A Unified Retrieval Framework with Document Ranking and EDU Filtering for Multi-document Summarization

Shiyin Tan*
Institute of Science Tokyo
Tokyo, Japan
tanshiyin1107@gmail.com

Jaeeon Park*
Institute of Science Tokyo
Tokyo, Japan
jaeeon@lr.pi.titech.ac.jp

Dongyuan Li†
The University of Tokyo
Center for Spatial Information Science
Tokyo, Japan
Institute of Science Tokyo
Tokyo, Japan
lidy94805@gmail.com

Renhe Jiang
The University of Tokyo
Center for Spatial Information Science
Tokyo, Japan
jiangrh@csis.u-tokyo.ac.jp

Manabu Okumura
Institute of Science Tokyo
Tokyo, Japan
oku@pi.titech.ac.jp

Abstract

In the field of multi-document summarization (MDS), transformer-based models have demonstrated remarkable success, yet they suffer an input length limitation. Current methods apply truncation after the retrieval process to fit the context length; however, they heavily depend on manually well-crafted queries, which are impractical to create for each document set for MDS. Additionally, these methods retrieve information at a coarse granularity, leading to the inclusion of irrelevant content. To address these issues, we propose a novel retrieval-based framework that integrates query selection and document ranking and shortening into a unified process. Our approach identifies the most salient elementary discourse units (EDUs) from input documents and utilizes them as latent queries. These queries guide the document ranking by calculating relevance scores. Instead of traditional truncation, our approach filters out irrelevant EDUs to fit the context length, ensuring that only critical information is preserved for summarization. We evaluate our framework on multiple MDS datasets, demonstrating consistent improvements in ROUGE metrics while confirming its scalability and flexibility across diverse model architectures. Additionally, we validate its effectiveness through an in-depth analysis, emphasizing its ability to dynamically select appropriate queries and accurately rank documents based on their relevance scores. These results demonstrate that our framework effectively addresses context-length constraints, establishing it as a robust and reliable solution for MDS.[1]

CCS Concepts

• **Information systems** → **Summarization**.

Keywords

Multi-document Summarization; Retrieval Framework; Document Ranking; Elementary Discourse Units

ACM Reference Format:
Shiyin Tan, Jaeeon Park, Dongyuan Li, Renhe Jiang, and Manabu Okumura. 2025. A Unified Retrieval Framework with Document Ranking and EDU Filtering for Multi-document Summarization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729884

1 Introduction

Multi-document summarization (MDS) is a task that aims to generate concise and coherent summaries by synthesizing information from multiple documents on the same topic [21, 29, 36, 40, 44]. MDS can lead to diverse applications, such as news aggregation [7, 13, 23], scientific research [11, 35, 59], and legal document analysis [17, 38, 55]. Current MDS approaches can be categorized into two classes: Graph-based models [9, 28, 45, 47, 65] and pre-trained language models [2, 46, 61]. Graph-based models rely on auxiliary information (*e.g.,* discourse structures) as an input graph to capture the cross-document relationships, while pre-trained language models use the attention mechanisms to capture them.

All these summarization models employ transformer [58] as a text encoder and are consequently constrained by a fixed input length, limiting the number of tokens they can process. A common solution is to apply truncation, dropping the last tokens of input documents to fit within the context length. However, this naive approach risks discarding critical information for summarization [37, 60]. Thus, *how to retain sufficient critical information within the length limitation* has become a crucial issue for MDS to enhance the quality of summaries. Current approaches often follow a "retrieve-then-summarize" paradigm to alleviate this issue [1, 12, 15, 64], as shown in Figure 1 (A). They use manually

*Both authors contributed equally to this research.
†Corresponding author.
[1]Our code is available at https://github.com/ShiyinTan/ReREF.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '25, July 13–18, 2025, Padua, Italy
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729884

created **queries** as guidance to rank **documents or sentences** from input documents or external knowledge bases and retrieve the top ranked contents to generate summaries. For instance, LightPAL [12] and DYLE [41] use dataset-provided queries to retrieve passages and sentences, respectively, for summarization.

Although these frameworks effectively retrieve documents from a large document collection, they face two major issues in MDS, as shown in Figure 1 (B): **(i) These retrieval methods [15] often heavily depend on manually well-crafted queries** to guide the ranking of passages or documents. For example, queries require precise human-written topic statements to ensure accurate ranking. However, MDS datasets do not provide the corresponding queries, and it is impractical to create them for each document set for MDS, as creating a topic statement requires reading the entire document set. Additionally, they overlook the fact that documents in MDS often cover the same topic from different perspectives [31], which suggests that queries can be directly inferred from documents. **(ii) Current retrieval models focus on a coarse granularity unit**, which inevitably includes irrelevant information for summarization, as pointed out in [6, 56, 62]. Specifically, some retrieval methods [12, 32, 49] focus on passages as retrieval units, while others [18, 25, 41] focus on sentences.

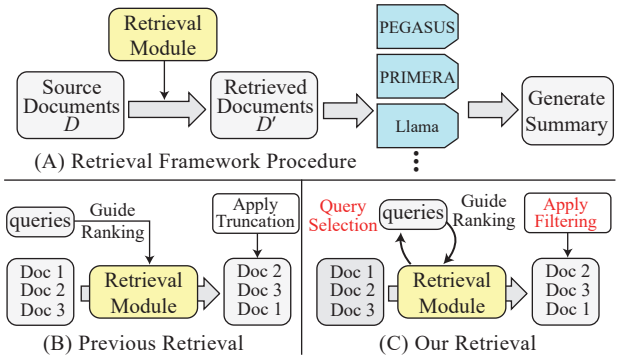

Figure 1: (A) The overall procedure of our retrieval model, which is model agnostic. (B) Previous retrieval needs queries as inputs and ranking over documents. (C) Our retrieval automatically selects queries for document ranking and applies filtering over EDUs.

To address the above issues, we propose a model-agnostic Retrieval framework, called ReREF, which unifies document Ranking and Elementary discourse unit (EDU)[2] Filtering for MDS. The overall procedure of ReREF is shown in Figure 1 (C). Firstly, to automatically select queries, we treat the problem as EDU ranking, where the top-ranked EDUs indicate that they have high relevance to the reference summary and can be used as queries for document ranking. Secondly, to fully utilize the limited length with critical information and alleviate the distraction from extraneous details, we not only rank documents to ensure that important ones are not truncated but also filter out bottom-ranked EDUs to eliminate irrelevant segments in sentences. Then, we employ the Expectation-Maximization (EM) algorithm [10] to unify automatic query selection, document ranking, and EDU filtering into a cohesive process.

[2]EDUs [4] are minimal coherent segments of text and are usually smaller than sentences. More details about EDUs are introduced in Section 3.3.

We apply ReREF to seven summarizers on four MDS datasets, using both fully supervised and few-shot settings. The results demonstrate that ReREF consistently improves summarization quality from the perspective of ROUGE-scores over the original summarizers. Additionally, human evaluations further assess the quality of the generated summaries, showing improvements in both informativeness and fluency with ReREF. The main contributions of this study can be summarized as follows:

- We design automatic query selection and irrelevant information filtering at the EDU level to avoid the need of human-written queries and reduce the space consumption of irrelevant content.
- We unify automatic query selection, document ranking, and EDU filtering into a model-agnostic retrieval framework using the EM algorithm to address the context length limitation.
- We evaluate ReREF with various summarizers in both fully supervised and few-shot settings to demonstrate its effectiveness. Additional human evaluations show that ReREF improves the quality of the generated summaries.

