ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230460

Graph-Based Abstractive Summarization of Extracted Essential Knowledge for Low-Resource Scenarios

Gianluca Moro[a;*], Luca Ragazzi[a] and Lorenzo Valgimigli[a]

[a]Department of Computer Science and Engineering (DISI), University of Bologna
Via dell'Università 50, I-47522 Cesena, Italy
ORCiD ID: Gianluca Moro https://orcid.org/0000-0002-3663-7877,
Luca Ragazzi https://orcid.org/0000-0003-3574-9962, Lorenzo Valgimigli https://orcid.org/0000-0003-0309-771X

Abstract. Although current summarization models can process increasingly long text sequences, they still struggle to capture salient related information spread across the lengthy size of inputs with few labeled training instances. Today's research still relies on standard input truncation without considering graph-based modeling of multiple semantic units to summarize only crucial facets. This paper proposes G-SEEK, a graph-based summarization of extracted essential knowledge. By representing the long source with a heterogeneous graph, our method extracts and provides salient sentences to an abstractive summarization model to generate the summary. Experimental results in low-resource scenarios, distinguished by data scarcity, reveal that G-SEEK consistently improves both the long- and multi-document summarization performance and accuracy across several datasets.

1 Introduction

The growing availability of unstructured information promotes text summarization solutions, which aim to generate concise synopses that convey the exact semantics of the source [69]. Recently, due to the latest advancements in natural language processing (NLP)—from information retrieval [13, 50, 51, 52] to entity relationships acquisition [14, 15], classification [10], and event extraction [16]—abstractive summarization [4, 12, 31] is receiving more interest, aiming to paraphrase the most meaningful details of documents instead of just retrieving them (i.e., extractive summarization). In this context, a particularly tough challenge is to process massive sequences, namely the task of long-input summarization. Two subtasks are long- and multi-document summarization (LDS and MDS, respectively). LDS [29] wants to capture and condense salient points scattered across a lengthy input (e.g., 10K words). In MDS [41], the synthesis is generated from a pool of sources related to the topic whose concatenation assembles a single long input to enable the summarization task to be addressed as in LDS [68]. Thus, from now on, we refer to "the long input" as the source of both LDS and MDS tasks.

State-of-the-art (SOTA) solutions for long-input summarization are built upon transformers [63], which proficiently capture long-distance relations between words[1] with the self-attention mechanism. Nevertheless, such models are denoted by a structural constraint that proportionally links their memory usage to the input size, making them unduly resource-demanding when processing long texts. Consequently, this problem poses complications in LDS and MDS due to the long and intricate sources, which is even more noticeable in real-world low-resource scenarios [45, 48] distinguished by the lack of labeled instances to supervise model training [70]. In fact, in realistic small and medium organizations, producing the gold summary of lengthy documents is costly, time-consuming, and may require domain knowledge specialists. For this reason, low-resource summarization is an important research topic that deserves more attention from the NLP community [21].

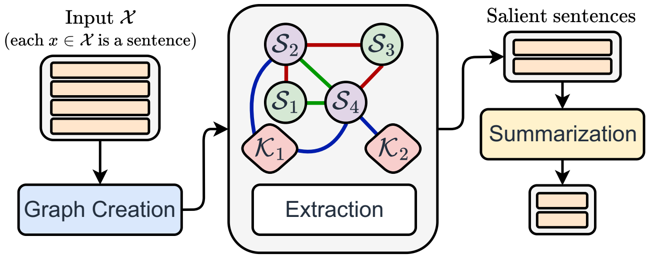

Figure 1. Overview of our approach. A long input $\mathcal{X}$ (each $x \in \mathcal{X}$ is a sentence) is converted into a heterogeneous graph: the pink diamonds represent the keyword nodes $\mathcal{K}$, the violet circles denote the sentence nodes $\mathcal{S}$ containing keywords, the green circles indicate the context of $\mathcal{S}$, and the different segments between nodes symbolize edges. The salient sentences are extracted and given to a generative PLM to produce the summary.

A promising approach to mitigate these obstacles is to use a semantic graph to represent the long input. Intuitively, by aggregating all source information, summary-worthy sentences (i.e., essential knowledge) can be pinpointed and extracted, avoiding standard input truncation and giving models more high-quality training samples that enable them to learn faster in data scarcity conditions. Previous contributions used graph representations for text summarization [60]. However, their methods are proposed for extractive summarization [7, 26, 65] and short texts [25, 58] or do not take advantage of SOTA generative pre-trained language models (PLMs) [67].

To fill this gap, we introduce G-SEEK (Figure 1),[2] a graph-based summarization of extracted essential knowledge. By representing

[*] Corresponding Author. Email: gianluca.moro@unibo.it
[1] Technically, these are subwords yielded by a subword tokenizer [30].
[2] https://disi-unibo-nlp.github.io/projects/gseek/

the long source with a heterogeneous graph, G-Seek extracts and gives the more relevant sentences to an abstractive summarizer for synthesis generation, avoiding feeding models only until their maximum input size, which prevents them from reading the overpart of the source, causing quality degradation [44]. Technically, our graph employs a lightweight PLM to embody different semantic units (i.e., sentences and keywords) whose relationships are handled through multiple informative edges; then, a graph attention network (GAT) [64] is trained to select the salient nodes, soft labeled with a heuristic. Our solution is proposed explicitly for low-resource environments with a few dozen labeled training instances for the following motivations: (i) we show that giving more highly correlated source-target samples helps PLMs generate better summaries in a data scarcity scenario; (ii) the small number of trainable parameters of our solution (4M) lets G-Seek not overfit over a few examples.

