Cross-Document Distillation via Graph-Based Summarization of Extracted Essential Knowledge

Luca Ragazzi, Gianluca Moro, Lorenzo Valgimigli, and Riccardo Fiorani

Abstract—**Abstractive multi-document summarization aims to generate a comprehensive summary that encapsulates crucial content derived from multiple input documents. Despite the proficiency exhibited by language models in text summarization, challenges persist in capturing and aggregating salient information dispersed across a cluster of lengthy sources. To accommodate more input, existing solutions prioritize sparse attention mechanisms, relying on sequence truncation without incorporating graph-based modeling of multiple semantic units to locate essential facets. Furthermore, the limited availability of training examples adversely impacts performance, thereby compromising summarization quality in real-world few-shot scenarios. In this paper, we present G-SEEK-2, a graph-enhanced approach designed to distill multiple topic-related documents by pinpointing and processing solely the pertinent information. We use a heterogeneous graph to model the input cluster, interconnecting various encoded entities via informative semantic edges. Then, a graph neural network locates the most salient sentences that are provided to a language model to generate the summary. We extensively evaluate G-SEEK-2 across seven datasets spanning various domains—including news articles, lawsuits, government reports, and scientific texts—under few-shot settings with a limited training sample size of only 100 examples. The experimental findings demonstrate that our model consistently outperforms advanced summarization baselines, achieving improvements as measured by syntactic and semantic metrics.**

Index Terms—**Multi-document summarization, graph-enhanced transformer, few-shot learning.**

I. INTRODUCTION

With a constant influx of new digital information, we are witnessing an exponential proliferation of textual data. Documentation plays a crucial role in grasping useful insights in various domains, including healthcare, law, and science journalism. In fact, workers invest considerable time in summarizing multiple topic-related documents into a unified text, whether it involves compiling outcomes from various lawsuits [1] or detecting key events from collections of news articles [2]. As a consequence, the proliferation of unstructured information has expanded the documentation workload, directly contributing to increased stress and burnout [3]. Even for attorneys with a high level of expertise, this intricate task naturally demands hours to accomplish, posing challenges for timely production [1]. Hence, there is a need to develop automated tools to accelerate human productivity.

In light of recent advances in natural language processing (NLP), there has been a surge in interest in abstractive summarization, which surpasses traditional extractive methods by adeptly paraphrasing the most significant details of documents. Automatic text summarization tools play a vital role in helping people access the information they need [4], including lay summarization to increase readability and comprehension for non-experts [5]. In this context, a particularly challenging and practical task is the processing, identification, and synthesis of key information from a multitude of related sources, known as multi-document summarization (MDS) [6]. These assistive tools have received widespread attention across needs, ranging from query-focused summarization [7] to opinion summarization [8]. However, the complexities arising from the large volume of information and the inherent nature of documents—which often complement, overlap, or even contradict each other [9], [10]—contribute to the strong attention that the research community devotes to the advancement of MDS.

State-of-the-art MDS solutions are predominantly based on transformers [11], characterized by a structural constraint that links memory usage directly with input size, making them excessively resource-intensive when processing long texts. This limitation restricts the models to read only a fixed number of tokens,[1] introducing complications in MDS as it results in the truncation of any surplus information. In fact, unlike single document summarization, MDS methods must handle multiple texts, whose concatenation can form extremely lengthy inputs (e.g., 119,072.6 average words in MULTI-LEXSUM [1]). Therefore, standard sequence-to-sequence models, such as BART [13], are inadequate for MDS as they inevitably truncate long inputs, causing information loss and model degradation [14].[2] These problems are further exacerbated in real-world low-resource scenarios [16] characterized by a shortage of labeled instances available for model training supervision [17]. First, within small and medium organizations, creating gold-standard summaries from multiple lengthy

Received 3 July 2024; revised 17 September 2024; accepted 8 October 2024. Date of publication 1 November 2024; date of current version 27 January 2025. The work of Luca Ragazzi and Lorenzo Valgimigli was supported by Maggioli Group for granting the Ph.D. scholarship. This work supported in part by the Artificial Intelligence for Public Administration Connected (AI-PACT), under Grant CUP B47H22004450008 and Grant B47H22004460001, PNRR, mission 4, component 2, investment 2.3, in part by the DigitAl lifelong pRevEntion (DARE) under Grant CUP B53C22006450001 and Grant PNC0000002, in part by the Complementary National Plan PNC-I.1, in part by the Future Artificial Intelligence Research (FAIR), Spoke 8, PNRR—M4C2—Investment 1.3, in part by the Extended Partnership under Grant PE00000013, in part by the Chips JU TRISTAN project, G.A., under Grant 101095947, and in part by the European Commission and Italian MIMIT. The associate editor coordinating the review of this article and approving it for publication was Prof. Jiajun Zhang. *(Luca Ragazzi and Gianluca Moro contributed Equally to this work.) (Corresponding author: Gianluca Moro.)*

The authors are with the Department of Computer Science and Engineering, University of Bologna, 47522 Cesena, Italy (e-mail: l.ragazzi@unibo.it; gianluca.moro@unibo.it; lorenzo.valgimigli@unibo.it; riccardo.fiorani2@studio.unibo.it).

Digital Object Identifier 10.1109/TASLP.2024.3490375

[1]Tokens are subwords yielded by a subword tokenizer [12].

[2]Following [15], the input is the concatenation of documents in the cluster, each truncated according to the input length limit divided by the total number of sources, ensuring that each one is represented in the input (see Fig. 1).

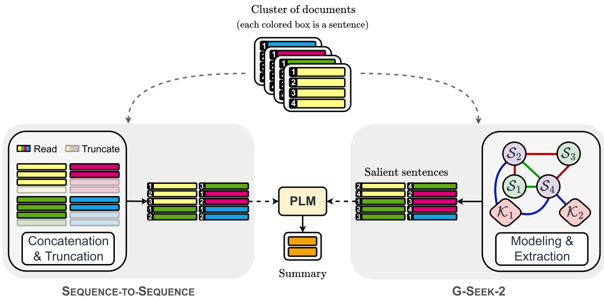

Fig. 1. Overview of G-SEEK-2 (right). Unlike standard solutions (left), we convert the document cluster into a heterogeneous graph: pink diamonds are keyword nodes $\mathcal{K}$, violet circles are sentence nodes $\mathcal{S}$ containing keywords, green circles are the context of $\mathcal{S}$, and the segments between nodes are edges. We extract salient sentences to produce the summary.

documents can be costly, time-consuming, and may necessitate the expertise of domain specialists. Second, poorly correlated input–output training pairs can hinder effective learning [18]. Therefore, few-shot MDS emerges as a significant research area that deserves more attention from the NLP community [19]. Addressing these limitations presents an opportunity to accelerate real-world processes, thus also alleviating the burden associated with documentation.

In this work, we draw inspiration from the conventional approach humans adopt when engaging in text summarization tasks. Specifically, humans read documents and highlight sentences that are deemed to be of greater importance. Subsequently, they review the underscored text and generate a summary based on it. This natural approach enables them to successfully synthesize extensive articles. A promising strategy to emulate this process involves using a two-stage pipeline approach that first extracts salient snippets and then summarizes them [20]. We face this task by using a semantic graph to represent the documents. This approach intuitively aggregates all source information, facilitating the identification and extraction of summary-worthy sentences. Previous contributions have used graph representations for text summarization [21]. Yet, they exhibit the following limitations: (i) they are mainly proposed for extractive summarization [22], [23], [24]; (ii) they are tailored for short texts [25], [26], which diverge highly from MDS settings; (iii) they do not leverage state-of-the-art generative pre-trained language models (PLMs) [27].

In light of this, we present G-Seek-2 (Fig. 1),[3] a graph-based summarization of extracted essential knowledge. Our approach selects the most relevant sentences from a cluster of related documents and feeds them to a PLM to generate the summary. Technically, we model documents with a heterogeneous graph composed of multiple semantic edges and nodes of different granularities (i.e., keywords and sentences). Then, a graph neural network (GNN) is trained to select the salient sentence nodes—soft labeled with a heuristic—which are thereby provided to a PLM to produce the summary. This approach bypasses the limitation of feeding models only until their maximum input size, which otherwise prevents them from fully processing the entire source, leading to performance drop [28]. Furthermore, by modeling a heterogeneous graph, we effectively capture cross-document relationships, a crucial aspect in handling multiple inputs [8]. Experimentally, we benchmark our solution in a realistic low-resource setting where a limited number of labeled training instances are available. This scenario is motivated by two key factors: (i) PLMs exhibit enhanced performance in generating summaries when trained with highly correlated source–target samples [18]; (ii) The limited number of trainable parameters of our learnable module (4M) allows G-Seek-2 to avoid overfitting with a small number of examples. The experimental results yielded by quantitative and qualitative analyses register improved performance of summarization baselines equipped with G-Seek-2 across various established evaluation metrics and datasets.