2 Related Work

2.1 Multi-document Summarization

MDS is a task that summarizes information from a set of related documents on the same topic [59], which can be categorized into three classes: Statistical and linguistic feature-based methods, graph-based models, and pre-trained language models. **Statistical and linguistic feature-based methods** are generally unsupervised and do not require labeled training data. Typical techniques include scoring sentences [16] based on term frequency inverse document frequency (TF-IDF), keyword occurrence, and semantic similarity measures [43]. **Graph-based models** require auxiliary information (*e.g.,* discourse structures) to build an input graph to capture the cross-document relationships. For example, models like BartGraphSum [45] and IESum [65] utilize graph representations, constructed by Information Extraction (IE), to encode complex relationships within and across documents. HGSUM [28] extends this concept by employing heterogeneous graphs to represent words, sentences, and documents at multiple semantic levels, further improving abstractive summarization performance. **Pre-trained language models** follow a transformer-based encoder-decoder architecture to capture the cross-document relationships by attention mechanisms. Efficient pre-trained transformers that can process long sequences (*e.g.,* Longformer [2]) have been also proven successful in summarization, typically by the ability to process long inputs, connecting information across the entire sequence. PRIMERA [61] and Centrum [46] introduce a specialized architecture for MDS, utilizing Longformer's global attention mechanisms to capture key information from multiple documents. Both the graph-based models and the pre-trained language models are well designed to capture key information across documents, but they still face challenges brought by context-length constraints. Specifically, they simply apply a truncation strategy that drops the last tokens in documents to fit the context length, leading to the lose of critical information necessary for summarization. ReREF builds upon such a challenge by introducing a retrieval framework that integrates both ranking and filtering into a unified process.

2.2 Retrieval Frameworks for Summarization

The studies on MDS often adopt a "retrieve-then-summarize" paradigm [1, 12, 15, 64], consisting of two main modules: a *Retriever* and a *Summarizer*. The *Retriever* selects passages based on the relevance between queries (either provided by humans or available in datasets) and passages (sourced from input documents or external knowledge bases). The *Summarizer* then generates summaries using the retrieved passages. For instance, LightPAL [12] implements a two-step retrieval process: initially retrieving passages with a conventional retriever (*e.g.,* BM25 [52]), followed by further refinement using large language models (LLMs). LogicSumm [32] utilizes LLMs for both document retrieval and summarization. DYLE [41] employs RoBERTa [33] to retrieve sentences from input documents. Despite effectively retrieving passages from a large document collection, these frameworks heavily depend on well-crafted queries [15] (*e.g.,* topic statements) to guide ranking and often use sentences or passages as their smallest retrieval units. However, in MDS, where documents typically cover the same topic from different perspectives [31], queries can often be inferred directly from the documents as they share critical information. Moreover, sentences and paragraphs frequently contain a mix of relevant and irrelevant information, and retrieval at this level inevitably includes irrelevant information for summarization. Our framework addresses these challenges by automatically generating queries from documents to avoid manual query creation and by filtering at the level of EDUs to ensure critical information is captured within the summarizer's limited context length.

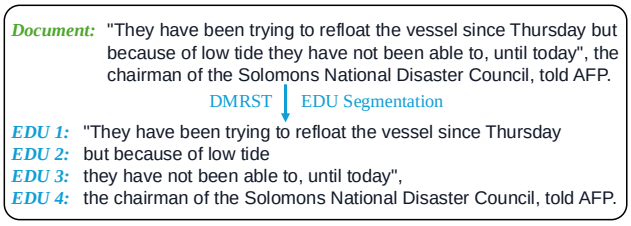

Figure 2: Example EDUs, segmented by DMRST [34], demonstrate their finer granularity than sentences while being more semantically meaningful than words or tokens.

3 Preliminaries

3.1 Notations

In this paper, we use capital and calligraphic letters to represent a document and a set of documents, respectively. For instance, D_i denotes the text of document i, while $\mathcal{D} = \{D_1, \ldots, D_n\}$ represents a collection of documents. Bold lowercase and uppercase letters are used to denote vectors and matrices, respectively. For example, $\mathbf{d}_i$ represents the embedding of document i, and $\mathbf{D}$ refers to all document embeddings in a document set $\mathcal{D}$. Additionally, we let T, E and D denote tokens, EDUs, and documents, respectively.

3.2 Problem Statement

MDS is defined as a generation task, given a set of documents on the same topic, where the objective is to generate a summary S^* that accurately and coherently represents the information in $\mathcal{D}$. Models for this task are typically trained in a supervised manner to minimize the difference between S^* and a reference summary.

However, MDS is inherently challenging due to the long-context nature of its inputs, often exceeding the input length limitation of summarization models. To address this limitation, many approaches [1, 3, 12, 15, 57] adopt a "retrieve-then-summarize" paradigm. In this study, we follow this paradigm and focus on designing a retrieval framework to enhance the quality of summary for MDS.

3.3 EDU Segmentation

EDUs [4], derived from Rhetorical Structure Theory (RST) [39], represent the minimal, coherent segments of discourse, that encapsulate individual propositions or ideas. A key aspect of our framework is leveraging EDUs to capture the semantic and structural granularity of input documents while filtering out irrelevant EDUs. This ensures that critical information is retained while irrelevant content is effectively filtered out. However, standard MDS datasets often lack EDU segmentation annotations. To address this limitation, we utilize the DMRST parser [34], a reliable tool designed for document-level segmentation. The DMRST parser offering robust multilingual support and delivering high-quality EDU segmentation. An example of EDU segmentation is shown in Figure 2.

4 Methodology

In this section, we introduce our unified retrieval framework of ranking and filtering for MDS. First, we describe how EDU and document embeddings are encoded and aggregated through pooling techniques (Section 4.1). Next, to automatically select latent queries and simultaneously perform ranking and filtering, we leverage the EM algorithm, which dynamically filters irrelevant content and ranks documents based on their relevance (Section 4.2). Then, we present the training dataset creation for our retrieval framework, which includes EDU-scoring and document-scoring (Section 4.3). Finally, we introduce the process of integrating our retrieval framework with the truncation strategy during inference and summarizing for the retrieved datasets with a summarizer (Section 4.4).

4.1 Encoding EDUs and Documents

As shown in Figure 3 (a), we first encode EDUs and documents into suitable representations. We encode input documents to token embeddings and apply a pooling strategy over the token embeddings to get the EDU and document representations. As we introduced in Section 3.3, the EDUs are obtained by the EDU segmentation.

Token Embeddings. Given a set of input documents $\mathcal{D}$, we encode each document D_i into token embeddings $\mathbf{T}_i$ by Longformer [2]. Since the document D_i may still meet the context limitation of Longformer, we apply a chunking strategy by splitting D_i into chunks with a fixed size, ensuring that all tokens are processed within the model's input constraints:

$$\mathbf{T}_i^j = \text{Enc}(C_i^j), \tag{1}$$

$$\mathbf{T}_i = \text{Concat}(\mathbf{T}_i^0, ..., \mathbf{T}_i^{|C_i|}), \tag{2}$$

where C_i^j represents the j-th chunk for document D_i, $|C_i|$ denotes the total number of chunks in D_i, and $\mathbf{T}_i^j$ corresponds to the token embeddings for the j-th chunk. All tokens in document D_i are