Extensive experimentations with multiple public datasets reveal that G-Seek enhances the performance of SOTA summarization models in low-resource settings. Our contributions are as follows:

- We propose G-Seek, a new low-resource long-input summarization approach that pinpoints and gives key sentences to generative PLMs by modeling the lengthy source with a semantic graph.
- We introduce a heterogeneous graph that captures the diversity of semantic units and their relationships through multiple edges.
- Experimental results in LDS and MDS datasets show that SOTA abstractive summarization models with G-Seek improve syntactic and semantic evaluation metrics in a data scarcity scenario.

2 Related Work

Generative Pre-trained Language Models. Recent generative PLMs have shown strong performance and adaptation to LDS and MDS tasks. These models are based on the transformer encoder-decoder architecture [63] characterized by layers of self-attention. To be precise, quadratic PLMs such as Bart [31] and Pegasus [71] can process up to 1024 tokens due to the quadratic complexity w.r.t. the input size. Differently, linear PLMs such as Led [4] and Primera [68] can read more input (up to 4096 tokens with 24 GB of GPU memory), making them more suitable for long-input summarization. Although their impressive performance, such linear transformers, like quadratic ones, still rely on input truncation, namely processing the source until the model's maximum input size, ignoring potentially relevant summary-worthy details.

Graph-based Summarization. Graphs and graph neural networks (GNNs) have played a thrilling role in MDS [60], adding scalability [27] and better domain modeling [22] to mitigate the flaws of transformer-based models. Several contributions employ GNNs as standalone solutions [53], where the summary is the composition of sentences extracted from the input [9]. Recent solutions are Heter-SumGraph [65], which leverages a multilevel node representation, HAHSum [26], which exploits NER entities, and SgSum [7], which extracts subgraphs to generate the summary. On the contrary, GNNs can be embedded with abstractive summarization models to improve performance [34]. Bass [67] introduces a unified semantic graph to represent the group of texts and modifies the transformer architecture to interact with the graph. SKGSum [25] exploits nodes of different levels to guide the summary generation. Finally, HGSum [33] extends a linear transformer by incorporating a heterogeneous graph, training them jointly at the expense of a costly pipeline.

Algorithm 1 Soft Labeling

```
Input:
$\mathcal{X} = \{x_1, \dots, x_x\}$                ▷ Input sentences
$\mathcal{Y} = \{y_1, \dots, y_y\}$                ▷ Output sentences
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 $\mathcal{S}$
```

Other Approaches. Two-stage methods [17, 32, 38, 39] utilize various strategies to score and rank documents before generating the summary. Hierarchical solutions combine document relations to obtain semantic rich representations by leveraging graph-based techniques [1, 2, 34, 36, 54], multihead grouping and inter-paragraph attention [28, 39], and maximal marginal relevance [11]. Marginalization-based techniques [20, 23, 47, 49] apply marginalization during decoding to produce a single output from many inputs.

Although the effectiveness of all these different methods is similar, generative PLMs are still the SOTA approaches for long-input summarization due to (i) fast domain adaptation with few training data and (ii) more scalability and efficiency in computational resources.

3 Method

In this section, we describe our novel *graph-based summarization of extracted essential knowledge* (G-Seek) in detail. In a nutshell, G-Seek comprises a document-to-graph module (blended with a GAT) and a pre-trained abstractive summarization model to generate a summary by focusing only on the essential input sentences. Section 3.1 explains the necessity of labeling the relevance of input sentences (i.e., soft labeling). Section 3.2 details the passages needed to construct the graph from the long input using semantic and structural information. Section 3.3 illustrates the GAT module devised to learn the relationships between graph nodes. Finally, Section 3.4 sums up the overall summarization pipeline of our proposed approach.

3.1 Soft Labeling

Recent advances in NLP have highlighted the need to identify summary-worthy sentences in the source to address the problem of processing large amounts of information [3, 42]. Intuitively, each sentence can be labeled as relevant or not for the target summary to train a model to recognize such salient sentences. Regardless, because of the lack of a ground-truth relevancy label, we need to heuristically mark the salience of the input sentences w.r.t. the gold summary, namely performing a soft labeling strategy. Concretely, let $\mathcal{X} = \{x_1, \dots, x_x\}$ and $\mathcal{Y} = \{y_1, \dots, y_y\}$ be the long input and the corresponding summary, respectively, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is a sentence. For each x_i, we produce a relevancy score $\in [0, 1]$ by computing the similarity between each y_i (and then taking the best value), using a greedy algorithm (Algorithm 1). We test different evaluation metrics (i.e., BLEU [56] and ROUGE [37]) as similarity functions on Multi-LexSum (short) [62], using 100 samples for training and validation sets. We employ Primera [68] as

