Please select the summary that best captures relevant information in the article following the given definitions and examples. If both summaries seem equally informative, or none of them are, please select both. We will reject your HIT if you input obviously wrong answers.

Informativeness: A summary is informative if it conveys the most relevant content in an article, such as the main object of study, experiments performed, and results obtained.

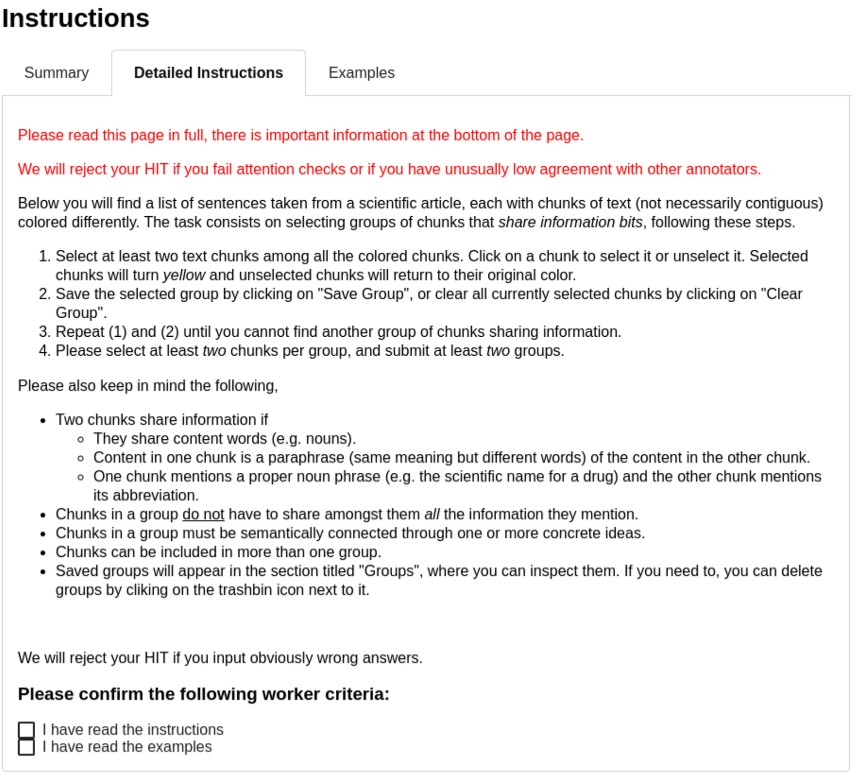
We recommend that you read carefully the article and the given summaries before procedding to the evaluation section. You can hide the article text by clicking on "Hide Article", in case you need to.

Please confirm the following worker criteria:

We will reject your HIT if you submit without checking these two boxes.

□ I have read the instructions
□ I have read the examples

a Ranking Campaign



Instructions

Please read this page in full, there is important information at the bottom of the page.

We will reject your HIT if you fail attention checks or if you have unusually low agreement with other annotators.

Below you will find a list of sentences taken from a scientific article, each with chunks of text (not necessarily contiguous) colored differently. The task consists on selecting groups of chunks that *share information bits*, following these steps.

1. Select at least two text chunks among all the colored chunks. Click on a chunk to select it or unselect it. Selected chunks will turn *yellow* and unselected chunks will return to their original color.
2. Save the selected group by clicking on "Save Group", or clear all currently selected chunks by clicking on "Clear Group".
3. Repeat (1) and (2) until you cannot find another group of chunks sharing information.
4. Please select at least *two* chunks per group, and submit at least *two* groups.

Please also keep in mind the following,

- Two chunks share information if
  - They share content words (e.g. nouns).
  - Content in one chunk is a paraphrase (same meaning but different words) of the content in the other chunk.
  - One chunk mentions a proper noun phrase (e.g. the scientific name for a drug) and the other chunk mentions its abbreviation.
- Chunks in a group do not have to share amongst them *all* the information they mention.
- Chunks in a group must be semantically connected through one or more concrete ideas.
- Chunks can be included in more than one group.
- Saved groups will appear in the section titled "Groups", where you can inspect them. If you need to, you can delete groups by cliking on the trashbin icon next to it.

