On the trade-off between redundancy and cohesiveness in extractive summarization

Journal of Artificial Intelligence Research 80 (2024) 273-326 Submitted 07/2023; published 06/2024

On the Trade-off between Redundancy and Cohesion in Extractive Summarization

Ronald Cardenas RONALD.CARDENAS@ED.AC.UK
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, UK

Matthias Gallé MATTHIAS@COHERE.COM
Cohere
51 Great Marlborough St, London, UK

Shay B. Cohen SCOHEN@INF.ED.AC.UK
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, UK

Abstract

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 paper, 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 implement a psycholinguistic theory that simulates how humans keep track of relevant content units and 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 analyzed summary properties.

1. Introduction

Automatic single-document summarization is the task of reading a text document and presenting an end-user (be it a human user or a module down a processing pipeline) with a shorter text, the *summary*, that retains the gist of the information consumed in the document. Such a complex task 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 relevant with respect to the source document, non-redundant among themselves, and

informative to the end-user; 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 & Vanetik, 2017; Shapira et al., 2017; Narayan et al., 2019; Xiao & 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 & 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 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 & 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 & Strube, 2020, 2022) and to incorporate into the summarization task without sacrificing performance in other aspects such as informativeness (Wu & Hu, 2018; Xu, Gan, Cheng, & Liu, 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 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 (Halliday & Hasan, 1976) such as word repetitions, pronouns, anaphoric expressions, and conjunctions (S. Garrod & 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 (E. Kintsch, 1990; S. C. Garrod & 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; W. Kintsch & Walter Kintsch, 1998; Ushiro et al., 2013; Spirgel & Delaney, 2016).

In this work, we investigate the trade-offs automatic summarization systems incur 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

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]$_\diamond$ ...
In this paper, 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]$_\diamond$ ...

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 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.

–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. In the RL setup, we compare systems that aim to balance informativeness and redundancy, against those which balance informativeness and cohesion. We model this trade-off as a linear combination of property-specific rewards, e.g. by 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 lexical cohesion. These models implement the Micro-Macro Structure theory of text comprehension (W. Kintsch & van Dijk, 1978), henceforth called KvD, which provides a principled way of discretizing content into semantic units and organizing them in short and long-term memory. 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, indicating the effectiveness of constraining working memory to keep only the most relevant units, compared to modeling relevancy through sentence centrality as done by the analyzed baseline. In the second study, we found that explicitly enforcing lexical cohesive links during reading allows the proposed KvD systems to extract summaries that exhibit a smooth topic transition between adjacent or near-adjacent sentences, with cohesive links connecting most sentences in the extracted summary. 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.[1]

The rest of the paper is organized as follows. An overview of previous related work is presented in § 2, followed by the problem formulation of the reward-guided control scenario in § 3. Then, § 4 elaborates on control strategies in the unsupervised scenario, providing a detailed description of the KvD theory(§ 4.1) and the proposed systems (§ 4.2). Lastly, § 5 and §6 describe our experimental setup and discuss our results, respectively.

2. Related Work

In this section we discuss previous efforts related to automatic summarization, both traditional and modern (neural based), how the problems of redundancy and cohesion are being tackled, and how cognitive science has influenced automatic summarization.

2.1 Summarization Approaches

Early approaches represented and organized content in a document using semantic and discourse methods such as lexical chains (Barzilay & Elhadad, 1997; Silber & McCoy, 2002), latent semantic analysis (Gong & Liu, 2001; Hachey et al., 2006), coreference information (Baldwin & Morton, 1998; Steinberger et al., 2007), and rhetorical structure theory (Ono et al., 1994; Marcu, 1998). In particular, graph representations proved effective in encoding relations between content units such as discourse relations (Wolf & Gibson, 2004; Louis et al., 2010) and word co-occurrence statistics (Mihalcea & Tarau, 2004; Erkan & Radev, 2004). After obtaining a representation of a document, the selection of content units (usually sentences) is posed as a unit ranking problem or a sequence labeling problem in which each

1. Code available at `https://github.com/ronaldahmed/trade-off-kvd/`

unit is labeled as 'select' or 'not select'. For this selection stage, machine learning approaches have proven effective at identifying summary-worthy units (i.e. relevant and informative) by leveraging manually-crafted features such as word frequency (Vanderwende et al., 2007; Nenkova et al., 2006), sentence length (Radev et al., 2004), and the presence of keywords of proper nouns (Kupiec et al., 1995; Jones, 2007).

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, Cardenas, et al., 2018; Narayan, Cohen, & Lapata, 2018; Cheng & Lapata, 2016), Transformers (Song et al., 2019; L. Dong et al., 2019) and lately by leveraging large pretrained language models (Zheng & Lapata, 2019; Y. Liu & Lapata, 2019; J. Zhang et al., 2020). Building upon traditional methods, neural summarization models leverage discourse (Clarke & Lapata, 2010; Cohan et al., 2018), topical (Narayan et al., 2019), and graph representations (Bichi et al., 2021; Qiu & Cohen, 2022). Even though most research concentrates on summarization of middle-sized documents like news articles and Reddit posts (Völske, Potthast, Syed, & Stein, 2017), recent work has shifted attention to long document summarization and its challenges (Cohan et al., 2018; Sharma et al., 2019; Xiao & Carenini, 2019; Fonseca, Ziser, & Cohen, 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 and extracting a summary in linear time w.r.t. the number of sentences. Note that the proposed systems do not employ neural networks during content representation or selection but instead operate over cognitively inspired data structures of propositions representing human memory.

Finally, of special interest to this work are unsupervised approaches to summarization, an area not explored as much as its supervised counterpart given the availability of large summarization datasets nowadays (Hermann et al., 2015; Cohan et al., 2018; Narayan et al., 2019). Central to most extractive approaches is a weighted graph representation of the source document (Bichi et al., 2021) followed by sentence ranking based on node centrality, where edge weights are calculated by TF-IDF (Mihalcea & Tarau, 2004) or by finetuned, dedicated architectures (Zheng & Lapata, 2019). Our work differs from this line of research in two aspects. First, content is organized in tree and graph structures where nodes are modeled as propositions instead of sentences or words. However, content selection is still performed at the sentence level. Second, the proposed node scoring strategy exploits cognitively-grounded properties of human memory structures. We demonstrate through extensive experiments that this scoring strategy outperforms previously proposed systems that model sentence relevancy through centrality.

2.2 Informativeness, Redundancy, and Cohesion

Traditional summarization approaches sought to provide more control over properties of generated summaries such as their informativeness (Jones, 1993; Carbonell & Goldstein, 1998; Nenkova & McKeown, 2011; Lloret, 2012; Teufel, 2016), non-redundancy (Carbonell

& Goldstein, 1998), or discourse organization (Marcu, 1998; Christensen et al., 2013). However, more modern approaches mostly employ neural end-to-end models (Cheng & Lapata, 2016; Lewis et al., 2020; J. Zhang et al., 2020), meaning that crucial intermediate steps such as content planning or selection are not explicitly modeled. Recent efforts have 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), whereas explicit content selection modules can be tailored to tackle problems such as factuality (Cao et al., 2018; Maynez et al., 2020; Z. Zhao et al., 2020), coverage (Kedzie et al., 2018; Puduppully et al., 2019; Wiseman et al., 2017), and redundancy (Y. Liu & Lapata, 2019; Jia et al., 2021; Bi et al., 2021). Specifically, production of low-redundant summaries has proven to be challenging, especially when the source document is highly redundant, such as scientific articles (Xiao & Carenini, 2020; Gu et al., 2022).

Regarding cohesion, Wu and Hu (2018) combined an informativeness reward with a cohesion reward in a reinforcement learning setup, reporting heavy trade-offs between the two properties. In contrast, Xu et al. (2020) reported an improvement in informativeness when incorporating information about the global discourse organization (RST trees) and coreference chains in a supervised setup. However, discourse organization of final summaries experimented only a marginal improvement.

In this work, we demonstrate that it is possible to improve lexical cohesion –a special case of local coherence– while maintaining a high level of informativeness and without selecting overly redundant content, under a reinforcement learning setup. In the unsupervised scenario, our proposed models are able to successfully balance cohesion and redundancy, although sacrificing informativeness in the process.