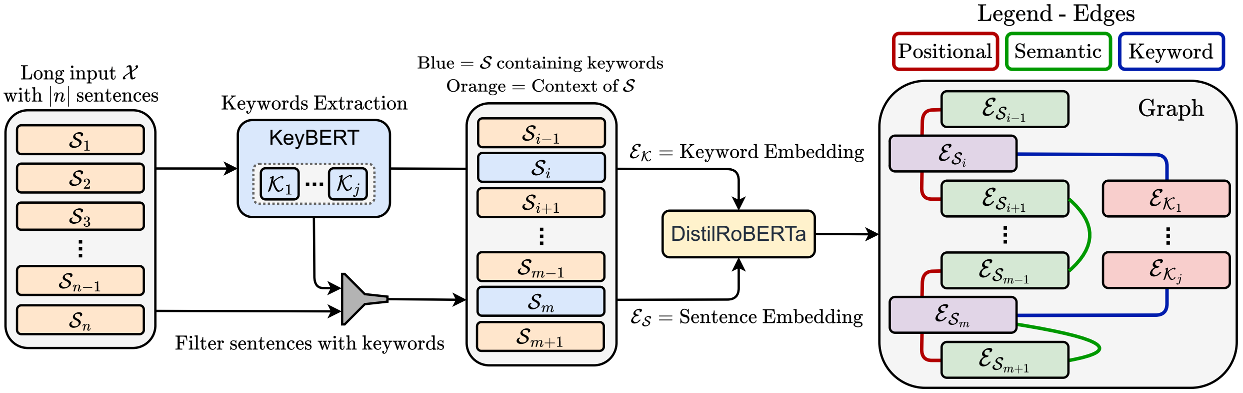
Long input $\mathcal{X}$
with $|n|$ sentences
$\mathcal{S}_1$
$\mathcal{S}_2$
$\mathcal{S}_3$
$\vdots$
$\mathcal{S}_{n-1}$
$\mathcal{S}_n$
Keywords Extraction
KeyBERT
$\mathcal{K}_1$ ⋯ $\mathcal{K}_j$
Filter sentences with keywords
Blue = $\mathcal{S}$ containing keywords
Orange = Context of $\mathcal{S}$
$\mathcal{S}_{i-1}$
$\mathcal{S}_i$
$\mathcal{S}_{i+1}$
$\vdots$
$\mathcal{S}_{m-1}$
$\mathcal{S}_m$
$\mathcal{S}_{m+1}$
$\mathcal{E}_{\mathcal{K}}$ = Keyword Embedding
DistilRoBERTa
$\mathcal{E}_{\mathcal{S}}$ = Sentence Embedding
Legend - Edges
Positional
Semantic
Keyword
Graph
$\mathcal{E}_{\mathcal{S}_{i-1}}$
$\mathcal{E}_{\mathcal{S}_i}$
$\mathcal{E}_{\mathcal{S}_{i+1}}$
$\vdots$
$\mathcal{E}_{\mathcal{S}_{m-1}}$
$\mathcal{E}_{\mathcal{S}_m}$
$\mathcal{E}_{\mathcal{S}_{m+1}}$
$\mathcal{E}_{\mathcal{K}_1}$
$\vdots$
$\mathcal{E}_{\mathcal{K}_j}$

Figure 2. The pipeline adopted by G-SEEK to produce a semantic heterogeneous graph from a long input. The document is given to KEYBERT, which creates a set of unique keywords. Then (i) the sentences containing at least one keyword, (ii) their context (sentences immediately next or before), (iii) and keywords are turned into embeddings using DISTILROBERTA, becoming the new graph nodes linked with different meaningful edges.

the backbone summarization model, which is a transformer with linear complexity in the input length pre-trained with an MDS-specific objective. Technically, after assigning a summary-relevance score to each $x_i \in \mathcal{X}$, we give PRIMERA only the sentences (in the order of occurrence in the source) with the highest scores until the maximum input size of the model (i.e., 4096 tokens). Table 1 indicates that the best metric for soft labeling is ROUGE-2 F1, so we use such a metric to label the relevancy of sentences. We denote that Table 1 reports the results by applying the soft labeling also on the validation set to investigate upper-bound performances (i.e., we simulate an oracle by accessing the ground-truth target summaries).

Table 1. The MDS results on MULTI-LEXSUM (SHORT). We give PRIMERA different inputs obtained by using diverse metrics as similarity functions for soft labeling.

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

3.2 *Document-to-Graph*

The document-to-graph module aims to create a semantic heterogeneous graph with the following steps (Figure 2):

- **Keyword Extraction**: we first remove the English stop words and general domain-specific terms that occur in more than 40% of the input (in each MDS document). Then we employ KEYBERT [18, 43], a lightweight method compared to more resource-demanding solutions [55], to select up to k keywords. For MDS, we extract k keywords for each document in the cluster and combine these keyword lists into a unique set, dropping duplicates.
- **Sentence Filtering**: we split the input into sentences and select those with at least one keyword. Additionally, we pick n sentences appearing before and after the selected one as context. For MDS, we define $\{[x_1^1, \ldots, x_x^1], \ldots, [x_1^z, \ldots, x_x^z]\}$ as the cluster of sources z. Let x_b^1 and x_e^1 be two sentences with keywords. We then select $[x_{b-n}^1, \ldots, x_{b+n}^1, x_{e-n}^1, \ldots, x_{e+n}^1]$, where $< x_{b-n}^1, \ldots, x_{b+n}^1 > -\{x_b\}$ is the context of x_b.
- **Sentence & Keyword Embedding**: we produce the embedding of all keywords and key sentences using DISTILROBERTA [61], a frozen distilled PLM characterized by a few parameters (82M) that let our solution be efficient in terms of GPU memory computation and occupation. Specifically, this model is already pre-trained to create sentence embeddings by using a self-supervised contrastive learning objective.[3] About creating sentence embeddings, the model yields one representation for each token within a sentence. Then, following [59], we average the token embeddings using mean pooling, generating the final vector e_i^x.
- **Graph Creation**: all keyword (KE) and sentence embeddings (SE) become the nodes of our graph. Inspired by [67], we use KE as supernodes, which means that all sentences that hold a keyword have a bidirectional *Keyword Edge* (blue lines in Figure 2) with the keyword node instead of among them. Then, we add bidirectional *Positional Edges* (red lines in Figure 2) between two SE if they appear consecutively in the source. Finally, similarly to [7, 34], we add *Semantic Edges* (green lines in Figure 2) between e_i and e_j if their cosine similarity is greater than a threshold t. The graph has as many nodes as the number of sentences and keywords.

3.3 *Graph Attention Network*

We operate a GAT to learn the relationships between the nodes in the graph (*structural information*) along with the information in their nodes (*semantic information*). By considering the edges connecting the nodes, a GAT can learn to propagate information across sentences, better understanding each sentence's context and meaning. In addition, GATs can efficiently handle enormous and complex graphs. In this study, we used the GAT to identify the most critical sentences by assigning an unbounded positive relevancy score to each node.

Technically, this module has the following layers:

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

Table 2. Statistics of the evaluation datasets including size, number of source documents per instance, number of total words in source and target texts, and source-target coverage, density, and compression ratio of words [19]. Except for the number of samples, all reported values are averaged across all instances.