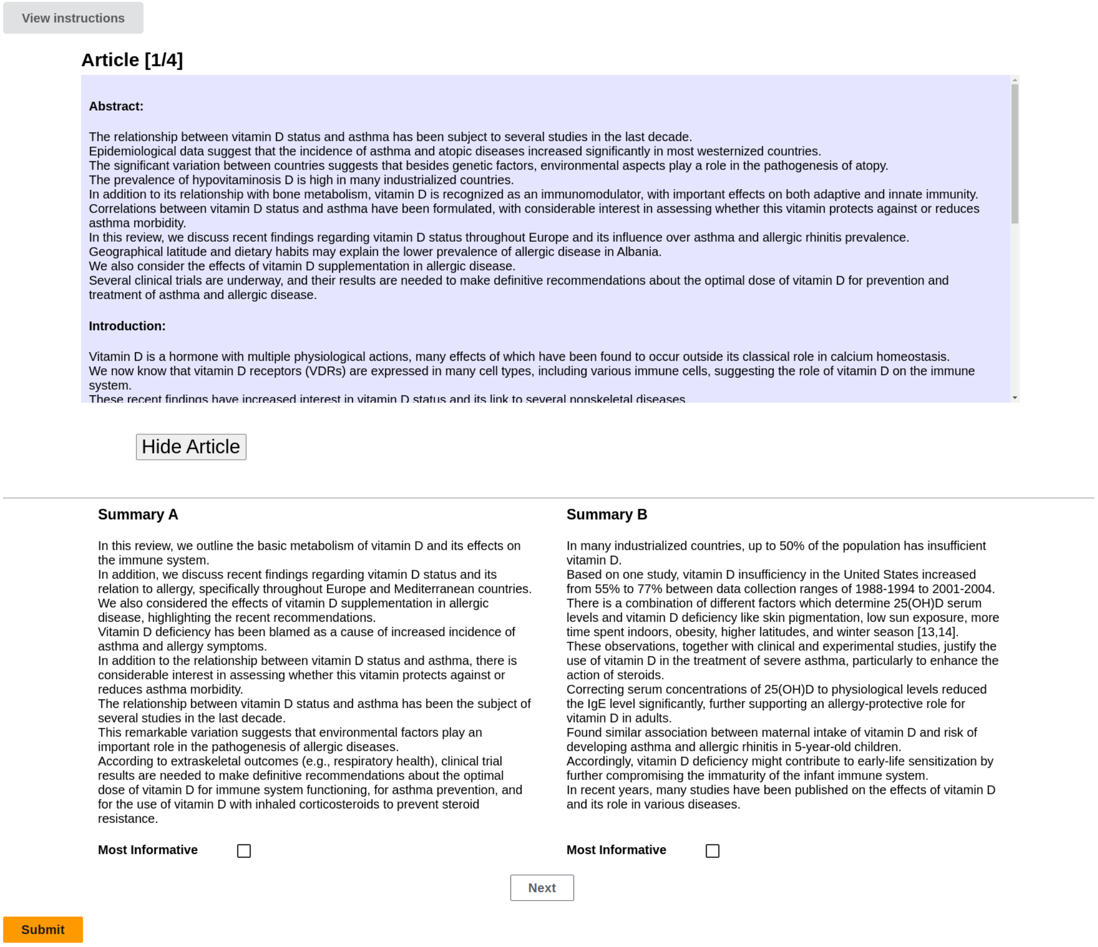
We will reject your HIT if you input obviously wrong answers.

Please confirm the following worker criteria:

□ I have read the instructions
□ I have read the examples

b Chaining Campaign

Figure B.1: Instructions given to annotators in the ranking (top) and chaining campaigns (bottom) in Chapter 4.



Article [1/4]

Abstract:

The relationship between vitamin D status and asthma has been subject to several studies in the last decade.
Epidemiological data suggest that the incidence of asthma and atopic diseases increased significantly in most westernized countries.
The significant variation between countries suggests that besides genetic factors, environmental aspects play a role in the pathogenesis of atopy.
The prevalence of hypovitaminosis D is high in many industrialized countries.
In addition to its relationship with bone metabolism, vitamin D is recognized as an immunomodulator, with important effects on both adaptive and innate immunity.
Correlations between vitamin D status and asthma have been formulated, with considerable interest in assessing whether this vitamin protects against or reduces asthma morbidity.
In this review, we discuss recent findings regarding vitamin D status throughout Europe and its influence over asthma and allergic rhinitis prevalence.
Geographical latitude and dietary habits may explain the lower prevalence of allergic disease in Albania.
We also consider the effects of vitamin D supplementation in allergic disease.
Several clinical trials are underway, and their results are needed to make definitive recommendations about the optimal dose of vitamin D for prevention and treatment of asthma and allergic disease.

Introduction:

Vitamin D is a hormone with multiple physiological actions, many effects of which have been found to occur outside its classical role in calcium homeostasis.
We now know that vitamin D receptors (VDRs) are expressed in many cell types, including various immune cells, suggesting the role of vitamin D on the immune system.
These recent findings have increased interest in vitamin D status and its link to several nonskeletal diseases.

Summary A

In this review, we outline the basic metabolism of vitamin D and its effects on the immune system.
In addition, we discuss recent findings regarding vitamin D status and its relation to allergy, specifically throughout Europe and Mediterranean countries.
We also considered the effects of vitamin D supplementation in allergic disease, highlighting the recent recommendations.
Vitamin D deficiency has been blamed as a cause of increased incidence of asthma and allergy symptoms.
In addition to the relationship between vitamin D status and asthma, there is considerable interest in assessing whether this vitamin protects against or reduces asthma morbidity.
The relationship between vitamin D status and asthma has been the subject of several studies in the last decade.
This remarkable variation suggests that environmental factors play an important role in the pathogenesis of allergic diseases.
According to extraskeletal outcomes (e.g., respiratory health), clinical trial results are needed to make definitive recommendations about the optimal dose of vitamin D for immune system functioning, for asthma prevention, and for the use of vitamin D with inhaled corticosteroids to prevent steroid resistance.