A conference version of this paper was presented in the Main Track of the 26th European Conference on Artificial Intelligence (ECAI 2023) [29]. In this manuscript, we enhance our work by refining model design, conducting additional in-depth comparative experiments, and expanding applications. The main differences are summarized as follows.

- To dissect the effectiveness of accurately capturing sentence node saliency in the graph, we conduct experiments with six different GNNs, such as Graph Convolutional Networks (GCN) [30], GraphSAGE [31], Graph Attention Networks (GAT) [32], Graph Isomorphism Networks (GIN) [33], Deep-GCN [34], and EdgeGCN [35].
- We evaluate our solution using three additional MDS datasets: Multi-News [36], WCEP [2], and WikiCatSum-animal [37], expanding our testbed to a total of seven corpora from various domains—news articles, lawsuits, government reports, and scientific texts—thoroughly exploring the generalizability of our approach.

The remainder of this paper is organized as follows. Section II offers an overview of related work on MDS. We present our method in Section IV and the experimental setup in Section V. Our results are reported and discussed in Section VI. Lastly, Section VII provides final remarks, highlighting limitations and future directions.

II. Related Work

In this section, we provide a brief overview of related work on MDS, focusing on commonly used approaches and the utilization of graphs to model cross-document relationships.

A. Sequence-to-Sequence Pre-Trained Language Models

Generative PLMs have demonstrated robust performance and adaptability to MDS [6], also when enhanced with reinforcement learning [38], and long-input summarization tasks in general. These sequence-to-sequence models are built on the transformer encoder–decoder architecture, denoted by stacks of self-attention layers. Vanilla transformers, such as Bart [13] and Pegasus [39], are limited to process up to 1024 tokens due to their quadratic memory and time complexity w.r.t. the input size. Consequently, they are not suitable for long sequences—including MDS—made up of tens of thousands of tokens. To address this obstacle, linear PLMs such as Led [40] and Primera [15] feature a sparse attention mechanism that replaces the full quadratic self-attention by allowing models to scale linearly w.r.t. the input length. As a result, these models are capable of reading longer texts (e.g., up to 4096 tokens for Primera), making them more suitable for MDS tasks. Despite the advantage of linear transformers in handling extensive information, they still rely on input truncation, like their quadratic counterparts. This constraint entails processing the source only up to the model's maximum input size (see Fig. 1), thus overlooking potentially relevant details that merit inclusion in the summary.

[3]https://disi-unibo-nlp.github.io/publications/

B. Approaches for Multi-Document Summarization

MDS methods that do not adhere to a standard transformer-based sequence-to-sequence approach for handling multiple documents fall primarily into three categories.

a) Two-Phase Solutions: These methods, often referred to as "extract-then-abstract," operate by selecting sentences deemed suitable for summarization, which are subsequently condensed by a PLM. Beyond unsupervised alternatives for sentence extraction [20] and abstractive summarization from clustered documents [41], most of the contributions embrace supervised techniques. Previous strategies relying on TF-IDF [42] have been replaced by ROUGE-based extractors, which aim to label sentences as relevant by measuring their similarity w.r.t. the summary [43]. This approach has also been addressed by jointly training the extractor with the summarization module [44]. In contrast to prior works, we explore different metrics for sentence soft-labeling, covering BLEU [45] and all ROUGE [46] variants, such as ROUGE-1, ROUGE-2, and ROUGE-L, with precision and F1 assessments.

b) Aggregation-based Methods: These techniques involve combining hidden-states to aggregate information sourced from various snippets. Fusion-in-decoder [47] generates a unified hidden-state by concatenating multiple representations before decoding. On the other hand, marginalization-based models employ logit likelihood summation during decoding across inputs to weigh the probability of the next token [48], [49], [50]. Nevertheless, these solutions face difficulties when dealing with a limited number of labeled samples, primarily due to the initial cold-start phase required to train the model to effectively exploit this aggregated representation.

c) Hierarchical Models: These solutions aim to capture intricate interactions among documents to attain semantic-rich representations. Efforts have focused on enhancing transformers through graph-based techniques [51], [52], [53], [54], multi-head grouping and inter-paragraph attention [55], [56], maximal marginal relevance [36], and the inclusion of global and local attention [57], [58], [59]. Although hierarchical solutions have shown promising results, they struggle to effectively accommodate and leverage state-of-the-art PLMs [15].

C. Graph-Based Summarization

Graphs and GNNs have emerged as integral components in MDS [21], offering enhanced scalability [60] and improved domain modeling [61] to mitigate transformer flaws. Several contributions leverage GNNs as standalone solutions [62], where the summary is generated by composing sentences extracted from input documents [63], [64]. Conversely, GNNs can be embedded with abstractive summarization models to improve performance [53]. Along this thread, BASS [27] introduces a unified semantic graph to represent the collection of texts and modifies the transformer architecture to interact with the graph. SKGSUM [25] exploits nodes at various levels to guide the summary generation process. To capture cross-document relationships, many studies have sought to construct different types of homogeneous graphs (e.g., topic graphs, discourse graphs) [53], [65], [66], [67]. Several studies have also delved into the use of heterogeneous graphs [68], [69], [70], [71]. However, these graphs accommodate only different types of nodes (e.g., word and sentence nodes) without considering diverse meaningful edges. On the other hand, HGSUM [72] extends a linear transformer with a heterogeneous graph of multiple semantic nodes and edges, but requires joint training that comes with the drawback of a costly pipeline.

Algorithm 1: Sentence Soft-Labeling.

Input: $\mathcal{X} = \{x_1, \ldots, x_x\}$ (*Input sent*)
$\mathcal{Y} = \{y_1, \ldots, y_y\}$ (*Output sent*)
Parameters: $\mathcal{M}$ (*Similarity metric*)
Output: $\mathcal{S}$ (*Set of scores*)
$\mathcal{S} \leftarrow \emptyset$
for $x_i \in \mathcal{X}$ **do**
  $s \leftarrow \emptyset$
  for $y_i \in \mathcal{Y}$ **do**
    $s.append(\mathcal{M}(x_i, y_i))$
  end for
  $\mathcal{S}.append(max(s))$
end for
return $\mathcal{S}$

III. Research Objective

The goal of this research is to develop an automatic MDS approach that can address the following real-world challenges.

- *Long-input processing:* Design an extract-to-abstract framework to adeptly manage large information prevalent in MDS, bypassing input truncation drawbacks encountered in conventional sequence-to-sequence solutions.
- *Few labeled examples:* Integrate a powerful PLM to work successfully under few-shot conditions.
- *Cross-document relationships:* Leverage a heterogeneous graph to accurately encode document interrelations, and employ state-of-the-art GNNs to analyze their role in discerning significant patterns.
- *Generalizability:* Analyze system effectiveness across a set of multiple corpora from different domains through both syntactic and semantic evaluation metrics.

IV. Method

We present G-SEEK-2, a *graph-based summarization of extracted essential knowledge* (see Fig. 1). Section IV-A delineates the preliminary procedures integral to our method, including the required sentence labeling and the compilation of passages essential for graph construction from extended inputs, using semantic and structural data. Section IV-B delves into the array of GNNs used to discern the interrelations among graph nodes, facilitating the identification of key sentences. Finally, Section IV-C provides the summarization pipeline.

A. Preliminaries

1) Sentence Soft-Labeling: Two-stage approaches have underscored the need to select suitable summary-worthy sentences from the source documents [44], [73]. Conceptually, each sentence can be categorized as relevant or irrelevant to the intended summary, enabling the training of a model to discern such noteworthy sentences. However, in the absence of ground-truth relevance labels—which reflects a characteristic of real datasets—we need to heuristically mark the salience of the input sentences w.r.t. the gold summary, namely performing a soft-labeling strategy. Formally, let $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$ and $\mathcal{Y} = \{y_1, \ldots, y_{|\mathcal{Y}|}\}$ be the long input (i.e., the concatenated documents of the cluster) and the corresponding summary,

TABLE I
DIVERSE SOFT-LABELING IN MULTI-LEXSUM

Metric	R-1$_{f1}$	R-2$_{f1}$	R-L$_{f1}$
BLEU	43.62	19.18	28.58
R1-F1	44.06	19.33	29.32
R1-P	40.97	16.96	26.51
R2-F1	**45.29**	**20.20**	**30.19**
R2-P	43.32	19.20	29.14
RL-F1	43.18	19.34	28.30
RL-P	43.90	19.14	29.15

Notably, ROUGE-2-F1 results the most effective metric for the sentence labeling task.