		Source			Target		Source → Target		
Dataset	**Samples**	Docs	Words	Sents	Words	Sents	Coverage	Density	Compress
MDS (MULTI-LEXSUM)									
TINY	1603	10.7	119072.6	5962.5	24.7	1.4	0.92	2.27	5449.6
SHORT	3138	10.3	99378.2	5017.0	130.2	5.1	0.96	3.33	840.7
LONG	4534	8.8	75543.2	3814.2	646.5	28.8	0.94	4.07	97.4
LDS									
GOVREPORT	19,463	1	8765.0	298.7	556.3	18.1	0.94	9.08	17.9

- **Reprojection Layer** consists of two linear feed-forward layers (FFL) that learn how to reproject (x') the node embeddings x in the vector space. Mechanically, it expands the dimension d_x of the input embeddings n ($\mathbb{R}^{n\times768}$) of a factor called the *Boom Factor* (BF), inspired by the transformer architecture [63].

$$x' = \text{FFL}_\sigma(\text{FFL}_\gamma(x, d_x \cdot \text{BF}), d_x) \tag{1}$$

where σ and γ are learnable parameters of different linear layers.

- **GAT Layer** [64] exploits structural information to enrich node semantics. The output $\widehat{x}$ is generated by interpolating all rows of x' weighed by a score a ($\mathbb{R}^{n\times768}$).[4]

$$\widehat{x}_{i,j} = a(\mathbf{W}x'_i, \mathbf{W}x'_j) \tag{2}$$

- **Scoring Layer** comprises two linear layers to reduce the size of each node ($\widehat{x}$) to a unique real number (s) used as the relevancy score of the sentence associated with the node ($\mathbb{R}^n$).

$$s = \text{FFL}_\beta(\text{FFL}_\theta(\widehat{x}, d_{\widehat{x}} \cdot \text{BF}), 1) \tag{3}$$

To supervise the GAT model, we take advantage of the soft labels (l) explained in Section 3.1. Technically, we train the model to produce scores (s) that minimize the following loss:

$$\mathcal{L}_{\text{mse}} = (s - l)^2 \tag{4}$$

3.4 Summarization Pipeline

Once the input is turned into a graph and the GAT module assigns a score to each sentence, we extract the most relevant and use a generative PLM to generate the output summary. Technically, according to their relevancy scores, we select the most salient sentences and create an input text for the model containing fewer tokens than its maximum input size. As a result, the new input comprises sentences in the order of occurrence in the raw textual source.

We train the summarizer using the standard cross-entropy loss, which requires the model to predict the next token w_i of the target $\mathcal{Y}$ given $\mathcal{X}$ and the previous target tokens $w_{1:i-1}$, as follows:

$$\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{|\mathcal{Y}|} \log p_\tau(w_i | w_{1:i-1}, \mathcal{X}) \tag{5}$$

where τ indicates the model parameters and p is the predicted probability over the vocabulary.

Note: G-SEEK *is not jointly trained with the summarization model, resulting in a non-overwhelming training process. Thus, the pipeline of our overall solution is designed to work with a few labeled samples.*

4 Experiments

We focus our investigation on low-resource summarization, a real-world scenario distinguished by the scarcity of data due to the high cost of labeling. Following previous work [8, 44], we take the first 100/10/100 samples from all datasets' training, validation, and test sets without performing additional data pre-processing.

Table 3. The number of trainable parameters of generative PLMs and their maximum input size. Each URL starts with "https://huggingface.co/." G-SEEK uses the max input length of the downstream model but provides salient sentences instead of truncating the exceeding ones.

	URL	**#Params**	**Input**
Models			
BART-B	facebook/bart-base	140M	1024
BART-L	facebook/bart-large	400M	1024
PEGASUS-L	google/pegasus-large	568M	1024
LED-L	allenai/led-large-16384	459M	4096
PRIMERA-L	allenai/PRIMERA	447M	4096
G-SEEK	-	4M	-

4.1 Experimental Setup

Datasets. We contemplate the **MULTI-LEXSUM** dataset [62] as the evaluation benchmark for MDS, which gathers real-world federal civil rights lawsuits with expert-authored summaries. The main challenge of MULTI-LEXSUM is the long size of the source documents and the different granularity of the summaries (i.e., tiny, short, and long). Due to this multi-target nature, we overall experiment with three distinct dataset renditions as testbeds. Regarding LDS, we consider **GOVREPORT** [24], comprising 19K U.S. government reports. Table 2 provides key measurements of the datasets.[5]

Baselines. We compare with cutting-edge abstractive summarization models. **BART** [31] is a transformer with a quadratic memory and time complexity in the input length; we use the base and large checkpoints. **PEGASUS** [71] is a quadratic transformer pre-trained with a summarization-specific objective to predict gap sentences as a pseudo summary; we utilize the large checkpoint. **LED** [4] is a transformer with a linear memory complexity thanks to a sparse attention mechanism; we employ the large checkpoint. **PRIMERA** [68] is a linear transformer with the same architecture as LED but with an

[4] Please refer to the original paper [64] for further information.

[5] All datasets are publicly available in Hugging Face: https://huggingface.co/datasets/allenai/multi_lexsum (MULTI-LEXSUM) and https://huggingface.co/datasets/ccdv/govreport-summarization (GOVREPORT).

Table 4. Evaluation F1 scores on the benchmarked Multi-LexSum (Tiny, Short, Long) and GovReport datasets. b and l denote base and large, respectively. The best score for each model is in bold. † means statistically significant results of G-Seek (p-value < 0.05 with student t-test).