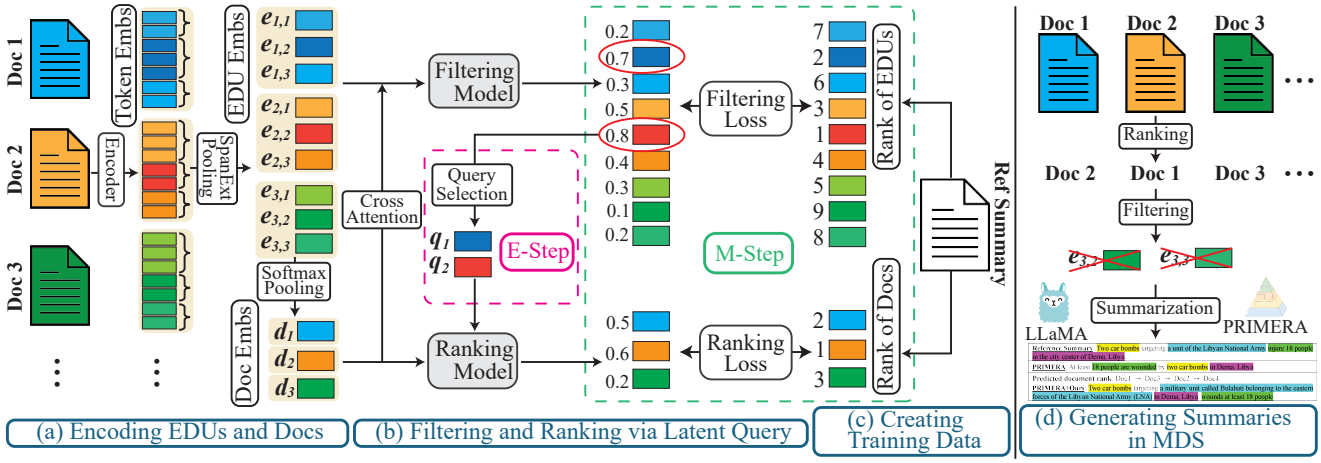

Figure 3: The overall process of ReREF, which extracts latent queries and performs ranking and filtering simultaneously.

concatenated into token embeddings $\mathbf{T}_i$, which can be pooled into EDU and document embeddings.

EDU Embeddings. After obtaining all token embeddings, EDUs can be derived using a pooling function, such as mean pooling. In this work, we adopt an attention-based weighted pooling method, SpanExt [26], which leverages a self-attentive span representation to effectively capture both the internal structure and contextual information of text spans. For EDU $E_{i,j}$, which denotes the j-th EDU in document D_i, its embedding is computed as:

$$\mathbf{e}_{i,j} = \sum_{k=1}^{\ell_{i,j}} a_{(i,j,k)} \cdot \mathbf{T}_{(i,j,k)}, \tag{3}$$

where $\ell_{i,j}$ indicates the total token number of EDU $E_{i,j}$ and $\mathbf{T}_{(i,j,k)}$ denotes the k-th token embedding of EDU $E_{i,j}$. The pooling weights $a_{(i,j,k)}$ are computed as:

$$a_{(i,j,k)} = \frac{\exp(\alpha_{(i,j,k)})}{\sum_{l=1}^{\ell_{i,j}} \exp(\alpha_{(i,j,l)})}, \tag{4}$$

$$\alpha_{(i,j,k)} = \mathbf{W}_2 \cdot \mathrm{ReLU}(\mathbf{W}_1 \cdot \mathbf{T}_{(i,j,k)} + \mathbf{b}_1) + \mathbf{b}_2, \tag{5}$$

where $\alpha_{(i,j,k)}$ is the unnormalized attention score, and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$ are trainable parameters. The normalized attention $a_{(i,j,k)}$ reflects the relative contribution of each token within all the tokens in the span of EDU $E_{i,j}$. The resulting EDU embeddings capture text semantics at a finer granularity, providing a robust foundation for subsequent filtering and ranking processes.

Document Embeddings. To obtain a holistic representation for each document, we aggregate the embeddings of its EDUs into a single document embedding using a gated pooling mechanism [22]. The process begins by transforming the EDU embeddings through a non-linear mapping function $g(\cdot)$, implemented by Feedforward Neural Network (FFN) in our approach. We then aggregate the transformed EDU embeddings using a Softmax Gating function [54]:

$$\mathbf{d}_i = \sum_{j=1}^{m_i} \beta_{i,j} \cdot g(\mathbf{e}_{i,j}), \quad \beta_{i,j} = \frac{\exp(\mathbf{w}^\top g(\mathbf{e}_{i,j}))}{\sum_{k=1}^{m_i} \exp(\mathbf{w}^\top g(\mathbf{e}_{i,k}))}, \tag{6}$$

where m_i is the total EDU number of document D_i, $\mathbf{w} \in \mathbb{R}^d$ is a trainable vector serving as a gating parameter, and $\beta_{i,j}$ represents the pooling weight assigned to j-th EDU in document D_i. This aggregation ensures that the document embedding $\mathbf{d}_i$ is primarily influenced by the most important EDUs, as determined by the attention weights $a_{(i,j,k)}$.

4.2 Filtering and Ranking via Latent Query

As shown in Figure 3 (b), after encoding EDUs and documents, the retrieval framework integrates filtering and ranking into a unified iterative process, represented by the filtering and ranking models, respectively. This process combines EDU-level salience estimation with document-level relevance scoring, and is performed within the EM algorithm, enabling the dynamic selection of latent queries and the updating of parameters in our retrieval framework.

Filtering Model. The filtering model performs two tasks of selecting the most salient EDUs as latent queries and identifying the least salient EDUs for filtering. Within this model, EDU embeddings are refined through cross-attention [5] with document embeddings, and then the salience score is estimated based on the refined EDU embeddings. The cross-attention updates the EDU embeddings by aligning them with the document embeddings $\mathbf{D}$:

$$\mathbf{E} = \mathrm{CrossAtt}(\mathbf{E}, \mathbf{D}, \mathbf{D}), \tag{7}$$

where $\mathbf{E} = \{\mathbf{e}_{i,j} | E_{i,j} \in \mathcal{E}\}$ are all EDU embeddings which serve as a **query**[3] in cross-attention, and $\mathbf{D} = \{\mathbf{d}_1, ..., \mathbf{d}_n\}$ are all document embeddings which serve as the **key** and **value** in cross-attention. This cross-attention ensures that the updated embeddings $\mathbf{E}$ emphasize EDUs shared across multiple documents, facilitating a more focused evaluation of their salience.

Then, a MLP-based classifier is applied to calculate the salience score $s_{i,j}$ of the j-th EDU in document D_i as:

$$s_{i,j} = \mathrm{softmax}(\mathbf{W}_{\text{filter}} \cdot \mathbf{e}_{i,j} + \mathbf{b}_{\text{filter}}), \tag{8}$$

[3]Please do not confuse the "query" in cross-attention with latent queries. The former refers to the query vector used in attention computation, while the latter serves as the selected EDUs for ranking.

where $\mathbf{W}_{\text{filter}}$ and $\mathbf{b}_{\text{filter}}$ are trainable parameters. Based on these salience scores, the top-k EDUs with the highest scores are selected as latent queries $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\}$, which represent critical aspects of documents and serve as the guidance for document ranking.

Ranking Model. The ranking model utilizes a multi-query ranking mechanism based on the latent queries selected by the EDU salience score. Building on the document embeddings in Eq. 6, the multi-query ranking evaluates the relevance score by considering the contribution of (all or selected) queries in $\mathbf{Q}$. For a given query $\mathbf{q} \in \mathbf{Q}$, the relevance score $r_{i,q}$ of document D_i is calculated as:

$$r_{i,q} = \frac{\exp(\text{sim}(\mathbf{d}_i, \mathbf{q}))}{\sum_{j=1}^{n} \exp(\text{sim}(\mathbf{d}_j, \mathbf{q}))}, \tag{9}$$

where $\text{sim}(\cdot,\cdot)$ is a similarity function, dot product in our setting, that evaluates the alignment between document embedding $\mathbf{d}_i$ and query embedding $\mathbf{q}$. The query embedding is selected based on the EDU salience score and corresponds to the EDU embedding in Eq. 7. This process is repeated for all queries and the relevance score across all queries are combined to form an overall relevance score for each document:

$$r_i = \frac{1}{k} \sum_{q \in \mathbf{Q}} r_{i,q}. \tag{10}$$

This aggregated score provides a unified measure of document importance across all queries, ensuring that the ranking reflects diverse perspectives, since multiple queries may reflect multiple aspects of documents.