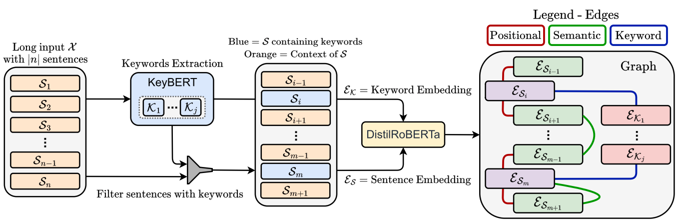

Fig. 2. The pipeline for generating a semantic heterogeneous graph. We provide the textual information to KEYBERT, which generates a set of unique keywords. Subsequently, (i) sentences containing at least one keyword, (ii) their surrounding context (i.e., sentences preceding or following), and (iii) the keywords, are transformed into embeddings using DISTILROBERTA, serving as the new nodes interconnected by meaningful edges (positional, semantic, and keyword edge).

respectively, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is a sentence. We perform a greedy algorithm (Algorithm 1). Initially, for each instance x_i, we yield a list of relevance scores $\in [0, 1]$ through pairwise similarity computations among the sentences in the corresponding instance y_i. Subsequently, the highest scoring value is selected and assigned to x_i. For the assessment of sentence similarity, we examine various evaluation metrics, such as BLEU and ROUGE-{1,2,L}, considering both precision and F1 scores. Evaluation is carried out using the MULTI-LEXSUM-SHORT dataset [1] as testbed, using the first 100 samples from both training and validation sets. To appraise the summarization quality, we employ PRIMERA [15] as the backbone model, which is a transformer with linear complexity in the input length pre-trained with an MDS-specific objective. Practically, after assigning a summary-relevance score to each $x_i \in \mathcal{X}$, PRIMERA is furnished only with sentences (sequenced as per the source text) possessing the highest scores, up to the model's maximum input size, which is 4096 tokens. The results, detailed in Table I, highlight ROUGE-2 F1 as the most effective metric for soft labeling. Consequently, this metric is adopted for the relevance labeling of sentences across all experiments. It is pertinent to note that Table I encapsulates the results derived from also extending soft labeling to the validation set. Pointedly, during inference, an oracle simulation is performed by accessing the ground-truth target summaries, thus facilitating the examination of upper-bound performance.

2) Heterogeneous Graph: We delineate the construction of our heterogeneous graph through the following steps, as illustrated in Fig. 2.

a) Keyword Extraction: First, we eliminate English stopwords and general domain-specific terms appearing in more than 40% of the cluster. Then, we employ KEYBERT [74], [75] to identify the top-k keywords, which is a lightweight method in

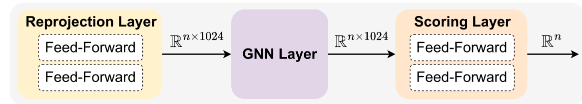

Fig. 3. Overview of our learnable module to model cross-document relationships over the heterogeneous graph.

contrast to more resource-intensive alternatives [76]. Operationally, we extract k keywords for each document in the cluster and combine these keyword lists into a unique set, removing any duplicates.

b) Sentence Filtering: We partition the documents into sentences and select those containing at least one keyword. Additionally, we capture the surrounding context by retrieving the n sentences preceding and succeeding the selected one. Formally, we define $\{[x_1^1, \ldots, x^1_{|\mathcal{X}^1|}], \ldots, [x_1^z, \ldots, x^z_{|\mathcal{X}^z|}]\}$ as the cluster of sources z. Let x_b^1 and x_e^1 be two sentences with keywords. We then select $[x^1_{b-n}, \ldots, x^1_{b+n}, x^1_{e-n}, \ldots, x^1_{e+n}]$, where $< x^1_{b-n}, \ldots, x^1_{b+n} > -\{x_b\}$ is the context of x_b .

c) Sentence & Keyword Embedding: We use a frozen pre-trained DISTILROBERTA model [77] to generate the embeddings of three types of texts with different semantics: (1) keywords, (2) sentences containing keywords, and (3) neighboring sentences (i.e., the surrounding context) of the latter. DISTILROBERTA is characterized by a relatively small parameter count (82M), ensuring efficiency in terms of GPU memory utilization and computation. Notably, the model is already pre-trained to create sentence embeddings through a self-supervised contrastive learning objective.[4] About creating sentence embeddings, the model produces a representation for each token within a sentence. Then, following [78], we employ mean pooling to aggregate token embeddings, yielding a final single vector representation denoted as e_i^x.

d) Graph Creation: All keyword and sentence embeddings, denoted as KE and SE, respectively, are represented as nodes within our graph. Inspired by [27], we designate KEs as supernodes, indicating that every sentence containing a keyword establishes bidirectional *Keyword Edges* (cf. the blue lines in Fig. 2) with the corresponding keyword node, rather than forming connections solely among themselves. Then, we introduce bidirectional *Positional Edges* (cf. the red lines in Fig. 2) between two SEs if they appear consecutively in the source text. Finally, in alignment with [53] and [24], we incorporate *Semantic Edges* (cf. the green lines in Fig. 2) between e_i and e_j if their cosine similarity is greater than a threshold t. The ultimate graph has as many nodes as the combined number of sentences and keywords.

B. Cross-Document Modeling

To capture cross-document relationships, we employ a learnable module featured by a GNN on our heterogeneous graph, discerning the significance of sentences by assigning an unbounded positive relevance score to each sentence node. We harness both nodes (semantic information) and edges (structural information), essential for propagating information across nodes. This enables the GNN to attain a deeper comprehension of the context and meaning of each sentence. Our module comprises the following layers (see Fig. 3):

[4]https://huggingface.co/sentence-transformers/all-distilroberta-v1.

- **Reprojection Layer** has two feed-forward layers (FFL) for learning the transformation of node embeddings x within the vector space, resulting in reprojected embeddings x'. Functionally, it augments the dimension d_x of input embeddings n ($\mathbb{R}^{(n \times 768)}$) by a factor termed Boom Factor (BF): $x' = \mathrm{FFL}_\sigma(\mathrm{FFL}_\gamma(x, d_x \cdot \mathrm{BF}), d_x)$ where sigma and gamma are parameters of distinct linear layers.
- **GNN Layer** operates on graph-structured data by exploiting structural information to improve the semantic representation of nodes and capture relational dependencies and patterns. We test multiple GNNs.[5]
  – GCN (Graph Convolutional Network) [30] uses convolutional operations to propagate information between nodes in the graph by leveraging a localized aggregation of neighboring node features.
  – GRAPHSAGE [31] is an inductive algorithm that learns a function to generate node embeddings by sampling and aggregating features from the local neighborhood of a target node.
  – GAT (Graph Attention Network) [32] assigns attention weights to each node, allowing the network to focus on the most influential neighbors throughout the information propagation process.
  – GIN (Graph Isomorphism Network) [33] aims to generalize the Weisfeiler-Lehman graph isomorphism test to achieve the maximum discriminative power.
  – DEEP-GCN [34] leverages residual and dense connections and dilated convolutions into GCNs.
  – EDGEGCN [35] uses multi-dimensional edge features for explicit relationship modeling.
- **Scoring Layer** has two linear layers for reducing the dimensionality of each node (x) to a singular real number (s), serving as relevance score for the sentence linked to the node $s = \mathrm{FFL}_\beta(\mathrm{FFL}_\theta(\widehat{x}, d_{\widehat{x}} \cdot \mathrm{BF}), 1)$.

C. Summarization Pipeline

After converting the multi-document input into a graph and determining the relevance scores for each sentence, we employ a PLM to generate the output summary from the most pertinent sentences. Technically, based on their relevance scores, we select the most salient sentences and construct an input text for the model with fewer tokens than its maximum input size (i.e., without text overflow that would require truncation). Consequently, the new input comprises only important sentences arranged in the order of their occurrence in the original source.

We train the summarizer using the standard cross-entropy loss, wherein the model is tasked with predicting the next token w_i of the target $\mathcal{Y}$ given the input $\mathcal{X}$ and the previous target tokens $w_{1:i-1}$, formulated as $\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{|\mathcal{Y}|} \log p_\tau(w_i | w_{1:i-1}, \mathcal{X})$ where τ denotes the model parameters and p represents the predicted probability distribution over the vocabulary.

V. EXPERIMENTAL SETUP

In this section, we describe the experimental datasets, evaluation metrics, implementation details, and baselines used for comparison. Our research focuses primarily on the task of few-shot summarization, characterized by a real-world scenario marked by limited data availability for model supervision, mainly due to the high cost associated with annotation. In line with prior studies [18], [79], we select the first 100/10/100 samples from the training, validation, and test sets of all datasets without engaging in further data pre-processing.

[5] We direct the reader to consult original papers for technical information.

TABLE II
STATISTICS OF THE DATASETS. EXCEPT FOR THE NUMBER OF SAMPLES, ALL REPORTED VALUES ARE AVERAGED ACROSS ALL INSTANCES.