	Multi-LexSum (Tiny)					Multi-LexSum (Short)					Multi-LexSum (Long)					GovReport				
Model	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**	**R-1**	**R-2**	**R-L**	$\mathcal{R}$	**BS**
Quadratic																				
Bart-b	18.49	**6.39**	16.10	13.62	72.70	39.20	**17.52**	35.20	**30.37**	79.41	41.22	15.53	39.12	31.53	78.72	47.26	13.58	44.27	34.24	80.61
w/ G-Seek	**20.85**†	5.82	**17.14**†	**14.54**†	**74.20**†	**40.45**†	15.34	**35.31**	30.01	**79.84**†	**42.28**†	**16.36**†	**40.08**†	**32.46**†	**79.62**†	**47.65**†	**14.96**†	**44.38**	**34.91**†	**81.18**†
Bart-l	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**†
Pegasus-l	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**†
Linear																				
Led-l	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**†
Primera-l	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**†

MDS-specific pretraining objective to generate pseudo summaries, which are text spans automatically extracted based on the entity salience; we use the large checkpoint. Technically, in MDS, we use the standard approach of concatenating documents from the same cluster to form a single long textual input (we add the special token `<doc-sep>` to separate documents [68]). Note: we denote the base and large versions by b and l, respectively. Table 3 reports the number of parameters and maximum input size of the models.[6]

Metrics. We embrace the conventional ROUGE-{1,2,L}[7] F1 [37] and BERTScore F1 (BS) [72] to calculate the syntactic and semantic overlap, respectively, between the inferred and ground truth summaries. Moreover, we tally $\mathcal{R} = {}^{\text{avg}(r_1, r_2, r_L)}/_{1+\sigma_r^2}$ [46] to also consider an aggregated judgment, where σ_r^2 is the variance of the average ROUGE scores that penalizes generations with heterogeneous results across dimensions. All metrics $\in [0, 1]$ (%); the higher, the better.

Table 5. Results of different graph settings on the validation set of Multi-LexSum (Short) evaluated with precision, recall, and f-measure.

	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

Implementation Details. We fine-tune the models based on the PyTorch [57] implementations of the HuggingFace library [66], setting the seed to 42 for reproducibility. All experiments are run on an internal workstation with a Nvidia RTX 3090 GPU of 24 GB memory, 64 GB of RAM, and an Intel(R) Core(TM) i9-10900X CPU @3.70GHz processor. Regarding the GAT module of G-Seek, the training lasted 75 epochs with a learning rate of 5e-5, using AdamW as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$.[8] About the summarization task, all models are trained for 5 epochs with a learning rate of 3e-5, using mixed precision and gradient checkpointing to preserve memory. For decoding, we operate beam search with 5 beams and n-gram repetition blocks for n>5, utilizing the following (min-max) summary length: Multi-LexSum's Tiny (10-50), Short (50-150), Long (350-750), and GovReport (500-1000).

4.2 Overall Results

We train and evaluate all models on the benchmark datasets with and without G-Seek to highlight our contribution. Table 4 reports the performance of the systems in MDS and LDS. In particular, G-Seek consistently improves model performance across datasets and metrics, showing an ameliorative contribution of our method that gives only salient information to generative PLMs.

Table 6. The results of the GAT module on the validation set of Multi-LexSum (Short) under different settings with 30 training epochs. The final GAT is the best setting and checkpoint after 75 epochs.

	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

	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 GAT			
	38.37	45.49	41.63

4.3 Graph — Experiments

We investigate different graph settings. In particular, we focus on the *Sentence Filtering* module, examining the precision and recall of all labeled sentences among the selected salient ones. Table 5 reports the results, experimenting over the validation set of Multi-LexSum (Short) with the following facets:

- **Keywords**. We study the maximum number of keywords extracted by KeyBERT for each document in the cluster, establishing that 5 achieves the best results.
- **Context**. We examine a different number of consecutive sequences selected as the context of the salient sentence, discovering that 4 leads to better results.

With our hardware, the average time to create the graph for a single long input of ≈ 100K words is about 34 seconds. We denote that the current implementation does not adopt a specific optimization.

[6] The maximum input length depends on the architecture of the models' encoder, whereas the output size hinges on the dataset.

[7] We use the summary-level R-L, where each summary is split into sentences.

[8] Since all evaluation corpora are legal datasets, we trained the GAT module once on Multi-LexSum (Short) using soft labels (Section 3.1).

4.4 GAT — Experiments

We experiment with our GAT module under various settings in the validation set of MULTI-LEXSUM (SHORT). Specifically, we train the GAT for 30 epochs and evaluate by picking the top 100 sentences according to the assigned score. We use these 100 items to compute precision, recall, and f-measure. Table 6 reports the results, where we test the following:

- **Boom Factor**. We explore the contribution of the Boom Factor in the *Reprojection Layer*, finding that 2 is the best value.
- **Layers**. We employ various layers, surprisingly uncovering that 1 is the best option. Therefore, the small pool of training examples favors a lightweight solution with few trainable parameters.
- **Cosine Similarity**. We investigate the threshold to create *Semantic Edges* between nodes, deciding that 0.86 is the best choice.

Table 6 further shows the results of an optimal GAT trained for 100 epochs. We then select the best model checkpoint, which is after 75 epochs. The average time for each epoch is about 60 s.

Note: after numerous experiments, we tested the module by altering each hyperparameter while keeping the other two constants, using their best values (highlighted in bold in Table 6).

4.5 Analysis on Input Quality

Table 7 shows a qualitative example in MULTI-LEXSUM (SHORT) of our G-SEEK approach with respect to the input to give to summarization models. Table 8 reveals the higher source-target correlation using G-SEEK that creates an input composed of more key sentences. Specifically, we compare our method to the standard input truncation approach on the GOVREPORT's test set by computing ROUGE, BERTScore (also providing precision and recall scores), and unique keyword occurrence (dubbed UKO) between the inputs to give to models and their corresponding gold summaries. About UKO, we count the % of unique keywords (i.e., all words that are not stopwords) in the target that also appear in the source.