Most Informative ☐

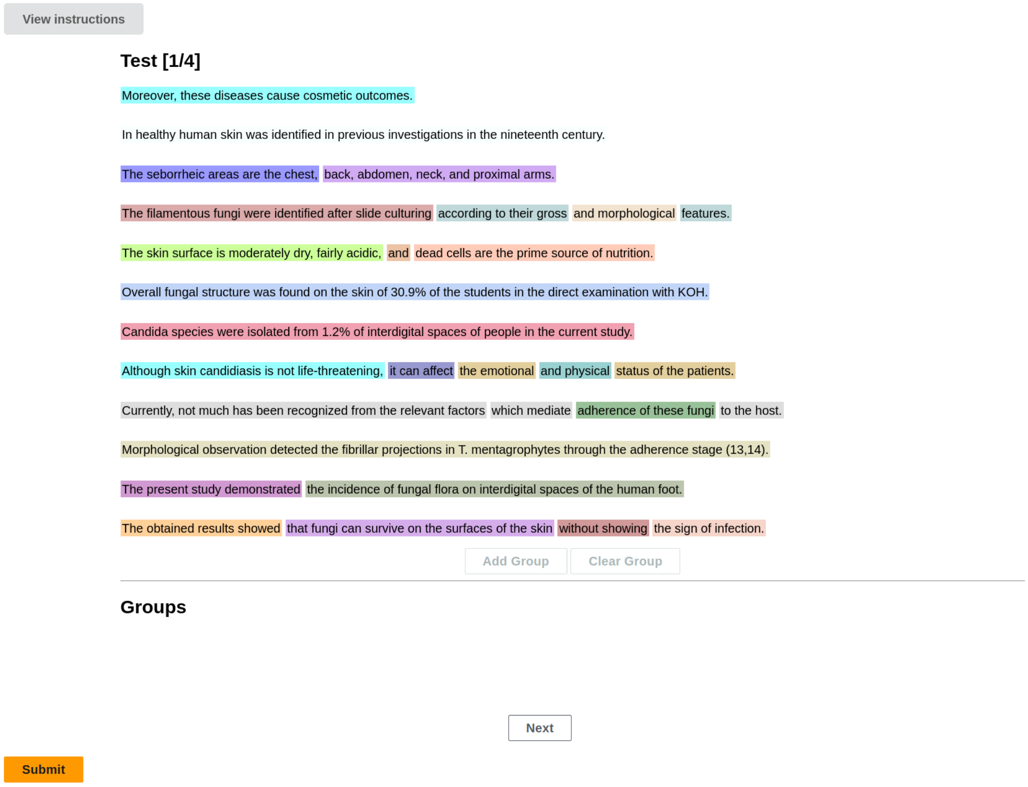
Summary B

In many industrialized countries, up to 50% of the population has insufficient vitamin D.
Based on one study, vitamin D insufficiency in the United States increased from 55% to 77% between data collection ranges of 1988-1994 to 2001-2004.
There is a combination of different factors which determine 25(OH)D serum levels and vitamin D deficiency like skin pigmentation, low sun exposure, more time spent indoors, obesity, higher latitudes, and winter season [13,14].
These observations, together with clinical and experimental studies, justify the use of vitamin D in the treatment of severe asthma, particularly to enhance the action of steroids.
Correcting serum concentrations of 25(OH)D to physiological levels reduced the IgE level significantly, further supporting an allergy-protective role for vitamin D in adults.
Found similar association between maternal intake of vitamin D and risk of developing asthma and allergic rhinitis in 5-year-old children.
Accordingly, vitamin D deficiency might contribute to early-life sensitization by further compromising the immaturity of the infant immune system.
In recent years, many studies have been published on the effects of vitamin D and its role in various diseases.

Most Informative ☐

Figure B.2: Example task from the informativeness ranking campaign on Amazon Mechanical Turk, described in Chapter 4.

oracle and the random baseline. After discarding annotations that failed the controls, we are left with 708 out of 810 instances (30 documents, 3 system pairs, 3 dimensions, and 3 annotations per pair). Figure B.4 depicts the instructions given to annotators for each campaign, and Figure B.5 presents and example.



Test [1/4]

Moreover, these diseases cause cosmetic outcomes.

In healthy human skin was identified in previous investigations in the nineteenth century.

The seborrheic areas are the chest, back, abdomen, neck, and proximal arms.

The filamentous fungi were identified after slide culturing according to their gross and morphological features.

The skin surface is moderately dry, fairly acidic, and dead cells are the prime source of nutrition.

Overall fungal structure was found on the skin of 30.9% of the students in the direct examination with KOH.

Candida species were isolated from 1.2% of interdigital spaces of people in the current study.

Although skin candidiasis is not life-threatening, it can affect the emotional and physical status of the patients.

Currently, not much has been recognized from the relevant factors which mediate adherence of these fungi to the host.