			Source			Target		Source → Target		
Dataset	**Domain**	**Samples**	Docs	Words†	Sents	Words	Sents	Coverage	Density	Compress
MULTI-LEXSUM-TINY [1]	Legal	1603	10.7	119072.6	5962.5	24.7	1.4	0.92	2.27	5449.6
MULTI-LEXSUM-SHORT [1]	Legal	3138	10.3	99378.2	5017.0	130.2	5.1	0.96	3.33	840.7
MULTI-LEXSUM-LONG [1]	Legal	4534	8.8	75543.2	3814.2	646.5	28.8	0.94	4.07	97.4
MULTI-NEWS [29]	News	56,206	2.8	2092.1	80.9	257.9	10.0	0.82	5.47	8.1
WCEP [2]	News	10200	10.0	4356.1	154.0	31.9	1.4	0.91	3.17	162.1
WIKICAT-SUM-ANIMAL [30]	Scientific	53,638	131.0	5091.8	288.5	92.0	4.6	0.78	3.07	81.4
GOVREPORT [69]	Legal	19,463	1.0	8765.0	298.7	556.3	18.1	0.94	9.08	17.9

† The total average number of words in the source cluster; we consider a single input by concatenating all the documents.
Except for the number of samples, all reported values are averaged across all instances. We observe a comprehensive range of input lengths, which challenges our model and provides an in-depth evaluation benchmark.

A. Datasets

We perform experiments using multiple datasets from various domains that are publicly available in HuggingFace, serving as widely recognized benchmarks for MDS tasks. **MULTI-LEXSUM** [1] comprises real-world federal civil rights lawsuits accompanied by summaries authored by experts. The primary challenge encountered in MULTI-LEXSUM lies in the extended length of the source documents and the varying granularity of the summaries, including tiny, short, and long versions (see Table II). Owing to this multifaceted nature, we conduct experiments using three distinct dataset renditions as testbed. **MULTI-NEWS** [36] is a large-scale dataset in which each instance comprises multiple news articles gathered from various sources, accompanied by a summary crafted by professional editors. **WCEP** [2] is constructed based on news events sourced from the Wikipedia Current Events Portal. It comprises clusters of news events along with human-authored summaries. We focus on the WCEP-10 dataset, which features 10 related articles per instance. **WIKICAT-SUM-ANIMAL** [37] is a collection of news articles related to the domain of animals. Finally, **GOVREPORT** [81] consists of lengthy reports from the US government. Although it is used in long document summarization benchmarks, we regard GOVREPORT as an intriguing testbed owing to its notably large input size, similar to [72] with the ARXIV dataset.

The key statistics of the datasets are presented in Table II. Specifically, we provide the number of samples in the corpus, the average number of documents per cluster, and the average number of words and sentences in both the source documents and target summaries—computed using the NLTK library [82]. Additionally, we report the average coverage, density, and compression ratio of extractive fragments, as defined by [80]. Technically, coverage is defined as the average fraction of token spans that can be identified in both the source and target. For instance, a coverage of 0.94 indicates that 94% of the summary words appear in extractive source fragments. Density, on the other hand, represents the average length of the extractive fragments. Finally, the compression ratio quantifies the extent to which a text is condensed to produce its summary.

Pointedly, our evaluation testbed encompasses several challenges: (i) processing very long legal reports (i.e., $> 100K$ words) with an extremely high compression ratio [1]; (ii) generating extremely short summaries [1], [2]; (iii) handling highly abstractive syntheses, where the targets contain few source-related tokens, as indicated by the low coverage [37]; and (iv) managing highly dense summary phrases [36], [81].

B. Metrics

We use **ROUGE**-{1,2,L}[6] F1[46] and **BERTScore** F1 (BS) [83] to quantify the syntactic and semantic correspondence, respectively, between the generated summaries and the ground-truth. Further, we compute $\mathcal{R} = \text{avg}(r_1, r_2, r_L) / 1+\sigma_r^2$ [84] to aggregate ROUGE, where σ_r^2 represents the variance of the average ROUGE scores, penalizing discrepancies in performance across different dimensions. All metrics lie within the range [0,1], with higher scores indicating better performance. Reports additional details.

C. Baselines

While decoder-only architectures driven by large language models (LLMs) have gained popularity for news summarization [85], [86], their application in MDS remains unexplored, and recent research proves that encode.decoder networks offer superior summarization performance [87]. Thus, to assess the efficacy of our proposed method in filtering relevant information prior to inputting it into generative models, we select several widely recognized and leading MDS solutions notable for their distinct capabilities in handling various input sizes. We then compare their performance when enhanced with G-SEEK-2 (4M parameters). **BART** [13] (400M) is a transformer-based model with quadratic memory and time complexity concerning input length. **PEGASUS** [39] (568M) is a quadratic transformer pre-trained specifically for summarization tasks, employing an objective to predict gap sentences as pseudo summaries. Like **BART** it can read up to 1024 tokens. **LED** [40] (459M) is a transformer model with linear memory complexity due to a sparse attention mechanism that reads 4096 tokens. **PRIMERA** [15] (447M) is a linear transformer built upon **LED** but with a pre-training objective specifically tailored for **MDS**, generating pseudo summaries by automatically extracting text spans based on entity salience.

D. Implementation Details

We fine-tune the models using the PyTorch implementations provided by the HuggingFace library [88], ensuring reproducibility by setting the seed to 42. All experiments are conducted on an internal workstation equipped with a Nvidia RTX 3090 GPU with 24 GB of memory, 64 GB of RAM, and an Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz processor. For the GNN module, training is carried out over 75 epochs with a learning rate of $5e^{-5}$, using AdamW as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. It should be noted that we trained all GNNs once on MULTI-LEXSUM-SHORT using soft labels as described in Section IV-A-1. Regarding the summarization task, all models are trained for 5 epochs with a learning rate of $3e^{-5}$, using mixed precision and gradient checkpointing techniques to conserve memory. During decoding, we employ beam search with 5 beams and n-gram repetition blocking for $n > 5$.

VI. Results and Discussion

A. Analysis of the Graph

We explore various configurations of graph settings, with particular emphasis on the *Sentence Filtering* module (see Section IV-A-2), where we evaluate Precision and Recall of all

TABLE III
PRECISION, RECALL, AND F-MEASURE RESULTS OF DIFFERENT GRAPH SETTINGS ON THE EVAL SET OF MULTI-LEXSUM-SHORT

#	P	R	F1
	Keywords		
4	22.12	76.43	34.31
5	**23.29**	**78.81**	**35.95**
6	22.98	75.42	35.22
7	22.65	77.26	35.03
8	22.57	77.12	34.92

#	P	R	F1
	Consecutive Sentences		
1	**23.26**	62.64	33.92
2	22.12	76.43	34.31
3	23.24	76.00	35.60
4	23.15	**78.85**	**35.79**
5	23.06	77.42	35.54

TABLE IV
SENTENCE CLASSIFICATION SCORES WITH DIFFERENT GNNS ON EACH DATASET EQUIPPED WITH SOFT LABELS. FOR EACH CORPUS, THE BEST RESULTS ARE BOLDED AND THE SECOND-BEST ARE UNDERLINED.

GAT		GCN		GIN		DEEP-GCN		EDGEGCN		GRAPHSAGE	
P	R	P	R	P	R	P	R	P	R	P	R
MULTI-LEXSUM-TINY											
35.2	48.0	30.1	43.5	28.7	42.0	**37.8**	**52.4**	36.9	51.5	34.4	48.9
MULTI-LEXSUM-SHORT											
38.1	52.5	35.4	50.3	32.7	47.7	**39.6**	**54.7**	39.1	54.2	37.5	52.5
MULTI-LEXSUM-LONG											
37.0	51.0	33.0	47.5	31.5	45.8	**38.5**	**53.2**	37.9	52.7	36.2	50.6
MULTI-NEWS											
63.0	63.0	60.0	60.0	58.5	58.5	62.5	62.5	63.0	63.0	**67.0**	**67.0**
WCEP											
78.3	78.3	75.0	75.0	74.0	74.0	75.1	75.1	74.8	74.8	**80.5**	**80.5**
WIKICAT-SUM-ANIMAL											
62.8	62.8	60.5	60.5	59.0	59.0	60.2	60.2	61.2	61.2	**63.4**	**63.4**
GOVREPORT											
38.0	50.0	35.2	47.8	32.5	45.6	**39.0**	**52.0**	38.5	51.5	37.0	50.0

For each corpus, the best results are bolded and the second-best results are underlined. Notably, DEEP-GCN consistently outperforms other models in legal datasets, marking it as the most effective choice for these corpora. Conversely, GRAPHSAGE proves to be the superior option for other types of datasets.

TABLE V
THE RESULTS OF THE LEARNABLE MODULE ON THE VALIDATION SET OF MULTI-LEXSUM-SHORT UNDER DIFFERENT SETTINGS WITH 30 TRAINING EPOCHS

Value	P	R	F1
	Boom Factor		
1	31.34	38.43	34.52
2	**32.12**	**39.17**	**35.29**
3	31.69	38.80	34.88
4	31.52	38.54	34.68
	GAT Layers		
1	**35.38**	**42.31**	**38.54**
2	32.13	39.18	35.31
3	28.34	35.13	31.37
4	29.45	36.37	32.55

Value	P	R	F1
	Cosine Similarity		
0.80	32.02	39.01	35.17
0.82	32.10	39.18	35.28
0.84	32.10	39.17	35.28
0.86	**32.49**	**39.55**	**35.67**
0.88	31.96	39.01	35.13
	Final Module		
-	38.37	45.49	41.63

The "Final Module" is the best setting and checkpoint after 75 epochs.

labeled sentences among the selected salient ones. The results of these investigations are summarized in Table III, conducted over the validation set of the MULTI-LEXSUM-SHORT dataset and considering the following facets:

- *Keywords:* We analyze the maximum number of keywords extracted by KEYBERT for each document in the cluster. We observe that 5 keywords yields the most favorable outcomes.
- *Context:* We investigate varying numbers of consecutive neighboring sentences selected as the context of the salient ones (i.e., those containing keywords). Results suggest that 4 sequences result in improved performance.