2.3 Cognitive Models for Summarization

In psycholinguistics, summarization as a task is often used as a method to investigate cognitive processes involved in text comprehension and production (W. Kintsch & van Dijk, 1978; E. Kintsch, 1990; Lehto, 1996; W. Kintsch & Walter Kintsch, 1998; Ushiro et al., 2013; Spirgel & 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 (W. Kintsch & van Dijk, 1978; W. 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 & Teufel, 2014; R. 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 & Teufel, 2014; R. Zhang et al., 2016; Fang, 2019) show a heavy reliance on NLP tools such as entity extractors and coreference resolution systems, as well as external resources like WordNet (Miller, 1992). 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 struc-

tures, modeling retrieval processes, or manipulating information at the right granularity level, e.g. ranking words or sentences instead of semantic propositions.

We address these limitations by introducing two computational implementations of the KvD theory (W. Kintsch & van Dijk, 1978) that require only a dependency parser and no external resources, making it possible to test these systems on other languages and domains. Moreover, our proposed systems better exploit memory structure properties and retrieval processes during reading simulation, which makes them capable of producing notoriously less redundant and more cohesive summaries than strong baselines.

3. Reward-guided Control Scenario

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 ($\mathcal{L}_{\text{CE}}$) over the sequence labeling task outlined above. In the rest of this paper, 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 (Y. Dong et al., 2018), as

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

where ROUGE F_1 scores are calculated using the reference summaries. Next, we define models employing policy gradient methods that maximize a reward function combining r_{I} with redundancy or coherence-aware rewards.

3.1 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 & Paliwal, 1997) with GRU cells (Cho, van Merriënboer, Bahdanau, & Bengio, 2014), i.e. $h_i^g = [f_i, b_i]$, 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 $l_1 = [f_{j+1} - f_{i+1}; b_{i+1} - b_j]$.

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_i^g, and section embedding l_t, where s_i belongs to section t, the final sentence representation z_i is obtained as follows,

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

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

3.2 Informativeness and Redundancy

We adapt MMR-Select+ (Xiao & 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

$$
\begin{aligned}
\mathcal{L} &= \gamma_{\mathrm{R}} \cdot \mathcal{L}_{\mathrm{R}} + (1 - \gamma_{\mathrm{R}}) \cdot \mathcal{L}_{\mathrm{CE}},\\
\mathcal{L}_{\mathrm{R}} &= -(r_{\mathrm{I}}(\hat{S}) - r_{\mathrm{I}}(\bar{S})) \sum_{s_i \in \hat{S}} \log p(y_i \mid s_i)
\end{aligned}
$$

where $r_{\mathrm{I}}(\bar{S})$ 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}

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

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 & Goldstein, 1998) that aims to minimize semantic similarity between sentences in $\hat{S}$. In our experiments, we dub this model as E.LG-MMRSel+.

3.3 Informativeness and Local Coherence

Building upon Wu and Hu (2018), we define a reward that combines informativeness and local coherence, $r = \lambda_{\text{LC}} \cdot r_{\text{I}} + (1 - \lambda_{\text{LC}}) \cdot r_{\text{LC}}$, where λ_{LC} controls the trade-off between informativeness and coherence and r_{LC} is a local coherence scorer. 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.

Local Coherence Scorer. Scorer r_{LC} receives a multi-sentence text and assigns a score between $[0, 1]$ quantifying its local coherence, and it is defined as follows. Following the methodology of Steen and Markert (2022), we train a RoBERTa model (Y. Liu et al., 2019) to distinguish shuffled from unshuffled summaries. The model is trained in a binary classification setup with chunks of 3 consecutive sentences as positive class and their shuffled versions as negative class. Then, the local coherence score of a summary is defined as the positive class probability, averaged over windows of 3 sentences taken with padding of one sentence.

4. Unsupervised Control Scenario

In this section, we formulate the second scenario, i.e. controlling for informativeness, non-redundancy, and cohesion in candidate summaries in an unsupervised setup. Similarly to the first scenario, we formulate the task of extractive summarization as a two-step process, sentence scoring and sentence selection. During sentence scoring, the document is consumed one sentence at a time, updating the score of a subset of sentences at each step. Then, the top-scoring sentences are selected according to a predefined budget.

This section is organized as follows. First, we elaborate on the Micro-Macro Structure theory of reading comprehension, KvD, explain in detail how it simulates short-term memory, and discuss how its operationalization can be leveraged for sentence scoring in extractive summarization. Then, we introduce two novel computational implementations of the KvD theory tailored to sentence scoring.

4.1 The KvD Theory of Human Memory

Proposed by W. 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. In this 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 word 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 and its building is guided by the reader's goals in mind whilst performing the task. For instance, if the task is summarization, KvD defines macro-processes concerned with generalization, fusion, insertion of details from background knowledge, among others. In this work, 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 & Lamb, 2020; Bengio, 2017) and common-sense reasoning (Speer et al., 2017; Bosselut et al., 2019) show a promising development path, we leave this path out of the scope of this work and for future work.

Experimentally, W. Kintsch and van Dijk (1978) tested the model on the tasks of recall and summarization which required human subjects to write down a short text after reading a document. The recall task aimed to measure how accurately subjects can reproduce specific propositions from the given document, whereas the summarization task aimed to quantify how many summary-worthy propositions are retrieved. The KvD theory models the probability of writing down (*to reproduce*) a proposition as a function of the frequency with which it was retained in working memory. The longer a proposition remained in working memory, either at the micro level or at the macro level, the higher its reproduction probability. This probability is then used to makes predictions about what content is more likely to be written down in a summary. Crucial to our work, KvD argued that reproduction probability can be used as a numerical score to rank propositions. In an extractive summarization setup, this score can be used to rank content and select them accordingly. We elaborate on how to define such a score function in detail in Section 4.2, along with our computational implementations of the KvD theory.

4.1.1 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 cohesion represented in the current

tree.[2] 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. 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 pruning, 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 the 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).

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, shown as a squared node in tree (3a) and (3b). Then, the selection strategy is applied and node 11 is selected as new root, obtaining (3b).

After analyzing how trees are shaped in each cycle, it is important to point out their importance for the task of extractive summarization. Next, we elaborate on how memory trees can be leveraged for this end.

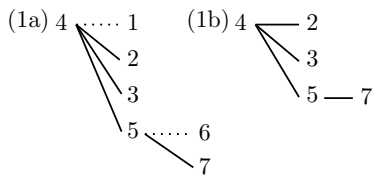
4.1.2 Properties Relevant to Summarization

The procedure for content manipulation described above imposes constraints on the shape, size, and content of memory trees during simulation. Such constraints bestow memory trees

2. It is worth noting that W. 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 2.

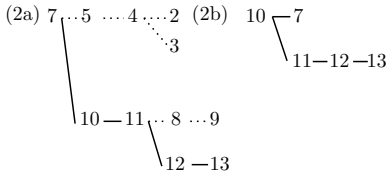
Cycle 1
In healthy people, reactive oxidant species are controlled by a number of enzymatic and non-enzymatic antioxidants.

```
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)
```