Expectation-Maximization Optimization. The EM algorithm is an iterative method used for parameter optimization, particularly when the model depends on unobserved (latent) variables or when dealing with incomplete data [42]. In our work, it is particularly well-suited as the latent queries correspond to the latent variables in the process. The EM algorithm alternates between the E-step and the M-step. In the E-step, we select the most salient EDUs as latent queries, according to the EDUs' salience score in the filtering model. In the M-step, we update the parameters of both filtering and ranking models, according to the filtering and ranking losses.

The filtering loss is served for query selection and EDU filtering, and is implemented as an EDU ranking problem based on the salience score in Eq. 8. This loss ranks the most salient EDUs at the top and the least ones at the bottom. The top-ranked EDUs are selected as "*query*" EDUs, while the bottom-ranked EDUs are filtered out during the retrieval process. From this, our approach focuses only on accurately ranking the top and bottom EDUs, without considering the order of the other EDUs. Therefore, the filtering loss is optimized by the Bayesian Personalized Ranking (BPR) loss [51]:

$$\mathcal{L}_{\text{filter}} = -\sum_{i \sim \mathcal{P}_q} \sum_{j \in \overline{\mathcal{P}}_q} \log \sigma(s_i - s_j) - \sum_{i \sim \overline{\mathcal{P}}_f} \sum_{j \in \mathcal{P}_f} \log \sigma(s_i - s_j), \tag{11}$$

where σ is the sigmoid function, s_i is the salience score of the i-th EDU from Eq. 8, and $\mathcal{P}_q$ and $\mathcal{P}_f$ represent the top EDUs (query EDUs) and the bottom EDUs (filtered EDUs) in the ground-truth EDU ranking, respectively. The $\overline{\mathcal{P}_q}$ and $\overline{\mathcal{P}_f}$ indicate the complement set of query EDUs and filtered EDUs, respectively.

Similarly, the ranking loss assigns higher ranks to documents based on their alignment with all latent queries. We utilize the BPR loss for document ranking as well:

$$\mathcal{L}_{\text{rank}} = -\sum_{(i,j) \sim \mathcal{P}_g} \log \sigma\left(r_i - r_j\right), \tag{12}$$

where $(i, j) \sim \mathcal{P}_g$ denotes that document i is ranked higher than document j in the ground-truth ranking, and r_i indicates the relevance score between document i and all queries in $\mathbf{Q}$. The overall objective function integrates both ranking and filtering losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rank}} + \lambda \mathcal{L}_{\text{filter}}, \tag{13}$$

where λ is the balance weight. During training, the EM algorithm iteratively refines both EDU representations $\mathbf{e}_{i,j}$ and document embeddings $\mathbf{d}_i$, ensuring mutual optimization of filtering and ranking.

At last, the E-step and the M-step are alternately repeated, with $\mathbf{e}_{i,j}$ and $\mathbf{d}_i$ being refined in each iteration. Since query embeddings share the same representation with EDU representation $\mathbf{e}_{i,j}$, the refined $\mathbf{e}_{i,j}$ guides the selection of salient EDUs, while $\mathbf{d}_i$ improves the ranking of documents based on their relevance to the queries. This alternative optimization ensures that both latent queries and document embeddings dynamically adapt to the content's salience, optimizing the final ranking.

4.3 Creating Training Dataset for Retrieval

We detail the preparation of a training dataset for our retrieval framework in Figure 3 (c). For both query selection and EDU filtering, we measure the salience score for each EDU by aligning the EDU with the reference summary. For document ranking, we create the rank for documents by the relevance between each document and the reference summary. These provide the ground truth (oracle) for filtering and ranking, ensuring the dataset aligns with the requirement of our framework and supports effective supervision.

EDU Scoring. EDU-level scores are determined by evaluating the relevance of each segmented EDU against the reference summary. Using the EDU spans from segmentation annotation, we embed both the EDUs and the reference summary into a shared semantic space using a pre-trained model (multi-qa-mpnet-base-cos-v1) designed for Semantic Search [50]. The relevance of each EDU is calculated as the cosine similarity between its embedding and the embedding of the reference summary. These scores reflect the salience of the EDUs within the overall context and are leveraged to supervise the EDU ranking process, including both latent queries selection and EDU filtering.

Document Ranking. To establish document ranking, we calculate relevance scores between each document and the corresponding reference summary using the same pre-trained model used for EDU scoring. Each document is represented as a collection of EDU embeddings, and the overall document relevance score is computed by aggregating the cosine similarity scores of its EDUs with the gold summary. Documents are ranked based on these aggregated scores, with higher-ranked documents prioritized for inputs.

4.4 Generating Summaries in MDS

As shown in Figure 3 (d), after obtaining the well-trained retrieval framework, we infer the EDU filtering scores and document ranking scores. We then apply filtering and ranking to generate summaries.

Table 1: The statistics of the datasets. ∗ denotes that we utilized the dataset provided by PRIMERA [61].

Dataset	Train/Dev./Test	Avg. Docs	Len_{src}	Len_{tgt}
Multi-News [13]	44,972/5,622/5,622	2.8	1793	217
Multi-Xscience [35]	30,369/5,066/5,093	4.4	700	105
Wikisum* [8]	38,219/38,144/3,200	40	2238	113
WCEP-10 [14]	8,158/1,020/1,020	9.1	3866	28

Specifically, the retrieval model takes documents as input and generates relevance scores for documents and salience scores for their EDUs. Using these inferred relevance scores, we rank the documents, ensuring that the least important documents are truncated. Instead of simply dropping the last tokens during truncation, we filter out the lowest-ranked EDUs by the salience scores to fit the context length of downstream summarizer.

5 Experiments

5.1 Settings

Evaluation Datasets. We evaluated our approach on four multi-document summarization datasets from various domains (News, Scientific literature, and Wikipedia) and adopted the provided data split from the datasets. See Table 1 for detailed dataset statistics.

Base Summarization Models. We evaluated our retrieval framework to understand its effectiveness using various models and configurations. The models include BartGraphSum [45], BART [27], PEGASUS [63], PRIMERA [61], LLaMA variants (3.1-8B[4] and 3.2-1B[5]), and StableLM-Zephyr-3B[6]. BartGraphSum is a graph-enhanced model that incorporates an IE graph, constructed from Open Information Extraction (OIE) triplets, to improve summarization quality. BART is a transformer-based denoising autoencoder designed for sequence-to-sequence tasks. PEGASUS employs the Gap Sentence Generation (GSG) pre-training objective, where key sentences are masked and reconstructed, making it well-suited for summarization. PRIMERA is built on the LED [2] architecture and uses pyramid-based masked sentence pre-training to effectively aggregate information across multiple documents. The LLaMA models include the resource-intensive 3.1-8B variant and the efficient 3.2-1B model. StableLM-Zephyr-3B offers a lightweight LLM. For fair comparison, we used the same input context as PRIMERA when evaluating LLaMA variants, as LLaMA-3.1 supports a context length of up to 128k tokens. The LLaMA and StableLM are fine-tuned with Low-Rank Adapters (LoRA) [19].

Evaluation Metrics. For summarization evaluation, we followed previous work [61] and used ROUGE [30] scores (R-1, R-2, and R-L), which are the standard evaluation metrics.[7] For both query selection and filtering evaluations, we used Precision@K (Pre@K), which measures the proportion of salient EDUs among the top-K ranked EDUs for query selection, and the proportion of irrelevant EDUs among the bottom-K ranked EDUs for filtering. For document ranking evaluation, we used NDCG@K [20], MRR_1st [48], and MRR_2nd, where NDCG@K assesses the quality of ranking by considering the positions of documents in the top-K list of the results, and MRR_1st/MRR_2nd focuses on the positions of the top-1/top-2 ground-truth documents in the list of the results.