In our hardware environment (see Section V-D), the average time to create the graph for a single long input containing approximately 100K words is $\approx$ 34 seconds. Regardless, it is pertinent to note that the current implementation does not incorporate any specific optimizations.

[6]We use ROUGE-L after segmenting the summary into sentences.ROUGE-{1,2,L}\footnote{We use ROUGE-L after segmenting the summary into sentences.

TABLE VI
EVALUATION F1 SCORES ON MULTI-LEXSUM-{TINY, SHORT, LONG} AND GOVREPORT

Model	MULTI-LEXSUM-TINY R-1	R-2	R-L	$\mathcal{R}$	BS	MULTI-LEXSUM-SHORT R-1	R-2	R-L	$\mathcal{R}$	BS	MULTI-LEXSUM-LONG R-1	R-2	R-L	$\mathcal{R}$	BS	GOVREPORT R-1	R-2	R-L	$\mathcal{R}$	BS
Quadratic																				
BART	22.37	7.91	19.74	16.61	76.17	41.45	**18.74**	35.81	31.70	79.89	41.41	16.47	38.98	31.88	79.13	48.46	14.03	44.83	34.94	80.82
w/ G-SEEK	24.46	7.70	20.34	17.41	77.07	41.57	16.72	35.78	31.01	79.97	43.92	17.43	41.08	33.67	80.19	51.46	17.12	48.05	37.97	**81.78**
w/ G-SEEK-2	**27.72**†	**11.04**†	**22.64**†	**20.49**†	**79.58**†	**42.48**†	15.98	**37.57**†	**31.85**	**80.77**†	**48.08**†	**22.06**†	**45.57**†	**38.48**†	**82.15**†	**52.30**†	**17.39**	**48.65**†	**39.45**†	81.77
PEGASUS	15.09	3.20	12.07	10.09	70.27	38.29	16.31	32.63	28.83	78.58	40.19	16.13	37.82	31.02	78.39	47.12	14.07	44.82	34.55	80.51
w/ G-SEEK	19.84	5.13	16.28	13.70	73.12	38.78	16.32	33.26	29.18	79.11	42.38	17.04	39.88	32.68	79.53	50.55	**17.06**	48.01	37.67	**81.30**
w/ G-SEEK-2	**23.57**†	**8.85**†	**17.78**†	**16.23**†	**75.23**†	**42.46**†	**17.82**†	**37.77**†	**32.30**†	**80.59**†	**46.79**†	**21.94**†	**43.94**†	**37.56**†	**81.99**†	**50.76**	16.93	**48.12**	**38.58**†	81.29
Linear																				
LED	22.86	7.98	18.86	16.50	76.20	40.09	17.50	35.15	30.63	79.51	45.26	20.01	42.66	35.52	81.31	53.86	19.53	49.28	39.96	82.65
w/ G-SEEK	24.39	7.96	20.55	17.55	77.19	40.95	16.28	35.31	30.51	80.56	45.42	18.87	42.93	35.23	81.03	55.63	21.08	50.67	41.49	82.77
w/ G-SEEK-2	**27.74**†	**12.99**†	**22.38**†	**20.94**†	**78.43**†	**46.98**†	**21.08**†	**41.42**†	**36.49**†	**82.70**†	**50.94**†	**25.75**†	**47.58**†	**41.48**†	**83.34**†	**58.29**†	**22.37**†	**54.04**†	**44.87**†	**83.01**
PRIMERA	25.37	8.13	20.84	18.02	76.45	40.20	14.88	34.88	29.63	80.31	45.31	21.06	42.44	35.85	81.34	54.20	19.37	50.20	40.28	79.75
w/ G-SEEK	25.76	7.59	21.36	18.13	77.26	**43.99**	18.67	37.55	33.02	81.32	45.92	19.61	42.59	35.55	81.36	**57.13**	**21.20**	**53.64**	**42.87**	80.37
w/ G-SEEK-2	**27.53**†	**10.92**†	**22.52**†	**20.32**†	**78.01**†	43.65	**20.87**†	**38.09**	**34.54**†	**82.29**†	**49.51**†	**23.61**†	**46.24**†	**39.63**†	**82.59**†	55.59	20.54	50.84	42.24	**82.54**†

The best intra-model score is bolded. † means statistically significant results of G-SEEK-2 (p-value < 0.05 with student t-test). Remarkably, G-SEEK-2 demonstrates superior performance compared to both standard solutions and those enhanced by its predecessor.

TABLE VII
EVALUATION F1 SCORES ON THE NEW ADDED DATASETS, SUCH AS WIKICAT-SUM-ANIMAL, MULTI-NEWS, AND WCEP

Model	WIKICAT-SUM-ANIMAL R-1	R-2	R-L	$\mathcal{R}$	BS	MULTI-NEWS R-1	R-2	R-L	$\mathcal{R}$	BS	WCEP R-1	R-2	R-L	$\mathcal{R}$	BS
Quadratic															
BART	37.04	**17.76**	**27.24**	**30.10**	78.00	**43.94**	**13.42**	**20.08**	**32.34**	**79.10**	41.33	**24.89**	**33.72**	**33.87**	83.00
w/ G-SEEK	45.86	14.49	24.04	27.62	78.29	38.25	10.54	18.83	29.20	78.36	36.51	15.79	26.49	26.93	80.56
w/ G-SEEK-2	**39.64**	14.28	25.38	29.74	**79.20**†	40.68	9.97	17.78	28.37	78.21	**41.35**	18.09	29.28	29.75	**84.39**†
PEGASUS	37.45	13.98	23.15	26.45	78.18	33.70	8.42	16.63	23.50	77.58	44.59	25.16	37.04	35.82	82.55
w/ G-SEEK	36.80	13.44	23.33	27.88	78.90	39.63	11.18	17.77	28.13	77.58	46.25	25.17	35.74	35.75	86.17
w/ G-SEEK-2	**37.80**	**14.47**†	**24.48**†	**28.84**†	**79.10**†	**44.93**†	**13.43**†	**19.67**†	**32.44**†	**79.13**†	**48.97**†	**27.54**†	**41.65**†	**39.37**†	**85.58**
Linear															
LED	39.12	**18.27**	**28.60**	31.57	78.35	44.47	12.54	**19.56**	32.20	**79.30**	51.10	23.90	37.37	37.34	85.01
w/ G-SEEK	40.81	15.72	25.75	**31.60**	78.61	42.92	11.25	18.97	30.14	79.29	45.32	21.83	34.06	33.75	85.85
w/ G-SEEK-2	**40.94**	17.13	26.85	31.20	**79.73**†	**44.68**	**13.94**†	19.47	**32.35**	78.96	**51.26**	**26.29**†	**41.22**†	**39.27**†	**87.28**†
PRIMERA	44.18	19.49	28.83	33.16	80.00	39.90	**9.79**	**18.67**	28.88	78.30	46.24	**26.33**	38.07	**37.15**	84.49
w/ G-SEEK	40.71	16.69	27.58	31.92	79.66	**42.49**	10.34	18.00	**29.38**	78.41	43.71	25.82	35.33	35.45	84.63
w/ G-SEEK-2	**44.32**	**21.06**†	**29.33**	**35.70**†	**80.71**†	40.20	9.09	17.02	28.18	**78.89**	**48.13**†	25.42	**38.11**	36.91	**86.06**†

The best intra-model score is bolded. † means statistically significant results of G-SEEK-2 (p-value < 0.05 with student t-test). G-SEEK-2 demonstrates superior performance compared to standard solutions and those enhanced by its predecessor.

B. Analysis of the GNN-Based Module

We investigate different settings of our learnable module related to the inner architecture and the hyperparameters.

1) Architecture: Table IV reports the results obtained through various GNN architectures. The performance of sentence classification is assessed using two commonly employed evaluation metrics, such as Precision and Recall. First, we used only MULTI-LEXSUM-SHORT as our benchmark dataset. The results reveal that DEEP-GCN outperforms other architectures, demonstrating superior performance across all metrics. This suggests its efficacy in accurately identifying relevant sentences while minimizing false positives. Conversely, GCN and GIN yield comparatively poorer results, indicating that these architectures might not be well-suited for the dataset under consideration. Consequently, we conduct this experiment on all the evaluation datasets to further validate these observations, as can be seen from Table IV. In fact, the results reveal that no single GNN architecture outperforms the others in all datasets. Nevertheless, it is evident that DEEP-GCN emerges as the optimal selection for legal corpora, while GRAPHSAGE is preferable for other types. These findings emphasize the importance of choosing the appropriate model based on the specific characteristics and requirements of the dataset. Upon closer examination, certain datasets such as MULTI-NEWS, WCEP, and WIKICAT-SUM-ANIMAL display identical values for precision and recall. These datasets are characterized by shorter average input lengths, as detailed in Table II. Therefore, this feature may contribute to a more balanced task environment for determining the relevance of sentences.