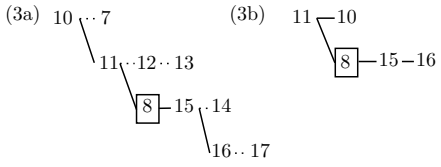
Cycle 2
In patients with Cystic Fibrosis (CF), deficiency of nonenzymatic antioxidants is linked to malabsortion of lipid-soluble vitamins.

```
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: 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.

with special properties relevant to the task of summarization, specifically with respect to 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, W. 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 lexical cohesion. For instance, proposition 8 in cycle 3 of Figure 2 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.

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). 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, 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 decrease redundancy in the final summary and becomes particularly important for highly redundant documents, e.g. scientific articles that repeat information in several sections.

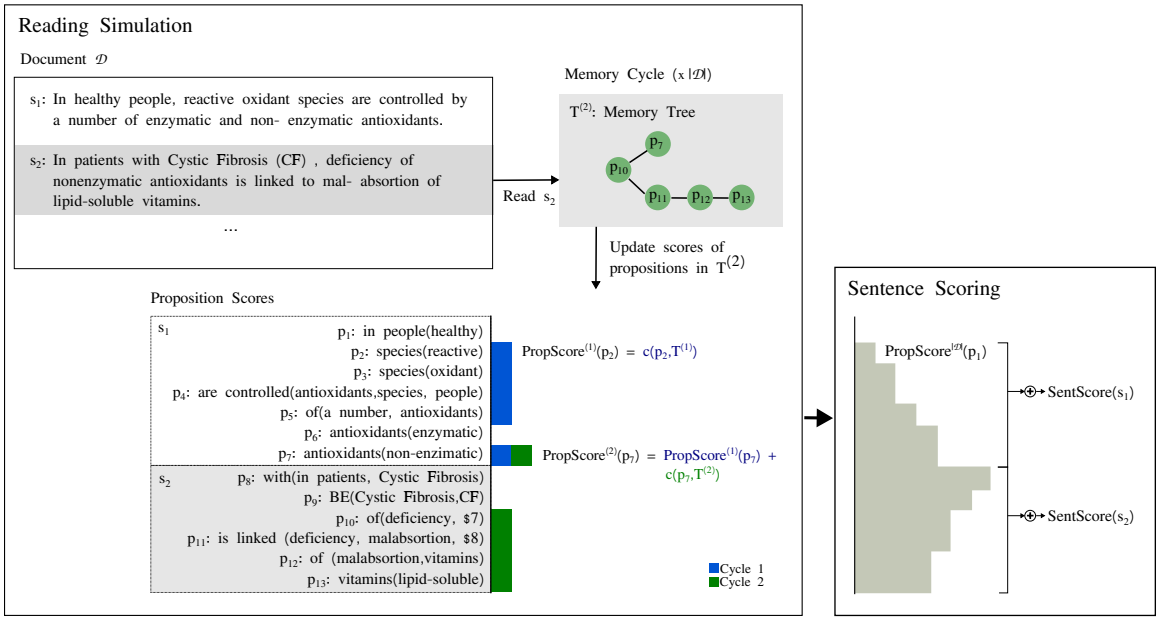


Figure 3: Pipeline of KvD reading simulation and sentence scoring using the simulation example in Fig.2.

4.2 Unsupervised Summarization as Human Memory Simulation

In this part, 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 and discuss the design choices made, and complement the explanation with a simulation example.

4.2.1 Pipeline Overview

The pipeline for sentence scoring is depicted in Figure 3. 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 of propositions populated by nodes demoted from working memory as reading progresses.

The outline of the the simulation procedure is presented in Algorithm 1. The algorithm consumes a document $\mathcal{D} = \langle s_0, \ldots, s_k, \ldots, s_{|\mathcal{D}|} \rangle$ iteratively in memory cycles, updating

Algorithm 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: $\texttt{WM}$, size of working memory
Require: Ψ, maximum tree persistence
1: **procedure** runSimulationKvD($\mathcal{D}, \texttt{WM}, \Psi$)
2: $T \leftarrow \emptyset$ ▷ Memory tree, initially empty
3: $G \leftarrow \emptyset$ ▷ Long-term memory, initially empty
4: $\psi \leftarrow 0$ ▷ Tree persistance counter
5: **for** $s_k \in \mathcal{D}$ **do**
6: $P_k \leftarrow \texttt{getPropositionTree}(s_k)$
7: $T, attached \leftarrow \texttt{attachPropositions}(P_k, T, G)$
8: **if** *attached* **then**
9: $\texttt{adjustRoot}(T)$
10: $\texttt{memorySelect}(\texttt{WM}, T)$
11: $\texttt{updateScore}(T)$
12: $\psi \leftarrow 0$
13: **else**
14: $\psi \leftarrow \psi + 1$
15: **end if**
16: **if** $\psi = \Psi$ **then**
17: $T \leftarrow \emptyset$
18: **end if**
19: **end for**
20: **end procedure**

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

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

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 proposed by W. Kintsch and van Dijk (1978) by incrementally scoring propositions based on how often they appeared in memory trees and in which part of said trees they were attached. Then, the 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 complete 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

$$\text{SentScore}(s_k) = \sum_{p \in V[P_k]} \text{PropScore}(p), \tag{2}$$

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. 2 until the budget of `B` tokens is met.

4.2.2 Proposition Building

Propositions are obtained by recursively merging and rearranging nodes in dependency trees, extending the procedure of Fang (2019). Given sentence $s = \langle w_0, w_1, ..., w_N \rangle$ and its corresponding dependency tree Q with nodes $\{q_0, .., q_N\}$,[3] the objective is to obtain proposition tree P with nodes $\{p_0, ..., p_M\}$, $M \leq N$, 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 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. dep(u, v) = goeswith).

Consider the example in Figure 4. Starting from dependency tree Q (Fig. 4a), 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.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.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 $v_i \in$ 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.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.d points to propositions `2` and `3` and hence, they are connected in P.

Under this procedure, the 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.

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

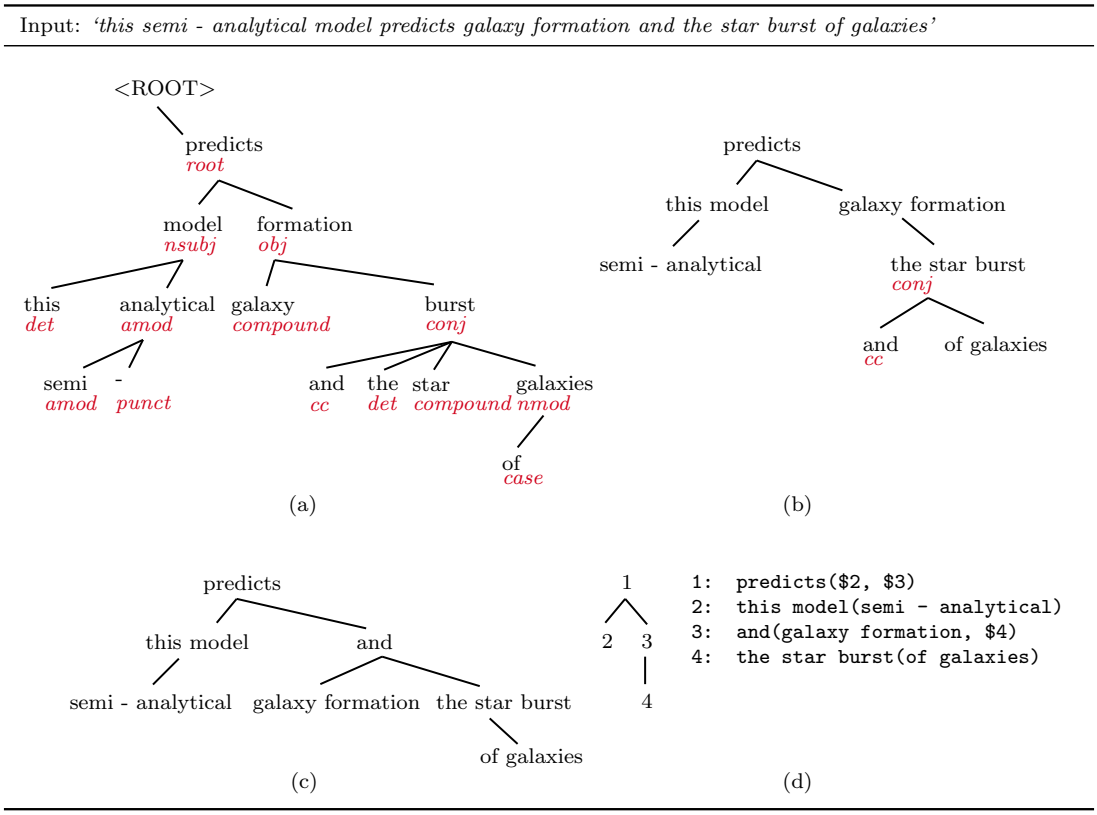


Figure 4: Step-by-step construction of proposition tree from an input sentence, starting from obtaining its dependency tree in UD format (a), merging dependent nodes into head nodes (b), promoting coordinating conjunctions to head status (c), to finally build propositions from non-leaf nodes (d).

Proposition Overlap. We connect propositions from different sentences by quantifying the lexical overlap between their functors –predicates and arguments. Let functors(p) be the set of the functors –predicate and arguments– in propositions p. Given $p_1 \in P_x$ and $p_2 \in P_y$, let $A^*(p_1, p_2)$ 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

$$\phi(p_1, p_2) = \frac{1}{|A^*|} \sum_{\langle a_1, a_2 \rangle \in A^*} \text{jaccard}(a_1, a_2). \tag{3}$$

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.

4.2.3 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 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 4.2.2.

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

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

where $\phi(\cdot)$ is the proposition overlap function defined in Equation 3. In case 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 $\mathbf{f} = \langle f_1, ..., f_n \rangle, f_i \in V[G] \wedge n \leq R$. 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 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 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 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 G.

Proposition Scoring. Following Eq. 1, 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

$$c(t, T) = \frac{|T_t|}{|T|} \exp\left(\frac{1}{\text{depth}(t)}\right), \tag{4}$$

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 W. 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, W. 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 & Phillips, 1965). In system TreeKvD, we limit ourselves to recall previously read propositions verbatim and further limit 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 the 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.

4.2.4 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 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 defined as the pair (p, t) such that $t = \text{argmax}_{\hat{t} \in V[T]} \phi(\hat{t}, p)$, where $\phi(\cdot)$ is again the proposition overlap function defined in Equation 3. 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 $\mathbf{f}^* = \langle f_1, ..., f_n \rangle$, $f_i \in V[G] \wedge n \leq R$, 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, Jonides, & Lewis, 2009), an idea also applied in the optimization of neural networks (Loshchilov & 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 (W. 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.[4]

This particular retrieval order follows *free recall* accuracy in human subjects (Glanzer, 1972).[5] 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 the maximum closeness score is chosen as the new root. Then, T is pruned down to have at most `WM` nodes using the same strategy as in § 4.2.3.

Proposition Scoring. The score of nodes in working memory T is updated according to Eq. 1 and Eq.4. 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

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

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.

4.2.5 Simulation Example

Next, we illustrate the procedures outlined in Algorithm 1 with an example, showcased in Figure 5. The example takes two sentences from a scientific article and simulates two memory cycles with TreeKvD (left) and GraphKvD (right). The propositions involved