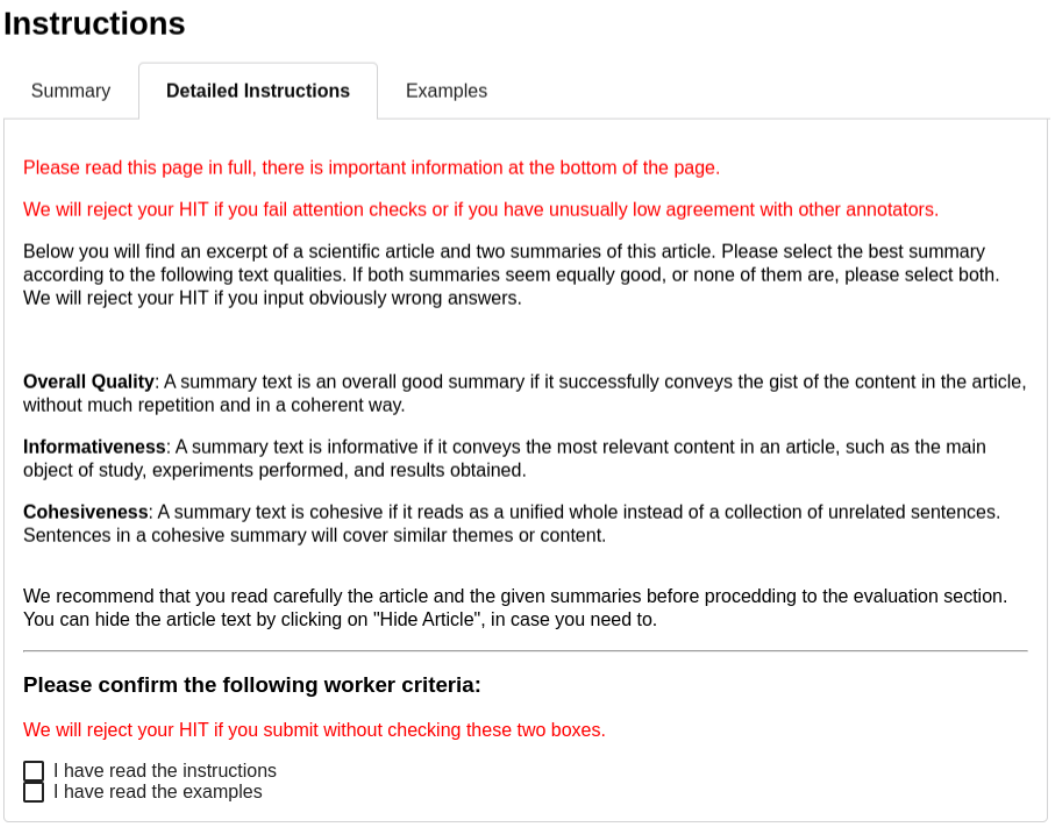
Morphological observation detected the fibrillar projections in T. mentagrophytes through the adherence stage (13,14).

The present study demonstrated the incidence of fungal flora on interdigital spaces of the human foot.

The obtained results showed that fungi can survive on the surfaces of the skin without showing the sign of infection.

Groups

Figure B.3: Example task from the chaining campaign on Amazon Mechanical Turk, described in Chapter 4.



Instructions

Please read this page in full, there is important information at the bottom of the page.

We will reject your HIT if you fail attention checks or if you have unusually low agreement with other annotators.

Below you will find an excerpt of a scientific article and two summaries of this article. Please select the best summary according to the following text qualities. If both summaries seem equally good, or none of them are, please select both. We will reject your HIT if you input obviously wrong answers.

Overall Quality: A summary text is an overall good summary if it successfully conveys the gist of the content in the article, without much repetition and in a coherent way.

Informativeness: A summary text is informative if it conveys the most relevant content in an article, such as the main object of study, experiments performed, and results obtained.

Cohesiveness: A summary text is cohesive if it reads as a unified whole instead of a collection of unrelated sentences. Sentences in a cohesive summary will cover similar themes or content.

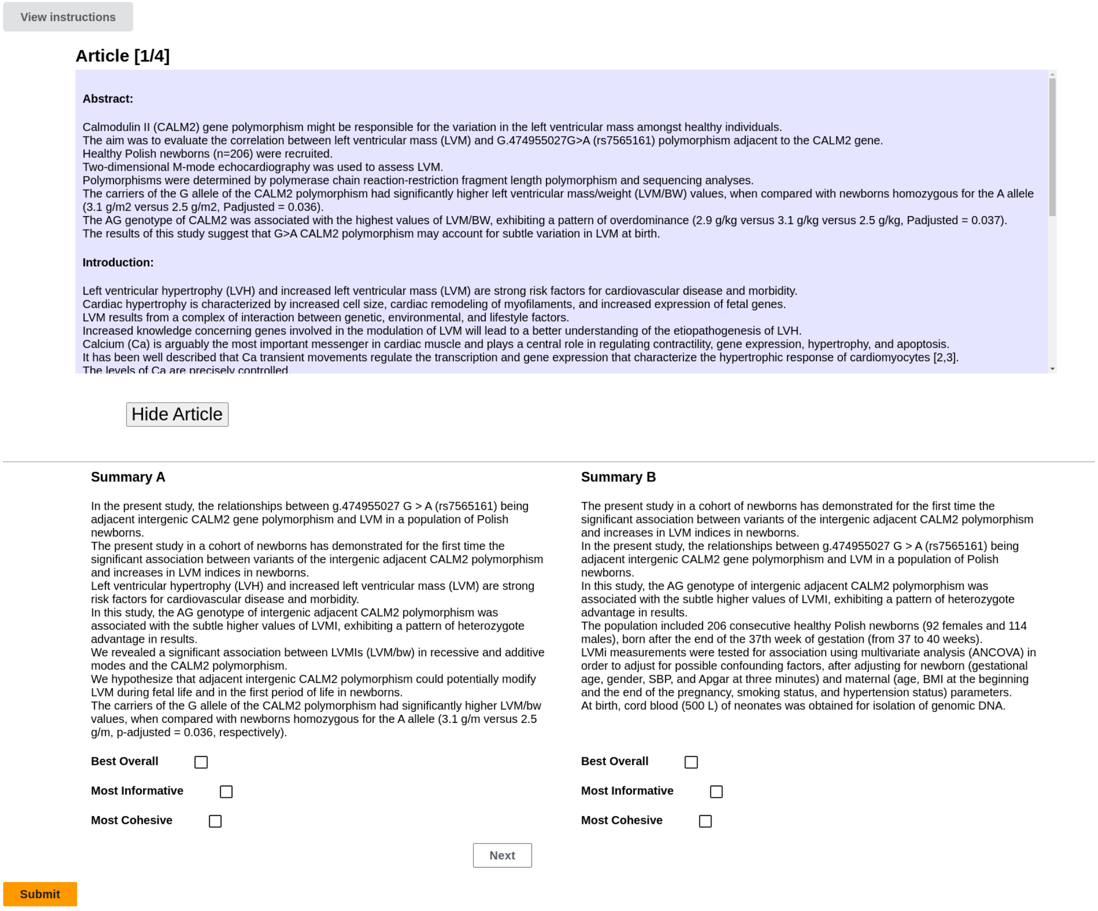
We recommend that you read carefully the article and the given summaries before procedding to the evaluation section. You can hide the article text by clicking on "Hide Article", in case you need to.