2) Ablation Studies: We conduct experiments with our trainable module using the validation set of MULTI-LEXSUM-SHORT to analyze the best setting for the Reprojection and Scoring layers. Specifically, we employ GAT as our GNN layer, performing 30 training epochs and assessing efficacy by selecting the top 100 sentences based on their assigned scores. Subsequently, we calculate precision, recall, and f-measure metrics for these 100 sentences.

Table V presents the results of our experiments, where we evaluate the following items:

- *Boom Factor:* We examine the impact of the Boom Factor in the *Reprojection Layer*. We found that a value of 2 yields optimal performance.
- *Layers:* We test different numbers of layers and surprisingly find that having only 1 layer achieves the best results. This suggests that, due to the limited pool of training examples, a lightweight solution with fewer trainable parameters is preferable.
- *Cosine Similarity:* We analyze the threshold for creating *Semantic Edges* between nodes. We uncover that a threshold of 0.86 produces the best outcomes.

Table V additionally displays the results of the optimal module configuration trained for 100 epochs. Subsequently, we identify the most favorable model checkpoint, which occurs after 75 epochs. The average duration for each epoch is approximately 60 seconds. It should be noted that, after extensive experimentation, we systematically varied each hyperparameter while maintaining the other two constants at their respective optimal values (marked in bold in Table V).

C. Summarization Results

We train and evaluate all models on the evaluation datasets both with and without G-SEEK-2 to underscore our contribution. Additionally, we provide the results for G-SEEK, which

Model	Tokens	BS	Time (s)
BART	1024	76.59	8
w/ G-SEEK-2	1024	77.97	8
BART	512	75.33	6
w/ G-SEEK-2	512	77.87	6
BART	256	71.48	4
w/ G-SEEK-2	256	76.38	4

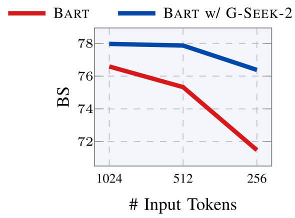

Fig. 4. Comparison of summarization quality with BERTScore (BS) by varying the number of input tokens on MULTI-LEXSUM-TINY. We report the time in seconds to compute the test set. On the right is furnished a graphical representation.

represents the initial version of our previous contribution [29]. Regarding the complexity analysis, models equipped with G-SEEK-2 have the same memory growth w.r.t. the input size of vanilla counterparts. More precisely, the space complexity to summarize the entire input document is $\mathcal{O}(L^2)$ for quadratic models (e.g., BART) and $\mathcal{O}(L)$ for linear ones (e.g., PRIMERA), where L is the minimum value between the source length and the model's maximum input size.

Table VI presents the performance of the systems on the datasets within the legal domain, while Table VII displays the results on the newly introduced corpora. Remarkably, our solution consistently enhances model performance across all datasets and metrics, underscoring its beneficial impact, which provides only salient information to generative PLMs. We positively highlight that G-SEEK-2 consistently surpasses on average our previous approach, underscoring the importance of selecting the appropriate GNN based on the data at hand.

To assess the effectiveness of input compression achieved through graph-based processing, we conduct an experiment where we vary the number of input tokens provided to the generative PLM. Consequently, we choose different quantities of pertinent sentences to retain from the multi-document cluster. In Fig. 4, we present the results based on semantic evaluation using BERTScore, employing BART as the summarizer and MULTI-LEXSUM-TINY as the dataset, and using three decreasing input sizes. Remarkably, as the input size decreases, the model equipped with G-SEEK-2 experience less impact compared to vanilla solutions that rely on reading the initial truncated tokens of the documents, which may include information unrelated to the summary. The drop in BERTScore as the number of input tokens decreases is due to important input information being truncated, leading to less relevant content in the summary and negatively impacting the score.

VII. CONCLUSION

This work introduces G-SEEK-2, a graph-based approach for multi-document summarization in low-data settings. By constructing a heterogeneous graph and applying a tailored relevance-scoring algorithm, G-SEEK-2 identifies key sentences, improving summary quality and enhancing PLM learning with limited data. Future work will explore model interpretability [89]. While aiming to make knowledge more accessible for professionals—like lawyers [90], [91]—it is essential that G-Seek-2 upholds ethical standards, transparently supporting human expertise in critical fields such as law.

REFERENCES

[1] Z. Shen, K. Lo, L. Yu, N. Dahlberg, M. Schlanger, and D. Downey, "Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities," in *Proc. 2012 Adv. Neural Inf. Process. Syst. 35: Annu. Conf. Neural Inf. Process. Syst.*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., New Orleans, LA, USA, Nov./Dec. 2022, pp. 13158–13173. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/552ef803bef9368c29e53c167de34b55-Abstract-Datasets_and_Benchmarks.html

[2] D. G. Ghalandari, C. Hokamp, N. T. Pham, J. Glover, and G. Ifrim, "A large-scale multi-document summarization dataset from the wikipedia current events portal," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Assoc. Comput. Linguistics, Jul. 2020, pp. 1302–1308, doi: 10.18653/v1/2020.acl-main.120.

[3] E. Gesner, P. Gazarian, and P. C. Dykes, "The burden and burnout in documenting patient care: An integrative literature review," in *Proc. MED-INFO 2019: Health Wellbeing E-Netw. All - Proc. 17th World Congr. Med. Health Inform.* (Studies in Health Technology and Informatics Series), L. Ohno-Machado and B. Séroussi, Eds., Lyon, France: IOS Press, Aug. 2019, vol. 264, pp. 1194–1198, doi: 10.3233/SHTI190415.

[4] G. Frisoni and G. Moro, "Phenomena explanation from text: Unsupervised learning of interpretable and statistically significant knowledge," in *Proc. 2020 DATA* (Communications in Computer and Information Science Series), 2020, vol. 1446, pp. 293–318, doi: 10.1007/978-3-030-83014-4_14.

[5] L. Ragazzi, P. Italiani, G. Moro, and M. Panni, "What are you token about? differentiable perturbed top-k token selection for scientific document summarization," in *Proc. Findings Assoc. Comput. Linguistics, Bangkok, Thailand and Virtual Meeting*, L. Ku, A. Martins, and V. Srikumar, Eds. Assoc. Comput. Linguistics, Aug. 2024, pp. 9427–9440. [Online]. Available: https://aclanthology.org/2024.findings-acl.561

[6] C. Ma et al., "Multi-document summarization via deep learning techniques: A survey," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 102:1–102:37, 2023.

[7] G. Moro, L. Ragazzi, L. Valgimigli, and D. Freddi, "Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, S. P. M. Nakov and A. Villavicencio, Eds. Dublin, Ireland:Assoc. Comput. Linguistics, May 2022, pp. 180–189, doi: 10.18653/v1/2022.acl-long.15.

[8] M. Zhang, G. Zhou, N. Huang, P. He, W. Yu, and W. Liu, "AsU-OSum: Aspect-augmented unsupervised opinion summarization," *Inf. Process. Manag.*, vol. 60, no. 1, 2023, Art. no. 103138, doi: 10.1016/j.ipm.2022.103138.

[9] S. Lamsiyah, A. E. Mahdaouy, B. Espinasse, and S. E. A. Ouatik, "An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings," *Expert Syst. Appl.*, vol. 167, 2021, Art. no. 114152, doi: 10.1016/j.eswa.2020.114152.

[10] C. Shen, L. Cheng, X. Nguyen, Y. You, and L. Bing, "A hierarchical encoding-decoding scheme for abstractive multi-document summarization," in *Proc. Findings Assoc. Comput. Linguistics: EMNLP 2023*, H. J. B. Pino and K. Bali, Eds., Singapore: Assoc. Comput. Linguistics, Dec. 2023, pp. 5872–5887. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.391

[11] A. Vaswani et al., "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[12] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. 2018 Conf. Empirical Methods Natural Lang. Process. EMNLP 2018: Syst. Demonstrations*, E. Blanco and W. Lu, Eds. Assoc. Comput. Linguistics, Oct./Nov. 2018, pp. 66–71, doi: 10.18653/v1/d18-2012.

[13] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 2020 Assoc. Comput. Linguistics*, Jul. 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[14] Y. Song, Y. Chen, and H. Shuai, "Improving multi-document summarization through referenced flexible extraction with credit-awareness," in *Proc. 2022 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, M. Carpuat, M. de Marneffe, and I. V. M. Ruíz, Eds. Seattle, WA, USA: Assoc. Comput. Linguistics, Jul. 2022, pp. 1667–1681, doi: 10.18653/v1/2022.naacl-main.120.