4. Experimentally, increasing this threshold does not impact downstream performance significantly.
5. 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.

(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 shown in italics. First, we analyze the processes involved during attachment in a memory cycle, including how the 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 5. 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).

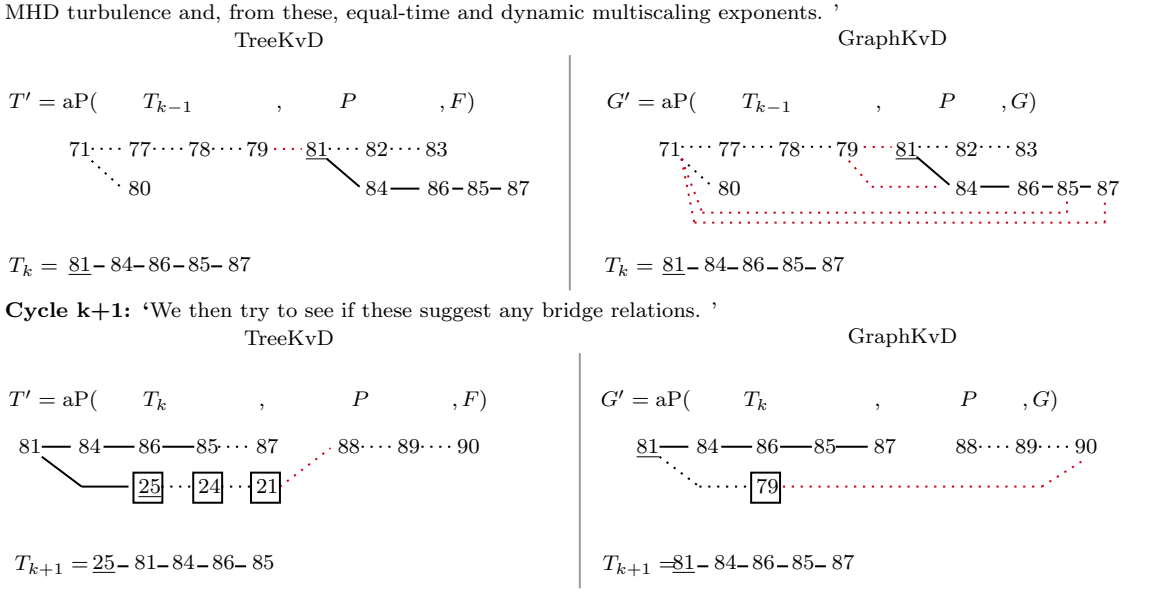
Properties of Memory Trees. Properties of memory structures at the micro level, as discussed in Section 4.1.2, have the potential to greatly influence the level of lexical cohesion and redundancy in output summaries, in addition to identifying relevant content to be included. We now elaborate on how 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 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 4.1.2, 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



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 5: Simulation example of TreeKvD (left) and GraphKvD (right). Each memory cycle shows the input sentence, extracted propositions, and the derived memory tree. Function aP refers to subroutine `attachPropositions` in Algorithm 1. 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 memory tree. Relevant content common in propositions and the gold summary is coloured in blue.

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 5.

5. 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 at quantifying 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.

5.1 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. Instances with documents with less than 5 tokens in the abstract are ignored. Sentences are capped at 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 & Carenini, 2020; Gu et al., 2022), we use a budget of B= 200 tokens for both arXiv and PubMed.

5.2 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 of greedily selecting a set of sentences that maximize the sum of ROUGE-1 and ROUGE-2 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 the performance of E.LG as a reference for an informativeness-oriented baseline, and use checkpoints provided by (Xiao & Carenini, 2020). For redundancy-oriented model E.LG-MMRSel+, we use the default hyper-parameter configuration (Xiao & 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 $\lambda_{\text{LC}} = 0.2$. Both models were trained using Adam optimizer (Loshchilov & 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 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, Cohen, & Lapata, 2018; Y. 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, 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 paper.

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, the decay factor is set to $\beta = 0.01$.

We compare against unsupervised systems that model a document as a graph of sentences and employ node centrality as a proxy for informativeness. First, we report on TextRank (Mihalcea & Tarau, 2004),[6] a system that employs TF-IDF as edge score between sentences and the PageRank algorithm (Brin & Page, 1998) to obtain node centrality. Second, we benchmark PacSum (Zheng & Lapata, 2019), which learns a specialized edge scorer and also uses PageRank. For computational purposes, we limit connection to sentences in a window of size 200.[7] We report results using a SciBERT (Beltagy, Lo, & Cohan, 2019) sentence embedded and two configurations: PacSum, using the default hyper-parameters reported by Zheng and Lapata (2019), and PacSum-FT*, finetuned over a sample of 1000 documents following the procedure therein.

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. 1 except subroutine `memorySelect`. Similarly to PacSum, proposition connection is limited to those in the previous 50 sentences. Finally, we compared our proposed models against a previous implementation of the KvD theory (Fang, 2019), labeled as FangKvD.

5.3 Automatic Evaluation

We evaluate the intrinsic performance of the analyzed models in terms of informativeness, redundancy, and lexical cohesion.

6. We use implementation in the Gensim library (Rehurek & Sojka, 2010).
7. Such a limitation was possibly not considered by Zheng and Lapata (2019) since their model was not designed for long documents, it was tested on the CNN/DM dataset in which documents are 50 sentences long in average.

5.3.1 Informativeness

We report F_1 ROUGE (Lin, 2004) which measures lexical –n-gram– overlap between extracted summaries and reference summaries, serving as an indicator for informativeness and relevancy. Even though many issues have been identified when using ROUGE outside its proposed setting (F. Liu & Liu, 2008; Cohan & Goharian, 2016), many variations of the original metric have shown a strong correlation with human assessment (Graham, 2015; ShafieiBavani et al., 2018; Fabbri et al., 2021).

Nevertheless, ROUGE is not designed to appropriately reward semantic and syntactic variation in summaries. For this reason, the semantic relevancy of summaries is assessed using F_1 BertScore (T. Zhang et al., 2020) which addresses semantic similarity by comparing contextual embeddings given by a pretrained BERT model. 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 (Y. Liu et al., 2019) as underlying model, and apply importance weighting to diminish the effect of non-content words, e.g. function words.[8]

5.3.2 Redundancy

We assess redundancy in a text with the following metrics, each of which computes a value in the range of $[0; 1]$, the higher it is the more redundant a text will be.

Inverse Uniqueness (IUniq). Defined as $\text{IUniq} = 1 - \text{Uniq}$, where Uniq refers to *uniqueness* (Peyrard, Botschen, & Gurevych, 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 & Cardie, 2020). Given candidate summary $\hat{S}$,

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

5.3.3 Cohesion

The following measures of cohesion are used.

Extended Entity Grid (EEG). The Entity Grid (Barzilay & Lapata, 2005) models cohesion in a text 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, the number of modifiers, among others Elsner and Charniak (2011). We use the implementation part of the Brown Coherence Toolkit[9] and train our models over 50 000 uniformly chosen samples from each training set.

8. IDF statistics were obtained from documents in the training set of each dataset.
9. **https://web.archive.org/web/20200505174052/https://bitbucket.org/melsner/browncoherence**

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

5.3.4 Local Coherence

The local coherence of a summary is assessed using the CCL scorer defined in § 3.2 with a sentence window of 3 and padding of 1.

5.3.5 Metric Reliability

The automatic metrics for cohesion and local coherence used in this article present the following limitations that might impact their reliability. Regarding metrics of cohesion, their reliability depends on the accuracy of noun extraction. EEG employs a co-reference resolution tool (Ng & 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 & Markert, 2022). In this article, 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.

5.4 Human Evaluation

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

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. In order to ensure the quality of annotations, 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. 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.[10] Annotations that failed the controls were discarded in order to maximize the quality. 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 (Fabbri et al., 2021; Wu & Hu, 2018). In each system pair

10. Time threshold obtained from pilot study measurements.