Original Truncation. For the original baseline models, we used their default truncation strategy, which truncates each input document evenly to fit the context length, as in [13, 61]. Specifically, if there are n documents with a total length of l tokens, the baseline models select l/n tokens from each document and concatenate the truncated documents as an input. Since LLaMA and StableLM are not specifically designed for summarization tasks, we apply the same truncation strategy as other baseline models for consistency.

Hyperparameter Settings. We performed a grid search to find the best hyperparameters on the validation dataset. Unless otherwise specified, we set the chunk size c to 1024 tokens, the filtering loss weight λ to 1.0, and the query number k to 10, ensuring consistent comparison across models and configurations. Additionally, we used the Adam optimizer with a learning rate of $3e-5$ and batch size of 16 for BartGraphSum, BART, PEGASUS, and PRIMERA, while a learning rate of $2e-4$ and batch size of 8 for LLaMA and StableLM.

5.2 Evaluation in Fully Supervised Settings

Table 2 shows consistent improvements achieved by our framework in the ROUGE metrics across various datasets and model configurations. Our retrieval framework achieves a significant performance gain in all datasets compared to the original base models. Specifically, our framework achieves an average gain of +0.43 on Multi-News, +0.95 on Multi-XScience, +1.85 on Wikisum, and +1.11 on WCEP-10 in PRIMERA. This demonstrates that our retrieval effectively filters out irrelevant information and prioritizes the salient content, enhancing overall summarization quality. Furthermore, it excels in settings with shorter context lengths (512 tokens), as seen with PEGASUS. The results for BART and PEGASUS across all MDS datasets highlight how our retrieval framework efficiently complements its pre-training strategy. Even LLMs such as the LLaMA family and StableLM show notable gains, emphasizing the universal importance of prioritizing salient information and minimizing irrelevant content, regardless of scale. These findings underscore the robustness of our retrieval framework in refining input quality and its versatility in enhancing diverse architectures.

5.3 Evaluation in Few-shot Settings

We further evaluated the retrieval framework under few-shot settings to understand its adaptability with limited supervision. We used only 1% of the available training data for few-shot evaluation. Table 3 presents the results. Our retrieval framework demonstrates consistent performance improvements across most datasets and models. Our retrieval framework shows impressive results on the Wikisum dataset, which is characterized by its highly fragmented input structure, with an average of 40 source documents per summary. Notably, PEGASUS achieves a significant +7.47 point improvement in ROUGE-2, demonstrating the framework's ability to filter irrelevant content and prioritize salient information effectively. This capability allows models to generate informative and concise summaries, even in the challenging few-shot settings, where limited supervision further complicates content selection. These results

[4] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[5] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct
[6] https://huggingface.co/stabilityai/stablelm-zephyr-3b
[7] We used https://github.com/google-research/google-research/tree/master/rouge with default stemmer settings.

Table 2: Comparison in different supervised settings. The number in parentheses is the token length limit for the model. – denotes that the improvement is not significant (p>=0.05) compared with the original score, using the paired bootstrap resampling method [24].

Model	Settings	Multi-News			Multi-XScience			Wikisum			WCEP-10		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
BartGraphSum(4050)	original	46.49	18.70	23.71	29.13	6.49	16.98	39.55	23.56	32.11	44.89	23.56	36.37
	retrieval	**47.73**	**19.38**	**24.66**	29.67^{-}	6.78^{-}	17.06^{-}	**41.33**	**25.50**	**34.01**	**46.29**	**24.64**	**37.70**
BART(1024)	original	44.87	16.93	22.21	22.21	4.91	13.70	39.66	22.54	32.26	43.24	21.96	34.66
	retrieval	**46.46**	**18.52**	**23.83**	**24.75**	**5.69**	**14.79**	**42.22**	**24.98**	**33.95**	**46.25**	**23.92**	**37.14**
PEGASUS(512)	original	47.70	18.36	23.62	28.73	5.21	15.99	31.21	16.46	25.68	42.43	17.33	32.35
	retrieval	**48.97**	**19.21**	**24.67**	**30.24**	**5.78**	**16.93**	**35.85**	**18.80**	**27.48**	**45.42**	**23.66**	**36.34**
PRIMERA(4096)	original	49.84	19.97	24.88	32.67	7.12	18.01	42.42	25.32	35.01	46.08	24.15	36.71
	retrieval	**50.27**	**20.38**	**25.33**	**34.73**	**7.44**	**18.29**	**44.53**	**27.10**	**36.72**	**47.33**	**25.08**	**37.87**
LLaMA 3.1-8B(4096)	original	43.25	14.24	20.74	31.78	6.16	17.46	41.64	24.65	34.08	43.50	20.77	34.01
	retrieval	**47.88**	**17.82**	**23.58**	32.49^{-}	**6.83**	**17.95**	**43.51**	**26.52**	**36.23**	**45.09**	**22.40**	**35.46**
LLaMA 3.2-1B(4096)	original	42.31	12.94	20.26	29.67	5.33	16.56	31.10	15.45	23.84	40.33	17.96	31.20
	retrieval	**46.37**	**16.46**	**22.88**	**30.73**	**5.76**	**17.13**	**36.81**	**19.35**	**28.90**	**41.44**	**19.46**	**32.83**
StableLm-Zephyr-3B (2048)	original	43.36	13.10	20.35	32.45	6.25	17.78	39.57	20.90	31.18	42.56	19.62	33.03
	retrieval	**46.84**	**16.31**	**22.64**	32.63^{-}	6.34^{-}	17.84^{-}	**40.56**	**21.93**	**31.98**	43.16^{-}	19.98^{-}	33.83^{-}

Table 3: Comparison in different few-shot evaluation settings. The notations are the same as those in Table 2.

Model	Settings	Multi-News			Multi-XScience			Wikisum			WCEP-10		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
BartGraphSum(4050)	original	41.26	14.99	20.12	25.76	4.89	14.65	31.80	14.29	24.63	37.87	17.00	29.09
	retrieval	**42.71**	**15.56**	**21.49**	26.48^{-}	4.94^{-}	**15.77**	**35.36**	**18.18**	**27.84**	38.03^{-}	16.97^{-}	29.18^{-}
BART(1024)	original	41.23	15.81	20.94	21.56	4.98	11.34	28.20	10.99	20.76	35.66	15.21	26.47
	retrieval	**42.68**	16.18^{-}	21.47^{-}	**23.29**	4.93^{-}	**12.40**	**31.21**	**13.07**	**23.46**	**38.79**	**17.72**	**29.56**
PEGASUS(512)	original	42.39	13.20	20.70	27.40	4.76	15.05	23.89	5.51	13.90	38.18	18.94	29.53
	retrieval	**43.78**	**14.13**	**22.51**	**28.69**	**5.01**	15.27^{-}	**31.36**	**11.76**	**19.67**	**39.54**	**19.88**	**30.49**
PRIMERA(4096)	original	46.10	16.80	22.63	29.53	5.31	15.27	36.48	17.85	28.43	40.66	18.64	31.34
	retrieval	**47.38**	**17.34**	**23.09**	**31.21**	**5.62**	**16.20**	**39.29**	**20.17**	**30.60**	**42.46**	**19.72**	**32.36**
LLaMA 3.1-8B(4096)	original	42.16	13.13	20.09	30.44	5.58	16.85	40.72	22.66	32.60	40.96	18.62	31.44
	retrieval	**45.56**	**16.01**	**21.97**	**31.01**	**5.83**	**17.09**	**41.85**	**23.93**	**33.92**	**43.36**	**19.81**	**33.35**
LLaMA 3.2-1B(4096)	original	38.16	12.02	18.57	27.64	4.15	14.48	29.79	15.64	25.33	36.62	16.68	29.41
	retrieval	**39.23**	**13.01**	**19.77**	**28.53**	**4.57**	**15.04**	**32.58**	**16.52**	**26.81**	**38.90**	**17.85**	**30.91**
StableLm-Zephyr-3B (2048)	original	41.58	12.45	19.71	30.20	5.28	16.55	36.26	17.10	27.41	40.58	17.92	31.33
	retrieval	**45.09**	**14.54**	**21.31**	30.36^{-}	5.33^{-}	16.67^{-}	**37.26**	**18.45**	**28.28**	41.28^{-}	18.28^{-}	32.23^{-}

highlight that our retrieval framework enhances performance not only in full-supervision scenarios but also in limited supervision ones. Its ability to handle dispersed inputs underscores its versatility and practical value in real-world summarization tasks.