[15] W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, "PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization," in *Proc. Assoc. Comput. Linguistics*, Dublin, Ireland, May 2022, pp. 5245–5263. [Online]. Available: https://aclanthology.org/2022.acl-long.360

[16] G. Moro, L. Ragazzi, L. Valgimigli, G. Frisoni, C. Sartori, and G. Marfia, "Efficient memory-enhanced transformer for long-document summarization in low-resource regimes," *Sensors*, vol. 23, no. 7, 2023, Art. no. 3542, doi: 10.3390/s23073542.

[17] T. Yu, Z. Liu, and P. Fung, "Adaptsum: Towards low-resource domain adaptation for abstractive summarization," in *Proc. NAACL-HLT*, 2021, pp. 5892–5904.

[18] G. Moro and L. Ragazzi, "Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes," in *Proc. 36th AAAI Conf. Artif. Intell., AAAI 2022, 34th Conf. Innov. Appl. Artif. Intell., IAAI 2022, 12th Symp. Educ. Adv. Artif. Intell.*, Feb./Mar. 2022, pp. 11085–11093, doi: 10.1609/aaai.v36i10.21357.

[19] M. A. Hedderich et al., "A survey on recent approaches for natural language processing in low-resource scenarios," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Jun. 2021, pp. 2545–2568. [Online]. Available: https://aclanthology.org/2021.naacl-main.201

[20] S. Tu, J. Yu, F. Zhu, J. Li, L. Hou, and J. Nie, "UPER: Boosting multi-document summarization with an unsupervised prompt-based extractor," in *Proc. 29th Int. Conf. Comput. Linguistics*, N. C. Calzolari, H. Huang, J. Kim, L. Pustejovsky, K. Wanner, P. Choi, H. Ryu, L. Chen, H. Donatelli, S. Ji, P. Kurohashi, N. Paggio, S. Xue, Y. Kim, Z. H. He, T. K. Lee, E. Santus, F. Bond, and S. Na, Eds. Gyeongju, South Korea: Int. Committee Comput. Linguistics, Oct. 2022, pp. 6315–6326. [Online]. Available: https://aclanthology.org/2022.coling-1.550

[21] M. F. Salchner and A. Jatowt, "A survey of automatic text summarization using graph neural networks," in *Proc. COLING*, 2022, pp. 6139–6150.

[22] R. Jia et al., "Neural extractive summarization with hierarchical attentive heterogeneous graph network," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 3622–3631.

[23] D. Wang et al., "Heterogeneous graph neural networks for extractive document summarization," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 6209–6219.

[24] M. Chen et al., "Sgsum: Transforming multi-document summarization into sub-graph selection," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4063–4074.

[25] X. Ji and W. Zhao, "SKGSUM: Abstractive document summarization with semantic knowledge graphs," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2021, pp. 1–8.

[26] Y. Qiu and S. B. Cohen, "Abstractive summarization guided by latent hierarchical document structure," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Abu Dhabi, U.A.E, Dec. 2022, pp. 5303–5317. [Online]. Available: https://aclanthology.org/2022.emnlp-main.355

[27] W. Wu et al., "BASS: Boosting abstractive summarization with unified semantic graph," in *Proc. Assoc. Comput. Linguistics/Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 6052–6067.

[28] G. Moro and L. Ragazzi, "Align-then-abstract representation learning for low-resource summarization," *Neurocomputing*, vol. 548, 2023, Art. no. 126356, doi: 10.1016/j.neucom.2023.126356.

[29] G. Moro, L. Ragazzi, and L. Valgimigli, "Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios," in *Proc. 26th Eur. Conf. Artif. Intell., 12th Conf. Prestigious Appl. Intell. Syst.* (Frontiers in Artificial Intelligence and Applications Series), K. A. Gal, G. Nowé, J. Nalepa, R. Fairstein, and R. Radulescu, Eds., Kraków, Poland, IOS Press, Sep. 2023, vol. 372, pp. 1747–1754, doi: 10.3233/FAIA230460.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, Apr. 2017, OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[31] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst.*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., Long, Beach, CA, USA, Dec. 2017, pp. 1024–1034. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[32] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, Apr. 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[33] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. 7th Int. Conf. Learn. Representations*, New Orleans, LA, USA, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km

[34] G. Li, M. Müller, A. K. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?," in *Proc. 2019 IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Korea South, Oct. 2019, pp. 9266–9275. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00936

[35] C. Zhang, J. Yu, Y. Song, and W. Cai, "Exploiting edge-oriented reasoning for 3D point-based scene graph analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 9705–9715. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_Exploiting_Edge-Oriented_Reasoning_for_3D_Point-Based_Scene_Graph_Analysis_CVPR_2021_paper.html

[36] A. R. Fabbri et al., "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proc. Assoc. Comput. Linguistics*, Florence, Italy, ACL, Jul. 2019, pp. 1074–1084, doi: 10.18653/v1/p19-1102.

[37] Y. Perez-Beltrachini, Y. Liu, and M. Lapata, "Generating summaries with topic templates and structured convolutional decoders," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Florence, Italy, Assoc. Comput. Linguistics, Jul./Aug. 2019, pp. 5107–5116, doi: 10.18653/v1/p19-1504.

[38] J. Parnell, I. J. Unanue, and M. Piccardi, "A multi-document coverage reward for relaxed multi-document summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland, Assoc. Comput. Linguistics, May 2022, pp. 5112–5128, doi: 10.18653/v1/2022.acl-long.351.

[39] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proc. 2020 Int. Conf. Mach. Learn.*, (Proceedings of Machine Learning Research Series), Jul. 2020, vol. 119, pp. 11328–11339. [Online]. Available: http://proceedings.mlr.press/v119/zhang20ae.html

[40] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.

[41] M. Trabelsi and H. Uzunalioglu, "Absformer: Transformer-based model for unsupervised multi-document abstractive summarization," in *Proc. Document Anal. Recognit. Workshops* (Lecture Notes in Computer Science Series), M. Coustaty and A. Fornés, Eds., San José, CA, USA, Springer, Aug. 2023, vol. 14194, pp. 151–166, doi: 10.1007/978-3-031-41501-2_11.

[42] P. J. Liu et al., "Generating wikipedia by summarizing long sequences," in *Proc. Int. Conf. Learn. Representations*, Vancouver, BC, Canada, Apr./May 2018. [Online]. Available: https://openreview.net/forum?id=Hyg0vbWC-

[43] Y. Zhang et al., "Summ^N: A multi-stage summarization framework for long input dialogues and documents," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland, Assoc. Comput. Linguistics, May 2022, pp. 1592–1604, doi: 10.18653/v1/2022.acl-long.112.

[44] Z. Mao et al., "DYLE: Dynamic latent extraction for abstractive long-input summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, Dublin, Ireland, May 2022, pp. 1687–1698, doi: 10.18653/v1/2022.acl-long.118.

[45] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, Philadelphia, Pennsylvania, USA: Assoc. Comput. Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[46] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, Barcelona, Spain, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[47] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics: Main Vol.*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Assoc. Comput. Linguistics, Apr. 2021, pp. 874–880, doi: 10.18653/v1/2021.eacl-main.74.

[48] X. Gu et al., "Generating representative headlines for news stories," in *Proc. WWW 2020: Taipei*, Taiwan, Apr. 2020, pp. 1773–1784, doi: 10.1145/3366423.3380247.

[49] C. Hokamp, D. G. Ghalandari, N. T. Pham, and J. Glover, "Dyne: Dynamic ensemble decoding for multi-document summarization," 2020, *arXiv:2006.08748*.

[50] G. Moro, L. Ragazzi, L. Valgimigli, and L. Molfetta, "Retrieve-and-rank end-to-end summarization of biomedical studies," in *Proc. Similarity Search Appl. 16th Int. Conf.*, (Lecture Notes in Computer Science Series), O. Pedreira and V. Estivill-Castro, Eds., Springer, Oct. 2023, vol. 14289, pp. 64–78, doi: 10.1007/978-3-031-46994-7_6.