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 & Hirst, 1991), and the distribution of these chains across a text has been shown to be a strong indicator of cohesion (Barzilay & Elhadad, 1997; Galley & 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 (S. Zhang & Bansal, 2021), we employ propositions –as extracted in Section 4.2.2– 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 6. Then, the task consists of selecting chains of colored text chunks that share 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[11]; 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 (Halliday & Hasan, 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 through 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 $a \in C_A$

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

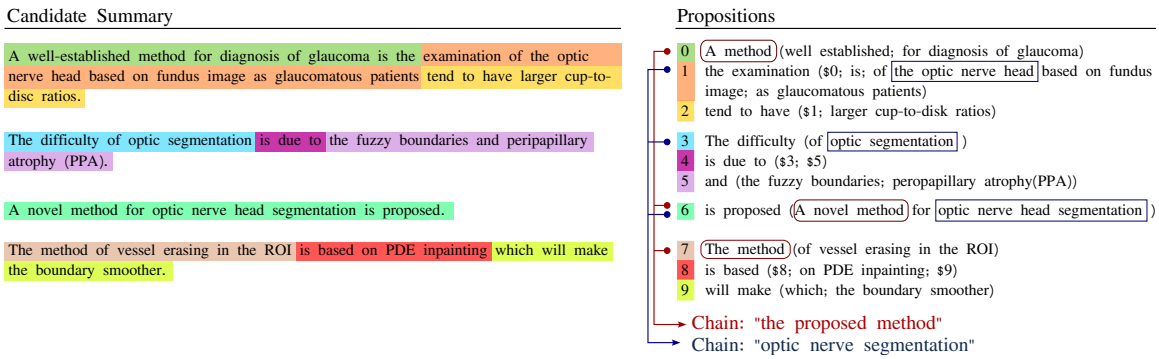


Figure 6: Example of proposition chain annotation in our cohesion evaluation campaign. Each coloured chunk in the candidate summary corresponds to a pre-extracted proposition. Users are tasked to group text chunks that share information by clicking on them. Best seen in colour.

and $b \in C_B$, we define Precision, Recall, and F_1 score as follows,

$$\mathrm{P}^{ov}(a,b) = \frac{\sum_{p\in a} \max_{q\in b} |\mathrm{LCS}(p,q)|}{\sum_{p\in a} |p|},$$
$$\mathrm{R}^{ov}(a,b) = \frac{\sum_{q\in b} \max_{p\in a} |\mathrm{LCS}(p,q)|}{\sum_{q\in b} |q|},$$
$$\mathrm{F}_1^{ov}(a,b) = \frac{2\cdot \mathrm{P}^{ov}\cdot \mathrm{R}^{ov}}{\mathrm{P}^{ov}+\mathrm{R}^{ov}},$$

where p and q are propositions included in chains a and b, respectively, $\mathrm{LCS}(p,q)$ 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

$$\mathrm{ChainOverlap}(A,B) = \frac{1}{|C_A|\cdot|C_B|} \sum_{a\in C_A, b\in C_B} \mathrm{F}_1^{ov}(a,b).$$

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

6. 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 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. In all our experiments, statistical significance at the 95% confidence level is estimated using bootstrap resampling (Davison & Hinkley, 1997).

Aim	System	PubMed R1	PubMed R2	PubMed RL	PubMed BSc	arXiv R1	arXiv R2	arXiv RL	arXiv 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 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.

6.1 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 1, whereas results on redundancy, cohesion, and local coherence metrics are presented in Table 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 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 & 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 1 and 2 indicate system groups in which no pairwise difference was found, one group per marker, for each metric reported.

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

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 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.

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 insights 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 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 into 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 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.

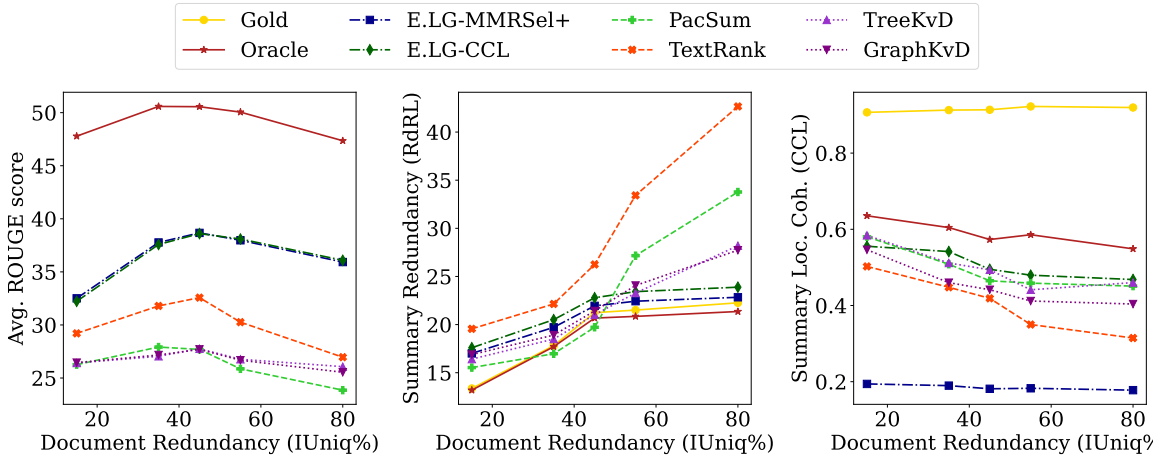
Lastly, it is important to point out the limited expressivity of the EEG metric (i.e. score gap at the system level) and the difference between trends in cohesion metrics and trends in CCL. As mentioned in § 5.3.5, EEG is limited by data sparsity —the limited lexical matching between nouns and entities– and the performance of coreference resolution tools they use. Hence, its expressivity is highly dependent on the accuracy of noun detection. Regarding metric trends, metrics EEG and EGr were designed to capture lexical and semantic links between sentences in a text, therefore measuring cohesion. While cohesion is considered a device to achieve local coherence, it does not model discourse structure. In contrast, CCL was trained to capture sensible sentence orderings as a proxy for discourse organization on nearby sentences. As such, it is capable of capturing not only lexical cohesive ties but also rhetorical orderings in a text.

6.2 Effect of Document Redundancy

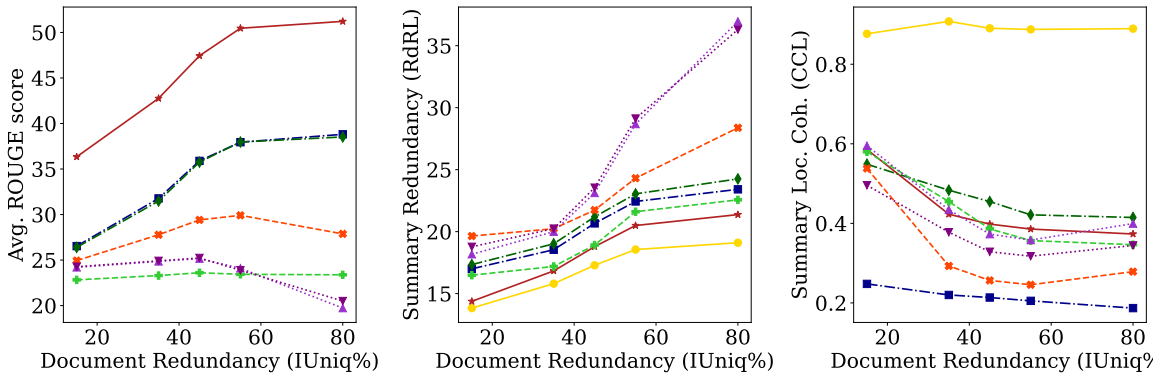
Next, we take a closer look at the redundancy and cohesion levels in summaries extracted from increasingly redundant documents. Figure 7 shows the performance of summarization systems in terms of informativeness (average ROUGE score, (ROUGE-1 + ROUGE-2 + ROUGE-L)/3), redundancy (RdRL), and local coherence (CCL) across different levels of document redundancy (IUniq). Test sets were divided into bins according to their document redundancy score and the average metric value per bin is reported. For simplicity, we only plot the 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 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.



(a) PubMed



(b) arXiv

Figure 7: 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.

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.[12]

12. 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.

6.3 Human Evaluation

The results of our human evaluation campaigns are showcased in Table 3. In both studies, statistical significance between system scores was assessed by making pairwise comparisons between all systems using a one-way ANOVA 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, n.d.)– was found to be 0.73.

We found that humans shown significantly higher preference for TreeKvD summaries compared to PacSum summaries ($p < 0.01$), 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 5.4, was calculated at 0.97. Score differences between system pairs TreeKvD–PacSum and GraphKvD–PacSum were found to be statistically significant, for all the analyzed 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 on 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 through another chain. On the other hand, we note that gold summaries show lower chain density than GraphKvD summaries on average. We hypothesize that the lower chain density in gold summaries is due to the high technicality 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.

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

6.4 Qualitative Analysis

We performed a qualitative analysis of system summaries extracted by the compared systems (Figure 8 and 9) by annotating the lexical chains in them and analyzing 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 8, 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 span windows of three to four sentences at a time. Note that chains spanning more than one sentence imply a non-trivial level of redundancy, as shown by RdRL>0. These smooth transitions are detected by our 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 shown 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 a 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 9, 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 lexical 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 steers 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 confirm 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 cohesion 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 cohesion, 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.

6.5 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 persistence) 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

System	Avg. ROUGE	RdRL	CCL
Gold Summary	-	12.6	0.86
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.			
E.LG-MMRSel+	31.60	16.13	0.20
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.			
E.LG-CCL	31.82	18.90	0.60
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.			
MemSum	34.16	22.34	0.33
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.			

Figure 8: 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.

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 shown in Figure 10. 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.

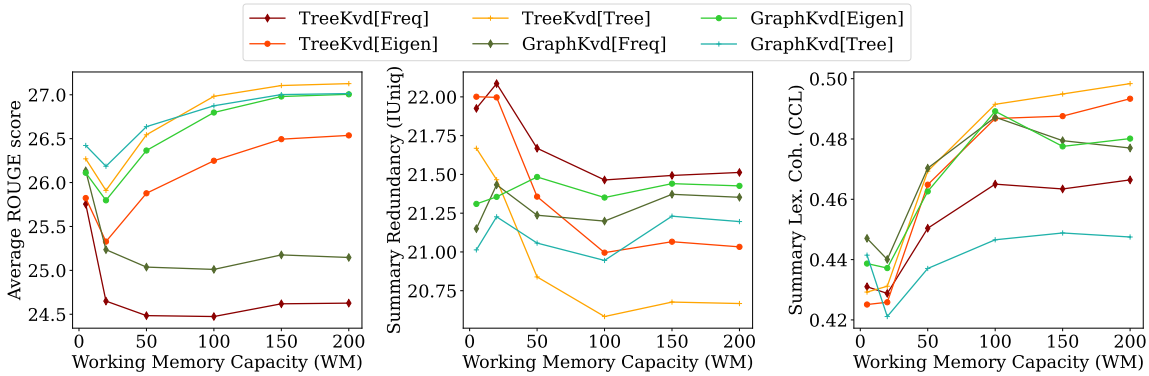
Moreover, as seen in Table 1, KvD systems with WM = 100 obtain consistently higher relevancy scores than FullGraph, a system that does not simulate working memory and which scoring strategy has access to all the propositions in a document at all times. This

System	Avg. ROUGE	RdRL	CCL
Gold Summary	-	21.49	0.91
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.			
TextRank	38.99	45.02	0.23