Please confirm the following worker criteria:

We will reject your HIT if you submit without checking these two boxes.

□ I have read the instructions
□ I have read the examples

Figure B.4: Instructions given to annotators in the ranking campaign in Chapter 5.



Article [1/4]

Abstract:

Calmodulin II (CALM2) gene polymorphism might be responsible for the variation in the left ventricular mass amongst healthy individuals.
The aim was to evaluate the correlation between left ventricular mass (LVM) and G.474955027G>A (rs7565161) polymorphism adjacent to the CALM2 gene.
Healthy Polish newborns (n=206) were recruited.
Two-dimensional M-mode echocardiography was used to assess LVM.
Polymorphisms were determined by polymerase chain reaction-restriction fragment length polymorphism and sequencing analyses.
The carriers of the G allele of the CALM2 polymorphism had significantly higher left ventricular mass/weight (LVM/BW) values, when compared with newborns homozygous for the A allele (3.1 g/m2 versus 2.5 g/m2, Padjusted = 0.036).
The AG genotype of CALM2 was associated with the highest values of LVM/BW, exhibiting a pattern of overdominance (2.9 g/kg versus 3.1 g/kg versus 2.5 g/kg, Padjusted = 0.037).
The results of this study suggest that G>A CALM2 polymorphism may account for subtle variation in LVM at birth.

Introduction:

Left ventricular hypertrophy (LVH) and increased left ventricular mass (LVM) are strong risk factors for cardiovascular disease and morbidity.
Cardiac hypertrophy is characterized by increased cell size, cardiac remodeling of myofilaments, and increased expression of fetal genes.
LVM results from a complex of interaction between genetic, environmental, and lifestyle factors.
Increased knowledge concerning genes involved in the modulation of LVM will lead to a better understanding of the etiopathogenesis of LVH.
Calcium (Ca) is arguably the most important messenger in cardiac muscle and plays a central role in regulating contractility, gene expression, hypertrophy, and apoptosis.
It has been well described that Ca transient movements regulate the transcription and gene expression that characterize the hypertrophic response of cardiomyocytes [2,3].
The levels of Ca are precisely controlled.

Summary A

In the present study, the relationships between g.474955027 G > A (rs7565161) being adjacent intergenic CALM2 gene polymorphism and LVM in a population of Polish newborns.
The present study in a cohort of newborns has demonstrated for the first time the significant association between variants of the intergenic adjacent CALM2 polymorphism and increases in LVM indices in newborns.
Left ventricular hypertrophy (LVH) and increased left ventricular mass (LVM) are strong risk factors for cardiovascular disease and morbidity.
In this study, the AG genotype of intergenic adjacent CALM2 polymorphism was associated with the subtle higher values of LVMI, exhibiting a pattern of heterozygote advantage in results.
We revealed a significant association between LVMIs (LVM/bw) in recessive and additive modes and the CALM2 polymorphism.
We hypothesize that adjacent intergenic CALM2 polymorphism could potentially modify LVM during fetal life and in the first period of life in newborns.
The carriers of the G allele of the CALM2 polymorphism had significantly higher LVM/bw values, when compared with newborns homozygous for the A allele (3.1 g/m versus 2.5 g/m, p-adjusted = 0.036, respectively).