[51] M. T. Nayeem, T. A. Fuad, and Y. Chali, "Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion," in *Proc. Coling*, Santa Fe, New Mexico, USA, ACL, Aug. 2018, pp. 1191–1204. [Online]. Available: https://aclanthology.org/C18-1102/

[52] D. Antognini and B. Faltings, "Learning to create sentence semantic relation graphs for multi-document summarization," 2019, *arXiv:1909.12231*. [Online]. Available: http://arxiv.org/abs/1909.12231
[53] W. Li et al., "Leveraging graph to improve abstractive multi-document summarization," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 6232–6243.
[54] R. K. Amplayo and M. Lapata, "Informative and controllable opinion summarization," in *Proc. Eur. Assoc. Comput. Linguistics*, Online, ACL, Apr. 2021, pp. 2662–2672. [Online]. Available: https://aclanthology.org/2021.eacl-main.229/
[55] Y. Liu and M. Lapata, "Hierarchical transformers for multi-document summarization," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 5070–5081.
[56] H. Jin, T. Wang, and X. Wan, "Multi-granularity interaction network for extractive and abstractive multi-document summarization," in *Proc. Assoc. Comput. Linguistics*, Jul. 2020, pp. 6244–6254, doi: 10.18653/v1/2020.acl-main.556.
[57] C. Zhu, R. Xu, M. Zeng, and X. Huang, "A hierarchical network for abstractive meeting summarization with cross-domain pretraining," in *Proc. Findings Assoc. Comput. Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds., Assoc. Comput. Linguistics, Nov. 2020, pp. 194–203, doi: 10.18653/v1/2020.findings-emnlp.19.
[58] S. Li and J. Xu, "HierMDS: A hierarchical multi-document summarization model with global-local document dependencies," *Neural Comput. Appl.*, vol. 35, no. 25, pp. 18553–18570, 2023, doi: 10.1007/s00521-023-08680-0.
[59] V. Nguyen, S. T. Mai, and M. Nguyen, "Learning to summarize multi-documents with local and global information," *Prog. Artif. Intell.*, vol. 12, no. 3, pp. 275–286, 2023, doi: 10.1007/s13748-023-00302-z.
[60] Z. Jia et al., "Improving the accuracy, scalability, and performance of graph neural networks with roc," in *Proc. Mach. Learn. Syst.*, mlsys.org, 2020, pp. 187–198.
[61] A. Hogan et al., "Knowledge graphs," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–37, 2021.
[62] D. Muratore, M. Hagenbuchner, F. Scarselli, and A. C. Tsoi, "Sentence extraction by graph neural networks," in *Proc. Int. Conf. Artif. Neural Netw.*, (Lecture Notes in Computer Science Series), 2010, vol. 6354, pp. 237–246.
[63] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. R. Radev, "Graph-based neural multi-document summarization," in *Proc. 21st Conf. Comput. Natural Lang. Learn.*, R. Levy and L. Specia, Eds. Vancouver, Canada, Assoc. Comput. Linguistics, Aug. 2017, pp. 452–462, doi: 10.18653/v1/K17-1045.
[64] X. Doan, L. Nguyen, and K. N. Bui, "Multi graph neural network for extractive long document summarization," in *Proc. Coling Int. Committee on Comput. Linguistics*, 2022, pp. 5870–5875.
[65] Y. Yin, L. Song, J. Su, J. Zeng, C. Zhou, and J. Luo, "Graph-based neural sentence ordering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, S. Kraus, Ed. Macao, China, ijcai.org, Aug. 2019, pp. 5387–5393, doi: 10.24963/ijcai.2019/748.
[66] R. Pasunuru et al., "Efficiently summarizing text and graph encodings of multi-document clusters," in *Proc. 2021 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, S. Cotterell, T. Chakraborty, and Y. Zhou, Eds. Assoc. Comput. Linguistics, Jun. 2021, pp. 4768–4779, doi: 10.18653/v1/2021.naacl-main.380.
[67] Z. Zhang, H. Elfardy, M. Dreyer, K. Small, H. Ji, and M. Bansal, "Enhancing multi-document summarization with cross-document graph-based information extraction," in *Proc. 17th Conf. Eur. Chapter Assoc. Comput. Linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia, Assoc. Comput. Linguistics, May 2023, pp. 1688–1699, doi: 10.18653/v1/2023.eacl-main.124.
[68] H. Wang, J. Chang, and J. Huang, "User intention-based document summarization on heterogeneous sentence networks," in *Proc. Database Syst. Adv. Appl. - 24th Int. Conf.*, (Lecture Notes in Computer Science Series), G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds., Chiang Mai, Thailand, Springer, Apr. 2019, vol. 11447, pp. 572–587, doi: 10.1007/978-3-030-18579-4_34.
[69] P. Cui and L. Hu, "Topic-guided abstractive multi-document summarization," in *Proc. 2021 Findings Assoc. Comput. Linguistics: EMNLP*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. Punta Cana, Assoc. Comput. Linguistics, Nov. 2021, pp. 1463–1472, doi: 10.18653/v1/2021.findings-emnlp.126.
[70] M. Lu, L. Liang, and G. Liu, "Parallel relationship graph to improve multi-document summarization," in *Proc. Artif. Neural Netw. Mach. Learn. 31st Int. Conf. Artif. Neural Netw.*, Sep. 2022, (Lecture Notes in Computer Science Series), E. Pimenidis, P. P. Angelov, C. Jayne, A. Papaleonidas, and M. Aydin, Eds., vol. 13530. Bristol, U.K., Springer, 2022, pp. 630–642, doi: 10.1007/978-3-031-15931-2_52.
[71] Y. Zhao et al., "Multi-granularity heterogeneous graph attention networks for extractive document summarization," *Neural Netw.*, vol. 155, pp. 340–347, 2022, doi: 10.1016/j.neunet.2022.08.021.
[72] M. Li, J. Qi, and J. H. Lau, "Compressed heterogeneous graph for abstractive multi-document summarization," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 13085–13093, doi: 10.1609/aaai.v37i11.26537.
[73] A. Bajaj et al., "Long document summarization in a low resource setting using pretrained language models," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.: Student Res. Workshop*, Aug. 2021, pp. 71–80. [Online]. Available: https://aclanthology.org/2021.acl-srw.7
[74] M. Grootendorst, "Keybert: Minimal keyword extraction with bert," 2020, doi: 10.5281/zenodo.4461265.
[75] A. Mathur and M. Suchithra, "Application of abstractive summarization in multiple choice question generation," in *Proc. 2022 IEEE Int. Conf. Comput. Intell. Sustain. Eng. Solutions*, 2022, pp. 409–413.
[76] N. Nikzad-Khasmakhi et al., "Phraseformer: Multimodal keyphrase extraction using transformer and graph embedding," 2021, *arXiv:2106.04939*.
[77] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
[78] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process./Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3980–3990.
[79] Y. Chen and H. Shuai, "Meta-transfer learning for low-resource abstractive summarization," in *Proc. 2021 AAAI Conf. Artif. Intell.*, AAAI Press, Feb. 2021, pp. 12692–12700. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17503
[80] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, New Orleans, Louisiana Jun. 2018, pp. 708–719. [Online]. Available: https://aclanthology.org/N18-1065
[81] L. Huang et al., "Efficient attentions for long document summarization," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.-Hum. Lang. Technol. Conf.*, Jun. 2021, pp. 1419–1436, doi: 10.18653/v1/2021.naacl-main.112.
[82] S. Bird, "NLTK: The natural language toolkit," in *Proc. Assoc. Comput. Linguistics*, 2006. [Online]. Available: https://aclanthology.org/P06-4018/
[83] G. Moro, L. Ragazzi, and L. Valgimigli, "Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy," in *Proc. 37th AAAI Conf. Artif. Intell., 35th Conf. Innov. Appl. Artif. Intell., 13th Symp. Educ. Adv. Artif. Intell.*, B. Williams, Y. Chen, and J. Neville, Eds. Washington, DC, USA, AAAI Press, Feb. 2023, pp. 14417–14425, doi: 10.1609/aaai.v37i12.26686.
[84] T. Zhang et al., "Bertscore: Evaluating text generation with BERT," in *Proc. 2020 Int. Conf. Learn. Representations*, Addis Ababa, Ethiopia, OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr
[85] T. Goyal, J. J. Li, and G. Durrett, "News summarization and evaluation in the era of GPT-3," 2022, *arXiv:arXiv.2209.12356*.
[86] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. R. McKeown, and T. B. Hashimoto, "Benchmarking large language models for news summarization," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 39–57, 2024, doi: 10.1162/tacl_a_00632.
[87] Z. Fu et al., "Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder," 2023, *arXiv:2304.04052*.
[88] T. Wolf et al., "Huggingface's transformers: State-of-the-art natural language processing," 2019, *arXiv:1910.03771*.
[89] G. Moro, L. Ragazzi, L. Valgimigli, F. Vincenzi, and D. Freddi, "Revelio: Interpretable long-form question answering," in *Proc. 2nd Tiny Papers Track At ICLR*, Vienna, Austria, OpenReview.net, May 2024. [Online]. Available: https://openreview.net/forum?id=fyvEJXsaQf
[90] G. Moro, N. Piscaglia, L. Ragazzi, and P. Italiani, "Multi-language transfer learning for low-resource legal case summarization," *Artif. Intell. Law*, vol. 32, pp. 1–29, 2023, doi: 10.1007/s10506-023-09373-8.
[91] L. Ragazzi, G. Moro, S. Guidi, and G. Frisoni, "Lawsuit: A large expert-written summarization dataset of italian constitutional court verdicts," *Artif. Intell. Law*, 2024, Art. no. 272530787. [Online]. Available: https://api.semanticscholar.org/CorpusID