5.4 In-depth Analysis

To evaluate the accuracy of our retrieval framework, we analyzed its performance on two key tasks: EDU selection and document ranking. For EDU selection, which includes both query selection and EDU filtering, we compared it against BM25 [52], a bag-of-words retrieval function that ranks content based on the occurrence of query terms. For document ranking, we used BM25 [52] and DYLE [41] as baselines. DYLE[8] jointly trains a retriever and a generator, treating retrieved sentences as latent variables to guide dynamic attention and highlight salient information for long-document summarization. We focused on the retriever component of DYLE to compare it with our module in the effectiveness in document ranking. Since both BM25 and DYLE require a query as an input, we used RAKE (Rapid Automatic Keyword Extraction) [53], an unsupervised keyword extraction algorithm, to simulate queries. For the EDU selection task, we used the ranked EDUs based on the EDU scoring from Section 4.3 as the ground truth. Specifically, the top-k EDUs were used as the ground truth for Precision@K in query selection and the bottom-K EDUs were used in EDU filtering. For the document ranking task, we used the document ranking labels from Section 4.3 as the ground truth.

For query selection, we evaluated how effectively the retrieval framework identifies the most salient EDUs. Unlike BM25, which

[8]We used the arXiv retriever checkpoint provided by the authors.

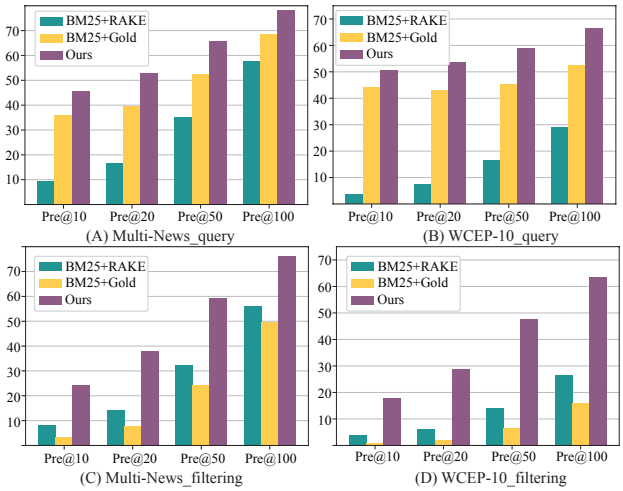

Figure 4: Precision@K (K = 10, 20, 50, 100) for query selection and filtering. (A) (B): the proportion of the top-K selected EDUs for query selection that align with the most salient EDUs. (C) (D): the proportion of the bottom-K selected EDUs for filtering that align with the least salient EDUs.

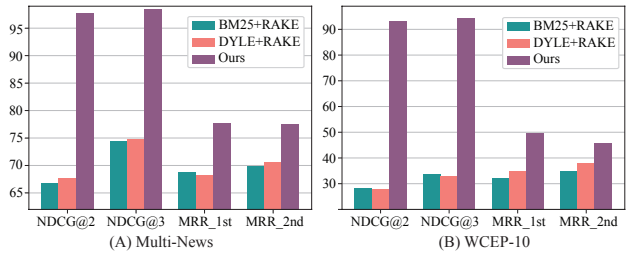

Figure 5: Document ranking accuracy. NDCG@K evaluates how effectively the most relevant documents are ranked within the top K positions. MRR_1st and MRR_2rd measure the average rank accuracy of the most and second most relevant documents, respectively. (A): Multi-News. (B): WCEP-10.

requires an explicit query to retrieve content, our retrieval framework dynamically selects EDUs directly from input documents. To ensure fair comparison, we constructed two variants of BM25: (1) BM25+RAKE, where the query is derived using RAKE to generate realistic and robust baselines; and (2) BM25+Gold, where the query is the EDU with the highest relevance score to the reference summary, as defined in Section 4.3. As shown in Figures 4 (A) and (B), our framework outperforms BM25+RAKE and BM25+Gold across all thresholds, demonstrating its ability to better align with salient EDUs. Remarkably, even when BM25+Gold utilizes the most relevant EDUs to the reference summary as a query, our retrieval framework significantly outperforms it across all thresholds. Similarly, for EDU filtering, we evaluated how effectively the retrieval framework identifies the least salient EDUs (Figures 4 (C) and (D)). Our module exhibits superior performance, effectively identifying least salient content. We attribute the lower performance of BM25+Gold compared to BM25+RAKE to differences in their query formulation: Gold ensures high specificity by aligning with the reference summary but lacks contextual breadth, whereas RAKE captures broader document context through co-occurrence patterns, albeit with less specificity. These results indicate that our module is not only adept at selecting salient EDUs but also proficient in filtering extraneous information, ensuring a context-aware retrieval that is independent of the human-written queries.

Building on these results, we next evaluated the document ranking accuracy of our retrieval framework. We compared our module against BM25+RAKE and DYLE+RAKE baselines. As shown in Figure 5, our framework consistently achieves higher NDCG@K and MRR scores across datasets. On Multi-News, for instance, it surpasses the baselines in both metrics, ranking the most salient documents at the top. Furthermore, on more challenging datasets like WCEP-10, our retrieval framework demonstrates robust performance with significant improvements over the baselines. These results emphasize the module's capability to align with reference content while reducing dependence on pre-defined explicit queries.

5.5 Human Evaluation

To assess the quality of generated summaries, we conducted a human evaluation on a randomly sampled 100 data points from the Multi-News test dataset. The evaluation was conducted using Amazon Mechanical Turk,[9] with 10 evaluators holding at least a US high school diploma and a bachelor's degree. Evaluators assess the overall quality of summaries on a Likert scale from 1 to 5 (5 being the highest) across three criteria, following the evaluation protocol of GraphSum [29]: (1) *Informativeness*: does the summary convey important facts of the input? (2) *Fluency*: is the summary fluent and grammatical? (3) *Succinctness*: does the summary avoid repeating information? The results of the evaluation are presented in Table 4. Our retrieval framework shows its effectiveness in improving the quality of generated summaries. Specifically, it outperforms the original truncation of the base model PRIMERA in informativeness and fluency, while maintaining comparable performance in succinctness. These results highlight the capability of our retrieval framework to generate summaries that are not only more informative and fluent but also concise.

Table 4: Human Evaluation of our retrieval framework. The notations are the same as those in Table 2.

Settings	Informativeness	Fluency	Succinctness
original	4.03	4.20	4.30
retrieval	**4.16**	**4.38**	4.31^{-}

5.6 Ablation Study and Parameter Analysis

To further evaluate the effectiveness of our retrieval framework, we conducted an ablation study and parameter analysis. These experiments were conducted using PRIMERA.[10]

Ablation Study. To investigate the effectiveness of ranking and filtering in our retrieval framework, we applied the following variants: (1) "w/o rank": we used only the EDU filtering in our truncation strategy, with using random permutation for the rank

[9] https://www.mturk.com/

[10] While we show the parameter analysis only on the Multi-News dataset, other datasets also exhibit the same parameter trends.

of documents. (2) "w/o filter": we used only the document ranking, where documents were ranked, but the last tokens were dropped in stead of filtering irrelevant EDUs. (3) "w/o both": both document ranking and filtering were removed, where randomly permuted documents were concatenated and the last tokens were dropped to fit the length limitation. Experimental results in Table 5 show that "w/o rank" and "w/o filter" result in a measurable decrease across metrics, revealing that both ranking and filtering benefit our retrieval framework in summarization. Furthermore, "w/o both" has the lowest performance, demonstrating the complementary nature of ranking and filtering in enhancing the retrieval framework.[11]

Table 5: Ablation study of our retrieval framework.