Best Overall ☐

Most Informative ☐

Most Cohesive ☐

Summary B

The present study in a cohort of newborns has demonstrated for the first time the significant association between variants of the intergenic adjacent CALM2 polymorphism and increases in LVM indices in newborns.
In the present study, the relationships between g.474955027 G > A (rs7565161) being adjacent intergenic CALM2 gene polymorphism and LVM in a population of Polish newborns.
In this study, the AG genotype of intergenic adjacent CALM2 polymorphism was associated with the subtle higher values of LVMI, exhibiting a pattern of heterozygote advantage in results.
The population included 206 consecutive healthy Polish newborns (92 females and 114 males), born after the end of the 37th week of gestation (from 37 to 40 weeks).
LVMi measurements were tested for association using multivariate analysis (ANCOVA) in order to adjust for possible confounding factors, after adjusting for newborn (gestational age, gender, SBP, and Apgar at three minutes) and maternal (age, BMI at the beginning and the end of the pregnancy, smoking status, and hypertension status) parameters.
At birth, cord blood (500 L) of neonates was obtained for isolation of genomic DNA.

Best Overall ☐

Most Informative ☐

Most Cohesive ☐

Figure B.5: Example task from the ranking campaign on Amazon Mechanical Turk, described in Chapter 5.

Bibliography

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chinatsu Aone, Mary Ellen Okurowski, James Gorlinsky, and Bjornar Larsen. 1997. A scalable summarization system using robust nlp. In *Intelligent Scalable Text Summarization*.

Alan D. Baddeley. 2018. *Exploring working memory : selected works of Alan Baddeley / Alan Baddeley.* World library of psychologists. Routledge, Taylor & Francis Group, Abingdon, Oxon ;.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Intelligent Scalable Text Summarization*.

Regina Barzilay and Noemie Elhadad. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

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

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Process-*

ing (EMNLP-IJCNLP), pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Yoshua Bengio. 2017. The consciousness prior. *ArXiv preprint*, abs/1709.08568.

Marc G Berman, John Jonides, and Richard L Lewis. 2009. In search of decay in verbal short-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(2):317.

Abdulkadir Abubakar Bichi, Ruhaidah Samsudin, Rohayanti Hassan, and Khalil Almekhlafi. 2021. A review of graph-based extractive text summarization models. In *Innovative Systems for Intelligent Health Informatics*, pages 439–448, Cham. Springer International Publishing.

Geetanjali Bihani and Julia Taylor Rayz. 2024. Learning shortcuts: On the misleading promise of nlu in language models. *arXiv preprint arXiv:2401.09615*.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Wolfram Bublitz, Uta Lenk, and Eija Ventola. 1999. *Coherence in Spoken and Written Discourse: How to create it and how to describe it. Selected papers from the International Workshop on Coherence, Augsburg, 24-27 April 1997*, volume 63. John Benjamins Publishing.

Jaime Carbonell and Jade Goldstein. 1998a. 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*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Jaime Carbonell and Jade Goldstein. 1998b. 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*, pages 335–336.

Ronald Cardenas, Bingsheng Yao, Dakuo Wang, and Yufang Hou. 2023. 'don't get too technical with me': A discourse structure-based framework for automatic science journalism. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1186–1202, Singapore. Association for Computational Linguistics.

Hou Pong Chan, Lu Wang, and Irwin King. 2021. Controllable summarization with constrained markov decision process. *Transactions of the Association for Computational Linguistics*, 9:1213–1232.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Thirdteenth International Conference on Learning Representations*.

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)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

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

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 806–813, Portorož, Slovenia. European Language Resources Association (ELRA).

Hoa Trang Dang, Karolina Owczarzak, et al. 2008. Overview of the tac 2008 update summarization task. In *Proceedings of First Text Analysis Conference, TAC 2008*. NIST.

Anthony Christopher Davison and David Victor Hinkley. 1997. *Bootstrap methods and their application*. 1. Cambridge university press.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Lrec*, volume 6, pages 449–454.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Micha Elsner and Eugene Charniak. 2011. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084.

Yimai Fang. 2019. *Proposition-based summarization with a coherence-driven incremental model*. Ph.D. thesis, University of Cambridge.

Yimai Fang and Simone Teufel. 2014. A summariser based on human memory limitations and lexical competition. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 732–741, Gothenburg, Sweden. Association for Computational Linguistics.

Yimai Fang and Simone Teufel. 2016. Improving argument overlap for proposition-based summarisation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 479–485.

Marcio Fonseca, Yftah Ziser, and Shay B. Cohen. 2022. Factorizing content and budget decisions in abstractive summarization of long documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6341–6364, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Michel Galley and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1486–1488.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Artur d'Avila Garcez and Luis C Lamb. 2020. Neurosymbolic ai: The 3rd wave. *ArXiv preprint*, abs/2012.05876.

Simon Garrod and Anthony Sanford. 1977. Interpreting anaphoric relations: The integration of semantic information while reading. *Journal of Verbal Learning and Verbal Behavior*, 16(1):77–90.

Simon C. Garrod and Anthony J. Sanford. 1994. Resolving sentences in a discourse context: How discourse representation affects language understanding. In *Handbook of psycholinguistics.*, pages 675–698. Academic Press, San Diego, CA, US.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research*, 77:103–166.

Dan Gillick and Yang Liu. 2010. Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151.

Murray Glanzer. 1972. Storage mechanisms in recall. In *Psychology of learning and motivation*, volume 5, pages 129–193. Elsevier.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.