5 Conclusion

We propose G-SEEK, a novel graph-based method to extract and provide essential knowledge from massive textual information to abstractive summarization models for synthesis generation. G-SEEK represents a long input with a heterogeneous graph and models its semantics through different units (i.e., sentences and keywords) to pinpoint only the salient sentences. Experimental results in a data scarcity scenario over multiple public LDS and MDS datasets show that G-SEEK significantly improves the performance of syntactic and semantic metrics of state-of-the-art summarization systems. Furthermore, we show the contribution of G-SEEK to yield more correlated source-target pairs that enable generative PLMs to learn faster with few labeled training instances.

Future works should explore lightweight end-to-end pipelines (to train the GAT with the generative PLM jointly) and the contribution of the number of salient sentences to the final summary generation. Furthermore, as presented for communication networks [5, 6, 40], tracking and propagating knowledge refinements in sentences could be crucial when modeling long texts with graphs.

Table 7. A random qualitative input example of G-SEEK against a truncation-based approach on MULTI-LEXSUM (SHORT).

Golden Summary

Pursuant to the Civil Rights of Institutionalized Persons Act ("CRIPA"), 42 U.S.C. § 1997, the Civil Rights Division of the U.S. Department of Justice ("DOJ") conducted an investigation of conditions at the Mercer County Geriatric Center ("MCGC"), a public nursing home facility in New Jersey, evidently operated by Mercer County. The investigation led the DOJ to find that certain conditions at MCGC violated residents federal rights. The parties settled and the case is now closed.

Truncation-based Input

[. . . Text . . .]

The Attorney General files this complaint on behalf of the United States of America pursuant to the Civil Rights of Institutionalized Persons Act, 42 U.S.C. § 1997, to enjoin the named Defendants from depriving residents housed in the Mercer County Geriatric Center (MCGC) of rights, privileges, or immunities secured and protected by the Constitution and laws of the United States.

[. . . Text . . .]

G-SEEK Input

Administrator ECROYD is sued in his official capacity. 11. Mercer County receives federal Medicare and Medicaid funds for care provided at MCGC. 15.Defendants and MCGC are public entit(ies)" under the ADA and implementing regulations. 18.Defendant MERCER COUNTY is the entity charged by the laws of the State of New Jersey with authority to operate the MCGC and is responsible for the living conditions and health and safety of persons living in MCGC. 8. Through their acts and omissions, Defendants have failed to provide "care for its residents in such matter and in such an environment as will promote maintenance or enhancement of the quality of life of each resident, and have further failed to provide "the necessary care and services to attain or maintain the highest practicable physical, mental, and psychosocial well.

[. . . Text . . .]

If that resident was a candidate for services at home or in another community setting, this failure to accommodate a disability effectively resulted in the lengthy and improper segregation of the resident from society.

Table 8. The correlation between source-target pairs on the test set of the GOVREPORT dataset.

Approach	R-1$_{f1}$	R-2$_{f1}$	R-L$_{f1}$	BS$_p$	BS$_r$	BS$_{f1}$	UKO
TRUNCATION	31.47	11.73	30.28	69.47	72.24	70.81	67
G-SEEK	**33.42**	**12.88**	**33.19**	**73.93**	**75.86**	**74.87**	**71**

Ethics Statement

Because of the high societal impact of the legislation and biases in current generative PLMs [35], experts should determine the quality of the predicted summaries to make the presented solution work in real-world applications. All datasets and models used in this study were already publicly available.

Acknowledgements

This research is partially supported by (i) the Complementary National Plan PNC-I.1, "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, DARE—DigitAl lifelong pRevEntion initiative, code PNC0000002, CUP B53C22006450001, (ii) the PNRR, M4C2, FAIR—Future Artificial Intelligence Research, Spoke 8 "Pervasive AI," funded by the European Commission under the NextGeneration EU program. The authors thank the Maggioli Group[9] for granting the Ph.D. scholarship to Luca Ragazzi and Lorenzo Valgimigli.

References

[1] Reinald Kim Amplayo and Mirella Lapata, 'Informative and controllable opinion summarization', in *EACL, Online, April 19-23 2021*, pp. 2662–2672. ACL, (2021).

[2] Diego Antognini and Boi Faltings, 'Learning to create sentence semantic relation graphs for multi-document summarization', *CoRR*, **abs/1909.12231**, (2019).

[3] Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, et al., 'Long document summarization in a low resource setting using pretrained language models', in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pp. 71–80, Online, (August 2021). ACL.

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan, 'Longformer: The long-document transformer', *CoRR*, **abs/2004.05150**, (2020).

[5] Walter Cerroni, Gianluca Moro, Roberto Pasolini, and Marco Ramilli, 'Decentralized detection of network attacks through P2P data clustering of SNMP data', *Comput. Secur.*, **52**, 1–16, (2015).

[6] Walter Cerroni, Gianluca Moro, Tommaso Pirini, and Marco Ramilli, 'Peer-to-peer data mining classifiers for decentralized detection of network attacks', in *ADC*, volume 137 of *CRPIT*, pp. 101–108. ACS, (2013).

[7] Moye Chen, Wei Li, Jiachen Liu, Xinyan Xiao, et al., 'Sgsum: Transforming multi-document summarization into sub-graph selection', in *EMNLP (1)*, pp. 4063–4074. ACL, (2021).

[8] Yi-Syuan Chen and Hong-Han Shuai, 'Meta-transfer learning for low-resource abstractive summarization', in *AAAI 2021, Virtual Event, February 2-9, 2021*, pp. 12692–12700. AAAI Press, (2021).

[9] Xuan-Dung Doan, Le-Minh Nguyen, and Khac-Hoai Nam Bui, 'Multi graph neural network for extractive long document summarization', in *COLING*, pp. 5870–5875. International Committee on Computational Linguistics, (2022).