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.			
TreeKvD	39.87	14.62	0.36
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.			
GraphKvD	39.73	21.65	0.51
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.			

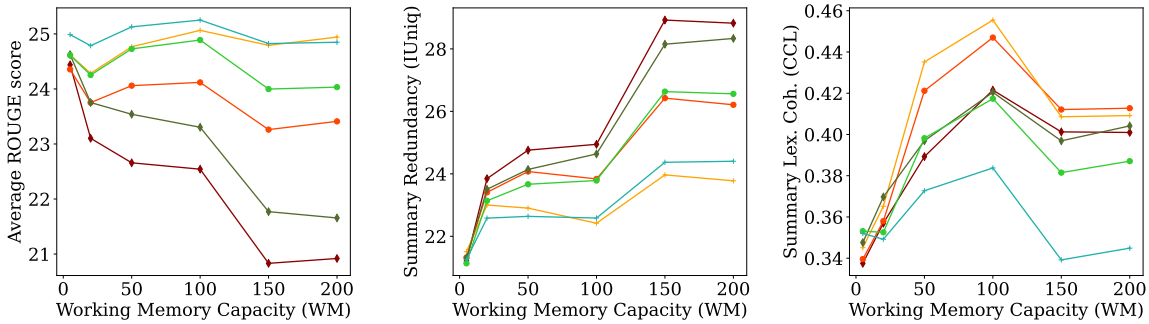
Figure 9: Summaries extracted by unsupervised systems for an arXiv sample with informativeness (average ROUGE score), redundancy (RdRL), and local coherence (CCL) scores. Text is annotated with color-coded lexical chains, and was detokenized and truecased for ease of reading.

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 4.2.3, 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 a 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



(a) PubMed



(b) arXiv

Figure 10: 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).

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 WM that yields the highest sum of ROUGE-1 and ROUGE-2 scores. The results are encouraging: when using the best possible 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 WM = 100 for all samples. Most surprisingly, the distribution of best WM per sample is rather balanced, with 26.5% of samples preferring a WM = 100, 26.7% a WM = 50, 24.11% a WM = 20, and 22.5% a 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, 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:

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

where λ is the largest eigen-value of the adjacency matrix of T.[13]

Figure 10 shows the performance of our KvD systems over the validation set of PubMed and arXiv. Systems using scoring function $c(t, T)$ in Eq. 1 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 (W. Kintsch & 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 `WM` $= 100$ and Tree scoring, and set the maximum allowed 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 indicates that our implementations of the KvD

13. 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: Effect of using lexical overlap and semantic similarity in argument overlap calculation, as measured by ROUGE F_1 scores, redundancy (IUniq), and local coherence (CCL), over the validation sets of PubMed and arXiv.

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 10. 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 $\texttt{WM} = 150$, 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. 3 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 & Strube, 2020, 2022) shown the advantage of using XLNet against other Transformer-based architectures when modeling local coherence in contexts a few sentences long.[14]

Table 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).

14. Indeed, preliminary experiments using SciBERT (Beltagy et al., 2019) shown poor results.

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 the 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.

7. Conclusions

In this paper, 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 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 how information is discretized into semantic propositions and organized in human working memory, according to the Micro-Macro Structure theory of reading comprehension. Extensive quantitative and qualitative analysis shown 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.

Acknowledgements

We thank the reviewers for their detailed feedback and Mausam and Julia Hockenmaier for handling the paper as associate editors. We also appreciate feedback from members of the Cohort, Ivan Titov, Arman Cohan, and Mark Steedman. We are grateful for a grant from NAVER Labs Europe, which funded this project. We appreciate the computing resources provided by the University of Birmingham and EPCC at the University of Edinburgh.

References

Amplayo, R. K., Angelidis, S., & Lapata, M. (2021). Aspect-Controllable Opinion Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 6578–6593). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Baldwin, B., & Morton, T. S. (1998). Dynamic coreference-based summarization. In *Proceedings of the Third Conference on Empirical Methods for Natural Language Processing* (pp. 1–6). Granada, Spain: Association for Computational Linguistics.

Barzilay, R., & Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In *Intelligent Scalable Text Summarization.*

Barzilay, R., & Lapata, M. (2005). Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 141–148). Michigan, USA: Association for Computational Linguistics.

Beltagy, I., Lo, K., & Cohan, A. (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 Processing (EMNLP-IJCNLP)* (pp. 3615–3620). Hong Kong, China: Association for Computational Linguistics.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. *ArXiv preprint*, *abs/2004.05150*.

Bengio, Y. (2017). The consciousness prior. *ArXiv preprint*, *abs/1709.08568*.

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

Bi, K., Jha, R., Croft, B., & Celikyilmaz, A. (2021). AREDSUM: Adaptive Redundancy-Aware Iterative Sentence Ranking for Extractive Document Summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 281–291). Online: Association for Computational Linguistics.

Bichi, A. A., Samsudin, R., Hassan, R., & Almekhlafi, K. (2021). A Review of Graph-Based Extractive Text Summarization Models. In F. Saeed, F. Mohammed, & A. Al-Nahari (Eds.), *Innovative Systems for Intelligent Health Informatics* (pp. 439–448). Cham: Springer International Publishing.

Bommasani, R., & Cardie, C. (2020). Intrinsic Evaluation of Summarization Datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP) (pp. 8075–8096). Online: Association for Computational Linguistics.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4762–4779). Florence, Italy: Association for Computational Linguistics.

Brin, S., & Page, L. (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer networks and ISDN systems*, *30*(1-7), 107–117.

Cao, Z., Wei, F., Li, W., & Li, S. (2018). Faithful to the Original: Fact Aware Neural Abstractive Summarization. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)* (pp. 4784–4791). AAAI Press.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 335–336).

Cheng, J., & Lapata, M. (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)* (pp. 484–494). Berlin, Germany: Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (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* (pp. 103–111).

Christensen, J., Mausam, Soderland, S., & Etzioni, O. (2013). Towards Coherent Multi-Document Summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1163–1173). Atlanta, Georgia: Association for Computational Linguistics.

Clarke, J., & Lapata, M. (2010). Discourse Constraints for Document Compression. *Computational Linguistics*, *36*(3), 411–441.

Cohan, A., Beltagy, I., King, D., Dalvi, B., & Weld, D. (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)* (pp. 3693–3699). Hong Kong, China: Association for Computational Linguistics.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (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)* (pp. 615–621). New Orleans, Louisiana: Association for Computational Linguistics.

Cohan, A., & Goharian, N. (2016). Revisiting Summarization Evaluation for Scientific Articles. In *Proceedings of the Tenth International Conference on Language Resources*

and Evaluation (LREC'16) (pp. 806–813). Portorož, Slovenia: European Language Resources Association (ELRA).

Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap methods and their application* (No. 1). Cambridge University Press.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., ... Hon, H. (2019). Unified Language Model Pre-training for Natural Language Understanding and Generation. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 2019* (pp. 13042–13054).

Dong, Y., Shen, Y., Crawford, E., van Hoof, H., & Cheung, J. C. K. (2018). BanditSum: Extractive Summarization as a Contextual Bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3739–3748). Brussels, Belgium: Association for Computational Linguistics.