Tomas Goldsack, Zhihao Zhang, Chen Tang, Carolina Scarton, and Chenghua Lin. 2023. Enhancing biomedical lay summarisation with external knowledge graphs. In *Proceedings* *of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8016–8032.

E Bruce Goldstein. 2015. *Cognitive psychology: Connecting mind, research and everyday experience*. Cengage Learning Stamford, CT.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. SNaC: Coherence error detection for narrative summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 444–463, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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*, pages 128–137, Lisbon, Portugal. Association for Computational Linguistics.

Bertram Myron Gross. 1964. *The managing of organizations: The administrative struggle*, 1 edition. Free Press of Glencoe.

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.

Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland. Association for Computational Linguistics.

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)*, pages 93–103.

Mandy Guo, Joshua Ainslie, David C Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.

Trevor A Harley. 1995. *The psychology of language: From data to theory*. Erlbaum (Uk) Taylor & Francis, Publ.

Halliday Hassan, Rukaya, and Michael A. K. Halliday. 1976. *Cohesion in English*. Routledge.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Tamanna Hossain, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Xinyu Hua, Ashwin Sreevatsa, and Lu Wang. 2021. DYPLOC: Dynamic planning of content using mixed language models for text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6408–6423, Online. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436.

Sungho Jeon and Michael Strube. 2020. Centering-based neural coherence modeling with hierarchical discourse segments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7458–7472, Online. Association for Computational Linguistics.

Sungho Jeon and Michael Strube. 2022. Entity-based neural local coherence modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7787–7805, Dublin, Ireland. Association for Computational Linguistics.

Ruipeng Jia, Yanan Cao, Fang Fang, Yuchen Zhou, Zheng Fang, Yanbing Liu, and Shi Wang. 2021. Deep differential amplifier for extractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 366–376, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Barbara Johnstone. 1994. *Repetition in discourse : interdisciplinary perspectives*. Advances in discourse processes. Ablex Publishing Corporation.

Karen Spärck Jones. 1999. Automatic summarizing: factors and directions. *Advances in Automatic Text Summarization*.

Prathyusha Jwalapuram, Shafiq Joty, and Xiang Lin. 2022. Rethinking self-supervision objectives for generalizable coherence modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6044–6059, Dublin, Ireland. Association for Computational Linguistics.

Hans Kamp and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition*, 57(2):129–191.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 2. Springer Science & Business Media.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Eileen Kintsch. 1990. Macroprocesses and microprocesses in the development of summarization skill. *Cognition and instruction*, 7(3):161–195.

Walter Kintsch. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological review*, 95(2):163.

Walter Kintsch and Teun A van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

Walter Kintsch and CBEMAFRS Walter Kintsch. 1998. *Comprehension: A paradigm for cognition*. Cambridge university press.

Donald E Knuth, Tracy Larrabee, and Paul M Roberts. 1989. *Mathematical writing*. 14. Cambridge University Press.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. *Computing*, 1:25–2011.

Kundan Krishna, Sopan Khosla, Jeffrey Bigham, and Zachary C. Lipton. 2021. Generating SOAP notes from doctor-patient conversations using modular summarization techniques. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4958–4972, Online. Association for Computational Linguistics.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. Booksum: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73.

Philippe Laban, Luke Dai, Lucas Bandarkar, and Marti A. Hearst. 2021. Can transformer models measure coherence in text: Re-thinking the shuffle test. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1058–1064, Online. Association for Computational Linguistics.

Jack Lanchantin, Shubham Toshniwal, Jason Weston, Sainbayar Sukhbaatar, et al. 2024. Learning to reason and memorize with self-notes. *Advances in Neural Information Processing Systems*, 36.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*.

Juhani Lehto. 1996. Working memory capacity and summarizing skills in ninth-graders. *Scandinavian Journal of Psychology*, 37(1):84–92.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Pengfei Liu, et al. 2023. Generative judge for evaluating alignment. In *The Twelfth International Conference on Learning Representations*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract meaning representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190.

Chin-Yew Lin. 1999. Training a selection function for extraction. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 55–62.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 768–774.

Marina Litvak and Natalia Vanetik. 2017. Query-based summarization using MDL principle. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 22–31, Valencia, Spain. Association for Computational Linguistics.

Feifan Liu and Yang Liu. 2008. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Ling Luo, Xiang Ao, Yan Song, Feiyang Pan, Min Yang, and Qing He. 2019. Reading like HER: Human reading inspired extractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3033–3043, Hong Kong, China. Association for Computational Linguistics.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. Felix: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255.

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

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Daniel Marcu. 1998. To build text summaries of high quality, nuclearity is not sufficient. In *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, pages 1–8.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT press.

Mohsen Mesgar, Leonardo F. R. Ribeiro, and Iryna Gurevych. 2021. A neural graph-based local coherence model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2316–2321, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*, pages 1414–1423, San Diego, California. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ritwik Mishra and Tirthankar Gayen. 2018. Automatic lossless-summarization of news articles with abstract meaning representation. *Procedia Computer Science*, 135:178–185.

Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. 2021. Improving factual completeness and consistency of image-to-text radiology report generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5288–5304, Online. Association for Computational Linguistics.

Han Cheol Moon, Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A unified neural coherence model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

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.