[10] Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, and Roberto Pasolini, 'On deep learning in cross-domain sentiment classification', in *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - (Volume 1), Funchal, Madeira, Portugal, November 1-3, 2017*, pp. 50–60. SciTePress, (2017).

[11] Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, et al., 'Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model', in *ACL, Florence, Italy, July 28- August 2 2019*, pp. 1074–1084. ACL, (2019).

[12] Giacomo Frisoni, Paolo Italiani, Stefano Salvatori, and Gianluca Moro, 'Cogito Ergo $Summ$: Abstractive Summarization of Biomedical Papers via Semantic Parsing Graphs and Consistency Rewards', in *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, February 7-14, 2023*, Washington, DC, USA, (February 2023). AAAI Press.

[13] Giacomo Frisoni, Miki Mizutani, Gianluca Moro, and Lorenzo Valgimigli, 'Bioreader: a retrieval-enhanced text-to-text transformer for biomedical literature', in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5770–5793, Abu Dhabi, United Arab Emirates, (December 2022). Association for Computational Linguistics.

[14] Giacomo Frisoni and Gianluca Moro, 'Phenomena Explanation from Text: Unsupervised Learning of Interpretable and Statistically Significant Knowledge', in *DATA*, volume 1446, pp. 293–318. Springer, (2020).

[15] Giacomo Frisoni, Gianluca Moro, and Antonella Carbonaro, 'Learning Interpretable and Statistically Significant Knowledge from Unlabeled Corpora of Social Text Messages: A Novel Methodology of Descriptive Text Mining', in *DATA*, pp. 121–134. SciTePress, (2020).

[16] Giacomo Frisoni, Gianluca Moro, and Antonella Carbonaro, 'A survey on event extraction for natural language understanding: Riding the biomedical literature wave', *IEEE Access*, **9**, 160721–160757, (2021).

[17] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush, 'Bottom-up abstractive summarization', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109, Brussels, Belgium, (October-November 2018). ACL.

[18] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[19] Max Grusky, Mor Naaman, and Yoav Artzi, 'Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies', in *NAACL, Volume 1 (Long Papers)*, pp. 708–719, New Orleans, Louisiana, (June 2018). Association for Computational Linguistics.

[20] Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, et al., 'Generating representative headlines for news stories', in *WWW 2020: Taipei, Taiwan, April 20-24, 2020*, pp. 1773–1784. ACM / IW3C2, (2020).

[21] Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, et al., 'A survey on recent approaches for natural language processing in low-resource scenarios', in *NAACL*, pp. 2545–2568, Online, (June 2021). ACL.

[22] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, et al., 'Knowledge graphs', *ACM Computing Surveys (CSUR)*, **54**(4), 1–37, (2021).

[23] Chris Hokamp, Demian Gholipour Ghalandari, Nghia The Pham, and John Glover, 'Dyne: Dynamic ensemble decoding for multi-document summarization', *CoRR*, **abs/2006.08748**, (2020).

[24] Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, et al., 'Efficient attentions for long document summarization', in *NAACL-HLT, Online, June 6-11, 2021*, pp. 1419–1436. ACL, (2021).

[25] Xin Ji and Wen Zhao, 'SKGSUM: abstractive document summarization with semantic knowledge graphs', in *IJCNN*, pp. 1–8. IEEE, (2021).

[26] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, et al., 'Neural extractive summarization with hierarchical attentive heterogeneous graph network', in *EMNLP (1)*, pp. 3622–3631. Association for Computational Linguistics, (2020).

[27] Zhihao Jia, Sina Lin, Mingyu Gao, Matei Zaharia, et al., 'Improving the accuracy, scalability, and performance of graph neural networks with roc', in *MLSys*. mlsys.org, (2020).

[28] Hanqi Jin, Tianming Wang, and Xiaojun Wan, 'Multi-granularity interaction network for extractive and abstractive multi-document summarization', in *ACL, Online, July 5-10 2020*, pp. 6244–6254. ACL, (2020).

[29] Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan, 'An empirical survey on long document summarization: Datasets, models, and metrics', *ACM Comput. Surv.*, **55**(8), (dec 2022).

[30] Taku Kudo and John Richardson, 'SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing', in *EMNLP*, pp. 66–71, Brussels, Belgium, (November 2018). Association for Computational Linguistics.

[31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, et al., 'BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension', in *ACL 2020, Online, July 5-10, 2020*, pp. 7871–7880. ACL, (2020).

[32] Chenliang Li, Weiran Xu, Si Li, and Sheng Gao, 'Guiding generation for abstractive text summarization based on key information guide network', 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. 55–60, New Orleans, Louisiana, (June 2018). ACL.

[33] Miao Li, Jianzhong Qi, and Jey Han Lau, 'Compressed heterogeneous graph for abstractive multi-document summarization', *CoRR*,

[9] https://www.maggioli.com/who-we-are/company-profile

abs/2303.06565, (2023).

[34] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, et al., 'Leveraging graph to improve abstractive multi-document summarization', in *ACL*, pp. 6232–6243. Association for Computational Linguistics, (2020).

[35] Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov, 'Towards understanding and mitigating social biases in language models', in *Proc. of the 38th ICML 2021, 18-24 July 2021, Virtual Event*, volume 139, pp. 6565–6576. PMLR, (2021).

[36] Kexin Liao, Logan Lebanoff, and Fei Liu, 'Abstract meaning representation for multi-document summarization', in *COLING, Santa Fe, New Mexico, USA, August 20-26, 2018*, pp. 1178–1190. ACL, (2018).

[37] Chin-Yew Lin, 'ROUGE: A package for automatic evaluation of summaries', in *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, (July 2004). Association for Computational Linguistics.

[38] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, et al., 'Generating wikipedia by summarizing long sequences', in *ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018*. OpenReview.net, (2018).