Elsner, M., & Charniak, E. (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* (pp. 125–129). Portland, Oregon, USA: Association for Computational Linguistics.

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. (2021). Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, *9*, 391–409.

Fang, Y. (2019). *Proposition-based Summarization with a Coherence-driven Incremental Model* (Unpublished doctoral dissertation). University of Cambridge.

Fang, Y., & Teufel, S. (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* (pp. 732–741). Gothenburg, Sweden: Association for Computational Linguistics.

Fonseca, M., Ziser, Y., & Cohen, S. B. (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* (pp. 6341–6364). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Galley, M., & McKeown, K. (2003). Improving word sense disambiguation in lexical chaining. In *Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1486–1488).

Garcez, A. d., & Lamb, L. C. (2020). Neurosymbolic AI: The 3rd Wave. *ArXiv preprint*, *abs/2012.05876*.

Garrod, S., & Sanford, A. (1977). Interpreting anaphoric relations: The integration of semantic information while reading. *Journal of Verbal Learning and Verbal Behavior*, *16*(1), 77–90.

Garrod, S. C., & Sanford, A. J. (1994). Resolving sentences in a discourse context: How discourse representation affects language understanding. In *Handbook of Psycholinguistics.* (pp. 675–698). San Diego, CA, US: Academic Press.

Glanzer, M. (1972). Storage mechanisms in recall. In *Psychology of learning and motivation* (Vol. 5, pp. 129–193). Elsevier.

Goldfarb-Tarrant, S., Chakrabarty, T., Weischedel, R., & Peng, N. (2020). Content Planning for Neural Story Generation with Aristotelian Rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4319–4338). Online: Association for Computational Linguistics.

Gong, Y., & Liu, X. (2001). Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 19–25).

Goyal, T., Li, J. J., & Durrett, G. (2022). SNaC: Coherence Error Detection for Narrative Summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 444–463). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Graham, Y. (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* (pp. 128–137). Lisbon, Portugal: Association for Computational Linguistics.

Gu, N., Ash, E., & Hahnloser, R. (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)* (pp. 6507–6522). Dublin, Ireland: Association for Computational Linguistics.

Guinaudeau, C., & Strube, M. (2013). Graph-based Local Coherence Modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (volume 1: Long papers)* (pp. 93–103).

Hachey, B., Murray, G., & Reitter, D. (2006). Dimensionality Reduction Aids Term Co-Occurrence Based Multi-Document Summarization. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering* (pp. 1–7). Sydney, Australia: Association for Computational Linguistics.

Halliday, M., & Hasan, R. (1976). *Cohesion in English* (No. 9). Routledge.

Harley, T. A. (1995). *The Psychology of Language: From Data to Theory.* Erlbaum (Uk) Taylor & Francis, Publ.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching Machines to Read and Comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015* (pp. 1693–1701). Quebec, Canada.

Hossain, T., Logan IV, R. L., Ugarte, A., Matsubara, Y., Young, S., & Singh, S. (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.

Hua, X., Sreevatsa, A., & Wang, L. (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)* (pp. 6408–6423). Online: Association for Computational Linguistics.

Huang, L., Cao, S., Parulian, N., Ji, H., & Wang, L. (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* (pp. 1419–1436). Online: Association for Computational Linguistics.

Jeon, S., & Strube, M. (2020). Centering-based Neural Coherence Modeling with Hierarchical Discourse Segments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 7458–7472). Online: Association for Computational Linguistics.

Jeon, S., & Strube, M. (2022). Entity-based Neural Local Coherence Modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (volume 1: Long papers)* (pp. 7787–7805). Dublin, Ireland: Association for Computational Linguistics.

Jia, R., Cao, Y., Fang, F., Zhou, Y., Fang, Z., Liu, Y., & Wang, S. (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* (pp. 366–376). Online: Association for Computational Linguistics.

Jones, K. S. (1993). What might be in a summary? *Information Retrieval*, *93*(1), 9–26.

Jones, K. S. (2007). Automatic summarising: The state of the art. *Information Processing and Management*, *6*(43), 1449–1481.

Kedzie, C., McKeown, K., & Daumé III, H. (2018). Content Selection in Deep Learning Models of Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 1818–1828). Brussels, Belgium: Association for Computational Linguistics.

Kintsch, E. (1990). Macroprocesses and Microprocesses in the Development of Summarization Skill. *Cognition and Instruction*, *7*(3), 161–195.

Kintsch, W. (1988). The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological review*, *95*(2), 163.

Kintsch, W., & van Dijk, T. A. (1978). Toward a Model of Text Comprehension and Production. *Psychological review*, *85*(5), 363.

Kintsch, W., & Walter Kintsch, C. (1998). *Comprehension: A paradigm for cognition*. Cambridge university press.

Krippendorff, K. (n.d.). Computing Krippendorff's Alpha-Reliability. *Computing*, *1*, 25–2011.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A Trainable Document Summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 68–73).

Lehto, J. (1996). Working Memory Capacity and Summarizing Skills in Ninth-graders. *Scandinavian Journal of Psychology*, *37*(1), 84–92.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871–7880). Online: Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text*

Summarization Branches Out (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics.

Litvak, M., & Vanetik, N. (2017). Query-based Summarization using MDL Principle. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres* (pp. 22–31). Valencia, Spain: Association for Computational Linguistics.

Liu, F., & Liu, Y. (2008). Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries. In *Proceedings of ACL-08: HLT, short papers* (pp. 201–204). Columbus, Ohio: Association for Computational Linguistics.

Liu, Y., & Lapata, M. (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)* (pp. 3730–3740). Hong Kong, China: Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv preprint*, *abs/1907.11692*.

Lloret, E. (2012). Text summarisation based on Human Language Technologies and its applications. *Procesamiento del lenguaje natural*(48), 119–122.

Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019.* New Orleans, USA.

Louis, A., Joshi, A., & Nenkova, A. (2010). Discourse Indicators for Content Selection in Summarization. In *Proceedings of the SIGDIAL 2010 Conference* (pp. 147–156). Tokyo, Japan: Association for Computational Linguistics.

Marcu, D. (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* (pp. 1–8).

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1906–1919). Online: Association for Computational Linguistics.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (pp. 404–411). Barcelona, Spain: Association for Computational Linguistics.

Miller, G. A. (1992). WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop.* New York, USA.

Miura, Y., Zhang, Y., Tsai, E., Langlotz, C., & Jurafsky, D. (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* (pp. 5288–5304). Online: Association for Computational Linguistics.

Moon, H. C., Mohiuddin, T., Joty, S., & Xu, C. (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) (pp. 2262–2272). Hong Kong, China: Association for Computational Linguistics.

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

Narayan, S., Cardenas, R., Papasarantopoulos, N., Cohen, S. B., Lapata, M., Yu, J., & Chang, Y. (2018). 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)* (pp. 2020–2030). Melbourne, Australia: Association for Computational Linguistics.