Settings	Multi-News			WCEP-10		
	R-1	R-2	R-L	R-1	R-2	R-L
Ours	50.27	20.38	25.33	47.33	25.08	37.87
w/o rank	49.75	20.25	25.15	46.08	24.26	36.74
w/o filter	49.95	20.26	25.22^{-}	46.75	24.66	37.23
w/o both	49.55	20.14	25.07	44.92	22.59	35.01

Parameter Analysis. We evaluated the parameter sensitivity by varying the target hyperparameter on the validation dataset while keeping other hyperparameters fixed. Specifically, we investigated the influence of query numbers and loss balance weight λ in the ranking accuracy and summarization performance. We found that the number of latent queries significantly impacts the ranking accuracy of documents. As shown in Figure 6 (A), MRR rapidly increases as the query number grows from 1 to 7. This suggests that a small number of queries is insufficient to capture the diversity of salient information in multi-document summarization. The ROUGE-1 score also follows a similar trend, saturating at around 10 queries, which shows the highest performance. We also identified that the balance weight λ in the loss function has minimal influence on both ranking and summarization performance. Figure 6 (B) illustrates that varying λ from 0.2 to 2.0 results in only marginal changes in both the ROUGE-1 score and MRR. This demonstrates that the framework is robust to changes in the balance weight λ, allowing for flexibility in tuning without significantly affecting performance. The consistent results across different λ values highlight the stability of the proposed model.

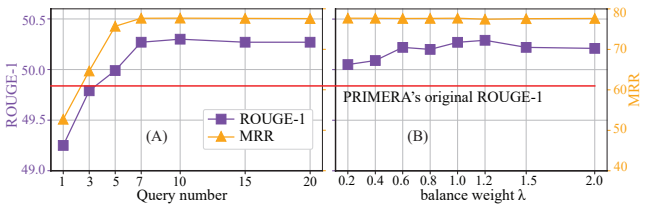

Figure 6: ROUGE score and document accuracy with different parameter settings. (A): varying the query number. (B): varying the loss balance weight λ.

[11]Please note that our truncation strategy variant, "w/o both", is different from the original truncation in PRIMERA, and the results of "w/o both" in Table 5 do not align with "original" in Table 2. Specifically, the original truncation evenly truncates each document. In contrast, the "w/o both" variant concatenates all randomly permuted documents and drops the last tokens, which truncates only the last ranked documents.

Documents (truncated):

Doc 1: ~~At least 18 people~~ were wounded in two car bomb explosions || that targeted a military unit in Libya's eastern coastal city of Derna, sending plumes of black smoke into the sky, a medical source and residents told Reuters early on Sunday. Residents said the car bombs targeted a military unit called Bulahati belonging to the eastern forces of the Libyan National Army (LNA) in the city center. "We heard the first explosion, but we thought it was fireworks, then we heard the second," one resident told Reuters by telephone. …

Doc 2: ~~Residents said~~ the car bombs targeted a military unit called Bulahati belonging to the eastern forces of the Libyan National Army (LNA) in the city center. "We heard the first explosion, but we thought it was fireworks, then we heard the second," ~~one resident told Reuters by telephone~~. "We found people around the Bulahati military unit ~~and there was huge black smoke in the sky~~," ~~another added.~~ "We then discovered it was car bombs." Derna, once a jihadist bastion, is about 292 km (182 miles) distant from Libya's second city, Benghazi, and was declared to be under the complete control of Khalifa Haftar's LNA in June 2018. After the ouster of longtime ruler Muammar Gaddafi in a NATO-backed uprising in 2011, militant groups Al Qaeda and Islamic State have used the oil-rich country as a base for attacks, ~~exploiting its chaos and lack of security~~.

Doc 3: Derna: Around 18 people were injured after two bomb-laden cars exploded near the headquarters of the Libyan National Army (LNA) here, according to local media reports on Sunday. This comes as more than 90,000 people have been displaced ever since the armed conflict between Khalifa Haftar-led army and UN-backed government broke out in Tripoli on April 12. …

Doc 4: The Libyan National Army ~~(LNA)~~ repelled the offensive of the forces of the Government of National Accord ~~(GNA)~~ on Tripoli airport, the LNA General Command press service director Khalifa Obeidi said on Monday. "LNA repelled an armed attack by [GNA armed] groups... on Tripoli airport on Sunday morning," he said. ~~Obeidi noted that~~ 15 GNA militants, including six mercenaries from Chad, were killed in the clash. In April, Khalifa Haftar, the head of the LNA, launched an offensive to retake Tripoli from control of the internationally-recognised Government of National Accord ~~(GNA)~~. The LNA has already recaptured a number of settlements near the capital and the Tripoli International Airport, located around 20 miles away from the city. The forces loyal to the GNA announced a counteroffensive, ~~dubbed Volcano of Rage~~. Since the overthrow and killing of Libya's long-time leader Muammar Gaddafi in 2011, ~~the country has been gripped by conflict~~. Libya is now divided between two governments, ~~with the eastern part controlled by the LNA, and the western part governed by the UN-backed GNA~~.

Reference Summary: Two car bombs targeting a unit of the Libyan National Army injure 18 people in the city center of Derna, Libya.

PRIMERA: At least 18 people are wounded by two car bombs in Derna, Libya.

Predicted document rank: Doc1 → Doc3 → Doc2 → Doc4

PRIMERA+Ours: Two car bombs targeting a military unit called Bulahati belonging to the eastern forces of the Libyan National Army (LNA) in Derna, Libya, wounds at least 18 people.

Figure 7: Example of generated summaries, latent query selection, ranking, and filtering.

5.7 Case Study

Figure 7 illustrates an example of the generated summary using PRIMERA with and without our retrieval framework. In the figure, key information is highlighted in the same color, *e.g.,* Two car bombs, selected queries are underlined in blue, *e.g.,* At least 18..., and filtered EDUs are struck through in pink, *e.g.,* ~~(GNA)~~. Our retrieval framework ranks Document 1 as the highest for its rich key information and Document 4 as the lowest for its irrelevance. By incorporating our retrieval framework into the truncation strategy, we filter out EDUs from the lower-ranked documents (Documents 2 and 4), effectively removing repetitive and irrelevant content. Additionally, we observe that selected queries, identified by our retrieval framework, align closely with the reference summary, such as *"two car bombs targeting a military unit in Derna."*

6 Conclusion

In this work, we aimed to propose a retrieval model that tackles the context length limitation in MDS models. To achieve this goal, we proposed a novel retrieval framework for multi-document summarization that integrates both query selection, document ranking and EDU filtering into an iterative Expectation-Maximization process. In our evaluation across diverse datasets and base models, the framework consistently improved ROUGE and ranking-specific metrics such as precision and MRR. The analysis demonstrated its scalability, flexibility, and potential for integration into existing pipelines. In future work, we aim to enhance our framework's efficiency when it is applied to large-scale, real-world multi-document summarization scenarios.

References

[1] Chenxin An, Ming Zhong, Zhichao Geng, Jianqiang Yang, and Xipeng Qiu. 2021. RetrievalSum: A Retrieval Enhanced Framework for Abstractive Summarization. *CoRR* abs/2109.07943 (2021). https://arxiv.org/abs/2109.07943

[2] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *CoRR* abs/2004.05150 (2020). https://arxiv.org/abs/2004.05150

[3] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. 152–161.

[4] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. *Current and new directions in discourse and dialogue* (2003), 85–112.

[5] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. 2021. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 347–356.