[39] Yang Liu and Mirella Lapata, 'Hierarchical transformers for multi-document summarization', in *ACL (1)*, pp. 5070–5081. Association for Computational Linguistics, (2019).

[40] Stefano Lodi, Gianluca Moro, and Claudio Sartori, 'Distributed data clustering in multi-dimensional peer-to-peer networks', in *(ADC), Brisbane, 18-22 January, 2010*, volume 104 of *CRPIT*, pp. 171–178. ACS, (2010).

[41] Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, et al., 'Multi-document summarization via deep learning techniques: A survey', *ACM Comput. Surv.*, **55**(5), 102:1–102:37, (2023).

[42] Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, et al., 'DYLE: dynamic latent extraction for abstractive long-input summarization', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 1687–1698. ACL, (2022).

[43] Ayushi Mathur and M Suchithra, 'Application of abstractive summarization in multiple choice question generation', in *2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, pp. 409–413, (2022).

[44] Gianluca Moro and Luca Ragazzi, 'Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes', in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 11085–11093. AAAI Press, (2022).

[45] Gianluca Moro and Luca Ragazzi, 'Align-then-abstract representation learning for low-resource summarization', *Neurocomputing*, **548**, 126356, (2023).

[46] Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli, 'Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy', *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**(12), 14417–14425, (Jun. 2023).

[47] Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi, 'Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature', in *ACL (1)*, pp. 180–189. Association for Computational Linguistics, (2022).

[48] Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, Giacomo Frisoni, Claudio Sartori, and Gustavo Marfia, 'Efficient memory-enhanced transformer for long-document summarization in low-resource regimes', *Sensors*, **23**(7), (2023).

[49] Gianluca Moro, Luca Ragazzi, Lorenzo Valgimigli, and Lorenzo Molfetta, 'Retrieve-and-marginalize end-to-end summarization of biomedical studies', in *Similarity Search and Applications - 16th International Conference, SISAP 2023, A Coruña, Spain, October 9-11, 2023, Proceedings*, Lecture Notes in Computer Science, pp. 1–9. Springer, (2023).

[50] Gianluca Moro and Stefano Salvatori, *Deep Vision-Language Model for Efficient Multi-modal Similarity Search in Fashion Retrieval*, 40–53, 09 2022.

[51] Gianluca Moro, Stefano Salvatori, and Giacomo Frisoni, 'Efficient text-image semantic search: A multi-modal vision-language approach for fashion retrieval', *Neurocomputing*, **538**, 126196, (2023).

[52] Gianluca Moro and Lorenzo Valgimigli, 'Efficient self-supervised metric information retrieval: A bibliography based method applied to COVID literature', *Sensors*, **21**(19), (2021).

[53] Donatella Muratore, Markus Hagenbuchner, Franco Scarselli, and Ah Chung Tsoi, 'Sentence extraction by graph neural networks', in *ICANN (3)*, volume 6354 of *Lecture Notes in Computer Science*, pp. 237–246. Springer, (2010).

[54] Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali, 'Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion', in *COLING, Santa Fe, New Mexico, USA, August 20-26 2018*, pp. 1191–1204. ACL, (2018).

[55] Narjes Nikzad-Khasmakhi, Mohammad-Reza Feizi-Derakhshi, Meysam Asgari-Chenaghlu, Mohammad Ali Balafar, et al., 'Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding', *CoRR*, **abs/2106.04939**, (2021).

[56] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: a method for automatic evaluation of machine translation', in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, (July 2002). Association for Computational Linguistics.

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, et al., 'Pytorch: An imperative style, high-performance deep learning library', in *NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, (2019).

[58] Yifu Qiu and Shay B. Cohen, 'Abstractive summarization guided by latent hierarchical document structure', in *EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 5303–5317. ACL, (2022).

[59] Nils Reimers and Iryna Gurevych, 'Sentence-bert: Sentence embeddings using siamese bert-networks', in *EMNLP/IJCNLP (1)*, pp. 3980–3990. Association for Computational Linguistics, (2019).

[60] Marco Ferdinand Salchner and Adam Jatowt, 'A survey of automatic text summarization using graph neural networks', in *COLING*, pp. 6139–6150. International Committee on Computational Linguistics, (2022).

[61] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, 'Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter', *CoRR*, **abs/1910.01108**, (2019).

[62] Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, et al., 'Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities', *CoRR*, **abs/2206.10883**, (2022).

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al., 'Attention is all you need', in *NIPS*, pp. 5998–6008, (2017).

[64] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, et al., 'Graph attention networks', *CoRR*, **abs/1710.10903**, (2017).

[65] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, et al., 'Heterogeneous graph neural networks for extractive document summarization', in *ACL*, pp. 6209–6219. Association for Computational Linguistics, (2020).

[66] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al., 'Huggingface's transformers: State-of-the-art natural language processing', *CoRR*, **abs/1910.03771**, (2019).

[67] Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, et al., 'BASS: boosting abstractive summarization with unified semantic graph', in *ACL/IJCNLP (1)*, pp. 6052–6067. Association for Computational Linguistics, (2021).

[68] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan, 'PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization', in *ACL (Volume 1: Long Papers)*, pp. 5245–5263, Dublin, Ireland, (May 2022). Association for Computational Linguistics.

[69] Divakar Yadav, Rishabh Katna, Arun Kumar Yadav, and Jorge Morato, 'Feature based automatic text summarization methods: A comprehensive state-of-the-art survey', *IEEE Access*, **10**, 133981–134003, (2022).

[70] Tiezheng Yu, Zihan Liu, and Pascale Fung, 'Adaptsum: Towards low-resource domain adaptation for abstractive summarization', in *NAACL-HLT*, pp. 5892–5904. Association for Computational Linguistics, (2021).

[71] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu, 'PEGASUS: pre-training with extracted gap-sentences for abstractive summarization', in *ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11328–11339. PMLR, (2020).

[72] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, et al., 'Bertscore: Evaluating text generation with BERT', in *ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, (2020).