Narayan, S., Cohen, S. B., & Lapata, M. (2018). 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)* (pp. 1747–1759). New Orleans, Louisiana: Association for Computational Linguistics.

Narayan, S., Cohen, S. B., & Lapata, M. (2019). What is this article about? Extreme summarization with topic-aware convolutional neural networks. *Journal of Artificial Intelligence Research*, *66*, 243–278.

Narayan, S., Maynez, J., Adamek, J., Pighin, D., Bratanic, B., & McDonald, R. (2020). Stepwise Extractive Summarization and Planning with Structured Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4143–4159). Online: Association for Computational Linguistics.

Nenkova, A., & McKeown, K. (2011). Automatic Summarization. *Foundations and Trends® in Information Retrieval*, *5*(2–3), 103–233.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A Compositional Context Sensitive Multi-Document Summarizer: Exploring the Factors that Influence Summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 573–580).

Ng, V., & Cardie, C. (2002). Improving Machine Learning Approaches to Coreference Resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 104–111).

Nivre, J., Zeman, D., Ginter, F., & Tyers, F. (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.

Ono, K., Sumita, K., & Miike, S. (1994). Abstract Generation Based on Rhetorical Structure Extraction. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics.* Kyoto, Japan.

Paulus, R., Xiong, C., & Socher, R. (2018). A Deep Reinforced Model for Abstractive Summarization. In *6th International Conference on Learning Representations (ICLR 2018).*

Perez-Beltrachini, L., Liu, Y., & Lapata, M. (2019). Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5107–5116). Florence, Italy: Association for Computational Linguistics.

Peyrard, M., Botschen, T., & Gurevych, I. (2017). Learning to Score System Summaries for Better Content Selection Evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization* (pp. 74–84). Copenhagen, Denmark: Association for Computational Linguistics.

Postman, L., & Phillips, L. W. (1965). Short-term temporal changes in free recall. *Quarterly journal of experimental psychology*, *17*(2), 132–138.

Puduppully, R., Dong, L., & Lapata, M. (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* (pp. 6908–6915). AAAI Press.

Qian, Y., Muaz, U., Zhang, B., & Hyun, J. W. (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* (pp. 223–228). Florence, Italy: Association for Computational Linguistics.

Qiu, Y., & Cohen, S. B. (2022). Abstractive Summarization Guided by Latent Hierarchical Document Structure. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 5303–5317). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., ... Zhang, Z. (2004). MEAD - A Platform for Multi-Document Multilingual Text Summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04).* Lisbon, Portugal: European Language Resources Association (ELRA).

Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.*

Schuster, M., & Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

ShafieiBavani, E., Ebrahimi, M., Wong, R., & Chen, F. (2018). A Graph-theoretic Summary Evaluation for ROUGE. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 762–767). Brussels, Belgium: Association for Computational Linguistics.

Shapira, O., Ronen, H., Adler, M., Amsterdamer, Y., Bar-Ilan, J., & Dagan, I. (2017). Interactive Abstractive Summarization for Event News Tweets. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System demonstrations* (pp. 109–114). Copenhagen, Denmark: Association for Computational Linguistics.

Sharma, E., Li, C., & Wang, L. (2019). BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2204–2213). Florence, Italy: Association for Computational Linguistics.

Silber, H. G., & McCoy, K. F. (2002). Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. *Computational Linguistics*, *28*(4), 487–496.

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. (2019). MASS: Masked Sequence to Sequence Pre-training for Language Generation. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019* (Vol. 97, pp. 5926–5936). California, USA: PMLR.

Speer, R., Chin, J., & Havasi, C. (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In S. P. Singh & S. Markovitch (Eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 4444–4451). California, USA: AAAI Press.

Spirgel, A. S., & Delaney, P. F. (2016). Does writing summaries improve memory for text? *Educational Psychology Review*, *28*(1), 171–196.

Steen, J., & Markert, K. (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* (pp. 6035–6049). Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Steinberger, J., Poesio, M., Kabadjov, M. A., & Ježek, K. (2007). Two uses of anaphora resolution in summarization. *Information Processing & Management*, *43*(6), 1663–1680.

Teufel, S. (2016). Deeper Summarisation: The Second Time Around. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 581–598).

Ushiro, Y., Takaki, S., Kobayashi, M., Hasegawa, Y., Nahatame, S., Hamada, A., & Kimura, Y. (2013). Measures of Macroproposition Construction in EFL reading: Summary writing task vs. the Meaning Identification Technique. *JLTA Journal*, *16*, 185–204.

Vanderwende, L., Suzuki, H., Brockett, C., & Nenkova, A. (2007). Beyond SumBasic: Task-focused Summarization with Sentence Simplification and Lexical Expansion. *Information Processing & Management*, *43*(6), 1606–1618.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is All you Need. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017* (pp. 5998–6008). California, USA.

Völske, M., Potthast, M., Syed, S., & Stein, B. (2017). TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization* (pp. 59–63). Copenhagen, Denmark: Association for Computational Linguistics.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. *ArXiv preprint*, *abs/2006.04768*.

Williams, R. J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, *8*, 229–256.

Wiseman, S., Shieber, S., & Rush, A. (2017). Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2253–2263). Copenhagen, Denmark: Association for Computational Linguistics.

Wolf, F., & Gibson, E. (2004). Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *Proceedings*

of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04) (pp. 383–390). Barcelona, Spain.

Wu, Y., & Hu, B. (2018). Learning to Extract Coherent Summary via Deep Reinforcement Learning. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18* (pp. 5602–5609). AAAI Press.

Xiao, W., & Carenini, G. (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)* (pp. 3011–3021). Hong Kong, China: Association for Computational Linguistics.

Xiao, W., & Carenini, G. (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* (pp. 516–528). Suzhou, China: Association for Computational Linguistics.

Xu, J., Gan, Z., Cheng, Y., & Liu, J. (2020). Discourse-Aware Neural Extractive Text Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5021–5031). Online: Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS 2019* (pp. 5754–5764). Vancouver, Canada.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020* (Vol. 119, pp. 11328–11339). PMLR.

Zhang, R., Li, W., Liu, N., & Gao, D. (2016). Coherent Narrative Summarization with a Cognitive Model. *Computer Speech & Language*, *35*, 134–160.

Zhang, S., & Bansal, M. (2021). Finding a Balanced Degree of Automation for Summary Evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 6617–6632). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020.* Addis Ababa, Ethiopia.

Zhao, W., Strube, M., & Eger, S. (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* (pp. 3847–3865).

Zhao, Z., Cohen, S. B., & Webber, B. (2020). Reducing Quantity Hallucinations in Abstractive Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 2237–2249). Online: Association for Computational Linguistics.

Zheng, H., & Lapata, M. (2019). Sentence Centrality Revisited for Unsupervised Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 6236–6247). Florence, Italy: Association for Computational Linguistics.