Shashi Narayan, Ronald Cardenas, Nikos Papasarantopoulos, Shay B. Cohen, Mirella Lapata, Jiangsheng Yu, and Yi Chang. 2018a. Document modeling with external attention for sentence extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2020–2030, Melbourne, Australia. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2019. What is this article about? extreme summarization with topic-aware convolutional neural networks. *Journal of Artificial Intelligence Research*, 66:243–278.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaz Bratanic, and Ryan McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4143–4159, Online. Association for Computational Linguistics.

Shashi Narayan, Gonçalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata. 2022. A well-composed text is half done! composition sampling for diverse conditional generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1339.

Ali Naserasadi, Hamid Khosravi, and Faramarz Sadeghi. 2019. Extractive multi-document summarization based on textual entailment and sentence compression via knapsack problem. *Natural Language Engineering*, 25(1):121–146.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2):4–es.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 104–111.

Iftitahu Nimah, Meng Fang, Vlado Menkovski, and Mykola Pechenizkiy. 2023. Nlg evaluation metrics beyond correlation analysis: An empirical metric preference checklist. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1240–1266.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

Kenji Ono, Kazuo Sumita, and Seiji Miike. 1994. Abstract generation based on rhetorical structure extraction. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, Kyoto, Japan.

Miles Osborne. 2002. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 1–8.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

Lawrence Page. 1998. The pagerank citation ranking: Bringing order to the web. technical report. *Stanford Digital Library Technologies Project, 1998*.

Rebecca J Passonneau. 2010. Formal and functional assessment of the pyramid method for summary content evaluation. *Natural Language Engineering*, 16(2):107–131.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, New Orleans, Louisiana. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations ICLR 2018*.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy. Association for Computational Linguistics.

Maxime Peyrard. 2019. A simple theoretical model of importance for summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy. Association for Computational Linguistics.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.

Leo Postman and Laura W Phillips. 1965. Short-term temporal changes in free recall. *Quarterly journal of experimental psychology*, 17(2):132–138.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, The Thirty-First Innovative Applications of Artificial Intelligence Conference IAAI, The Ninth Symposium on Educational Advances in Artificial Intelligence, EAAI*, pages 6908–6915. AAAI Press.

Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 223–228, Florence, Italy. Association for Computational Linguistics.

Yifu Qiu and Shay B. Cohen. 2022. Abstractive summarization guided by latent hierarchical document structure. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5317, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 492–501.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. Citeseer.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80.

Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 41–45. Association for Computational Linguistics.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5032–5042, Online. Association for Computational Linguistics.

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

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. 2018. A graph-theoretic summary evaluation for ROUGE. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 762–767, Brussels, Belgium. Association for Computational Linguistics.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of NAACL-HLT*, pages 682–687.

Ori Shapira, Hadar Ronen, Meni Adler, Yael Amsterdamer, Judit Bar-Ilan, and Ido Dagan. 2017. Interactive abstractive summarization for event news tweets. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 109–114, Copenhagen, Denmark. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213.

Yotam Shichel, Meir Kalech, and Oren Tsur. 2021. With measured words: Simple sentence selection for black-box optimization of sentence compression algorithms. In *16th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2021*, pages 1625–1634. Association for Computational Linguistics (ACL).

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference* *on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Arie S Spirgel and Peter F Delaney. 2016. Does writing summaries improve memory for text? *Educational Psychology Review*, 28(1):171–196.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Julius Steen and Katja Markert. 2022. How to find strong summary coherence measures? a toolbox and a comparative study for summary coherence measure evaluation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6035–6049, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Maite Taboada. 2004. *Building Coherence and Cohesion: Task-oriented dialogue in English and Spanish*. John Benjamins.

Ana María Vigara Tauste. 1995. Comodidad y recurrencia en la organización del discurso coloquial. In *El español coloquial: actas del I Simposio sobre análisis del discurso oral: Almería, 23-25 de noviembre de 1994*, pages 173–208. Servicio de Publicaciones.

Petroc Taylor. 2023. Data growth worldwide 2010-2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yuji Ushiro, Shuichi Takaki, Mayuko Kobayashi, Yusuke Hasegawa, Shingo Nahatame, Akira Hamada, and Yukino Kimura. 2013. Measures of macroproposition construction in efl reading: Summary writing task vs. the meaning identification technique. *JLTA Journal*, 16:185–204.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Marilyn A Walker. 1993. *Informational redundancy and resource bounds in dialogue*. Ph.D. thesis, University of Pennsylvania.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *ArXiv preprint*, abs/2006.04768.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3752–3761.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18*, pages 5602–5609. AAAI Press.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528, Suzhou, China. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Dani Yogatama, Fei Liu, and Noah A Smith. 2015. Extractive summarization by maximizing semantic volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

David Zajic, Bonnie J Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Renxian Zhang, Wenjie Li, Naishi Liu, and Dehong Gao. 2016. Coherent narrative summarization with a cognitive model. *Computer Speech & Language*, 35:134–160.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shiyue Zhang, David Wan, and Mohit Bansal. 2023. Extractive is not faithful: An investigation of broad unfaithfulness problems in extractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2153–2174, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Michael Strube, and Steffen Eger. 2023. Discoscore: Evaluating text generation with bert and discourse coherence. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3847–3865.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663.