[6] Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X Retrieval: What Retrieval Granularity Should We Use?. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 15159–15177.

[7] Xiuying Chen, Mingzhe Li, Shen Gao, Xin Cheng, Qingqing Zhu, Rui Yan, Xin Gao, and Xiangliang Zhang. 2024. Flexible and Adaptable Summarization via Expertise Separation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2018–2027.

[8] Nachshon Cohen, Oren Kalinsky, Yftah Ziser, and Alessandro Moschitti. 2021. WikiSum: Coherent Summarization Dataset for Efficient Human-Evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL: Short Papers)*. 212–219.

[9] Peng Cui and Le Hu. 2021. Topic-Guided Abstractive Multi-Document Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 1463–1472. https://doi.org/10.18653/V1/2021.FINDINGS-EMNLP.126

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B* 39 (1977), 1–38.

[11] Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. MSˆ2: Multi-Document Summarization of Medical Studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7494–7513.

[12] Masafumi Enomoto, Kunihiro Takeoka, Kosuke Akimoto, Kiril Gashteovski, and Masafumi Oyamada. 2024. LightPAL: Lightweight Passage Retrieval for Open Domain Multi-Document Summarization. *CoRR* abs/2406.12494 (2024). https://arXiv.2406.12494

[13] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 1074–1084.

[14] Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 1302–1308.

[15] John M. Giorgi, Luca Soldaini, Bo Wang, Gary D. Bader, Kyle Lo, Lucy Lu Wang, and Arman Cohan. 2023. Open Domain Multi-document Summarization: A Comprehensive Study of Model Brittleness under Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 8177–8199.

[16] Jade Goldstein, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 workshop: automatic summarization*.

[17] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, and et al. 2023. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS)*.

[18] Itay Harel, Hagai Taitelbaum, Idan Szpektor, and Oren Kurland. 2022. A Dataset for Sentence Retrieval for Open-Ended Dialogues. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2960–2969.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations (ICLR)*.

[20] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[21] Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-Granularity Interaction Network for Extractive and Abstractive Multi-Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 6244–6254.

[22] M.I. Jordan and R.A. Jacobs. 1993. Hierarchical mixtures of experts and the EM algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN)*. 1339–1344.

[23] Subhendu Khatuya, Koushiki Sinha, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2024. Instruction-Guided Bullet Point Summarization of Long Financial Earnings Call Transcripts. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2477–2481.

[24] Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 388–395.

[25] Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, and Manabu Okumura. 2021. Considering Nested Tree Structure in Sentence Extractive Summarization with Pre-trained Transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4039–4044.

[26] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 188–197.

[27] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 7871–7880.

[28] Miao Li, Jianzhong Qi, and Jey Han Lau. 2023. Compressed Heterogeneous Graph for Abstractive Multi-Document Summarization. In *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*. 13085–13093.

[29] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 6232–6243.

[30] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, 74–81.

[31] Chin-Yew Lin and Eduard Hovy. 2002. From single to multi-document summarization. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 457–464.

[32] Shengjie Liu, Jing Wu, Jingyuan Bao, and et al. 2024. Towards a Robust Retrieval-Based Summarization System. *CoRR* abs/2403.19889 (2024). https://arxiv.org/abs/2403.19889

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). http://arxiv.org/abs/1907.11692

[34] Zhengyuan Liu, Ke Shi, and Nancy F. Chen. 2021. DMRST: A Joint Framework for Document-Level Multilingual RST Discourse Segmentation and Parsing. *CoRR* abs/2110.04518. https://arxiv.org/abs/2110.04518

[35] Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8068–8074.

[36] Congbo Ma. 2021. Improving Deep Learning based Multi-document Summarization through Linguistic Knowledge. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2704. https://doi.org/10.1145/3404835.3463268

[37] Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z. Sheng. 2023. Multi-document Summarization via Deep Learning Techniques: A Survey. *ACM Comput. Surv.* 55, 5 (2023), 102:1–102:37.

[38] Manuj Malik, Zheng Zhao, Marcio Fonseca, Shrisha Rao, and Shay B. Cohen. 2024. CivilSum: A Dataset for Abstractive Summarization of Indian Court Decisions. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2241–2250.

[39] William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse* 8, 3 (1988), 243–281.

[40] Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document Summarization with Maximal Marginal Relevance-guided Reinforcement Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1737–1751.

[41] Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir R. Radev. 2022. DYLE: Dynamic Latent Extraction for Abstractive Long-Input Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. 1687–1698.

[42] Geoffrey J McLachlan and Thriyambakam Krishnan. 2008. *The EM algorithm and extensions*. John Wiley & Sons.

[43] Şaziye Betül Özateş, Arzucan Özgür, and Dragomir Radev. 2016. Sentence similarity based on dependency tree kernels for multi-document summarization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2833–2838.

[44] Richard Yuanzhe Pang, Ádám Dániel Lelkes, Vinh Q. Tran, and Cong Yu. 2021. AgreeSum: Agreement-Oriented Multi-Document Summarization. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 3377–3391. https://doi.org/10.18653/V1/2021.FINDINGS-ACL.299

[45] Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently Summarizing Text and Graph Encodings of Multi-Document Clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 4768–4779.

[46] Ratish Surendran Puduppully, Parag Jain, Nancy Chen, and Mark Steedman. 2023. Multi-Document Summarization with Centroid-Based Pretraining. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, (ACL: Short Papers)*. 128–138.

[47] Yutong Qu. 2024. Leveraging Knowledge-aware Methodologies for Multi-document Summarization. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Roy Ka-Wei Lee, Ravi Kumar, and Hady W. Lauw (Eds.). ACM, 1206–1209. https://doi.org/10.1145/3589335.3651262

[48] Dragomir R. Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating Web-based Question Answering Systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*.

[49] Thilina Chaturanga Rajapakse, Andrew Yates, and Maarten de Rijke. 2024. Negative Sampling Techniques for Dense Passage Retrieval in a Multilingual Setting. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 575–584.

[50] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[51] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, 452–461.

[52] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389.

[53] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory* (2010), 1–20.

[54] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *5th International Conference on Learning Representations (ICLR)*.

[55] Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. 2022. Multi-LexSum: Real-World Summaries of Civil Rights Lawsuits at Multiple Granularities. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS)*.

[56] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large Language Models Can Be Easily Distracted by Irrelevant Context. In *International Conference on Machine Learning (ICML)*, Vol. 202. PMLR, 31210–31227.

[57] Zhengliang Shi, Shen Gao, Zhen Zhang, Xiuying Chen, Zhumin Chen, Pengjie Ren, and Zhaochun Ren. 2023. Towards a Unified Framework for Reference Retrieval and Related Work Generation. In *Findings of the Association for Computational Linguistics: EMNLP*. 5785–5799.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurPS)*. 6000–6010.

[59] Pancheng Wang, Shasha Li, Dong Li, Kehan Long, Jintao Tang, and Ting Wang. 2024. Disentangling Instructive Information from Ranked Multiple Candidates for Multi-Document Scientific Summarization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2028–2037.

[60] Ruben Wolhandler, Arie Cattan, Ori Ernst, and Ido Dagan. 2022. How "Multi" is Multi-Document Summarization?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 5761–5769.

[61] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. 5245–5263.

[62] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 14672–14685.

[63] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, Vol. 119. PMLR, 11328–11339.

[64] Weijia Zhang, Svitlana Vakulenko, Thilina Rajapakse, Yumo Xu, and Evangelos Kanoulas. 2021. Tackling query-focused summarization as a knowledge-intensive task: A pilot study. *arXiv preprint arXiv:2112.07536* (2021).

[65] Zixuan Zhang, Heba Elfardy, Markus Dreyer, Kevin Small, Heng Ji, and Mohit Bansal. 2023. Enhancing Multi-Document Summarization with Cross-Document Graph-based Information Extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. 1688–1699.