Language Models of Code are Few-Shot Planners and Reasoners for Multi-Document Summarization with Attribution

Abhilash Nandy[1*], **Sambaran Bandyopadhyay**[2]
[1]Indian Institute of Technology Kharagpur, India
[2]Adobe Research
nandyabhilash@gmail.com, samb.bandyo@gmail.com

Abstract

Document summarization has greatly benefited from advances in large language models (LLMs). In real-world situations, summaries often need to be generated from multiple documents with diverse sources and authors, lacking a clear information flow. Naively concatenating these documents and generating a summary can lead to poorly structured narratives and redundancy. Additionally, attributing each part of the generated summary to a specific source is crucial for reliability. In this study, we address multi-document summarization with attribution using our proposed solution ***MiDAS-PRo***, consisting of three stages: (i) Planning the hierarchical organization of source documents, (ii) Reasoning by generating relevant entities/topics, and (iii) Summary Generation. We treat the first two sub-problems as a code completion task for LLMs. By incorporating well-selected in-context learning examples through a graph attention network, LLMs effectively generate plans and reason topics for a document collection. Experiments on summarizing scientific articles from public datasets show that our approach outperforms state-of-the-art baselines in both automated and human evaluations.

1 Introduction

The exploration of text-to-text generation within the NLP community has evolved through a variety of methodologies, experiencing considerable progress (Zhang et al. 2020; Song et al. 2018; Wiseman, Shieber, and Rush 2018; Zhang et al. 2020). In recent years, the exponential growth of digital information has led to an increased demand for automated methods to summarize this vast amounts of data (El-Kassas et al. 2021). Emergence of Large Language Models (LLMs) enables effective and efficient summarization of huge amount of text data (Grail, Perez, and Gaussier 2021; Jin et al. 2024). In real-world situations, text may be derived from a variety of sources, each with its distinct characteristics. In these instances, a user might have a particular intent in mind, such as a specific topic, and desire to create a coherent summary that encompasses relevant information from those diverse sources in relation to the intended topic.

With ever-increasing context windows, LLMs could be leveraged for the task of multi-document summarization. However, such works (Kurisinkel and Chen 2023; Xiao et al. 2022; Wang et al. 2023; Shi et al. 2023) are few in number, and suffer from drawbacks such as need for expensive fine-tuning and lack of a source document attribution (citation) module, which is essential to make the summary more reliable, transparent, and interpretable. Also, several methods augment annotated data/metadata as training inputs to the collection of documents to be summarized, such as graphs (Zhang et al. 2023; Pasunuru et al. 2021), document timestamps (Song et al. 2024; Chen et al. 2023b, 2019), which are difficult to obtain. Hence, there is a lack of an LLM-based multi-document **attribution-inclusive** summarization framework to generate **intent-based** summaries with **negligible training cost** and **annotations**.

In this paper, we propose ***MiDAS-PRo*** (Multi-Document Attribution-inclusive Summarization via Planning-cum-Reasoning) to mitigate aforementioned issues. ***MiDAS-PRo*** uses a Code-LLM to plan the layout of the attribution of source documents in the final summary (where attribution is the citations to the source documents) and to reason by generating important domain-specific entities/-topics corresponding to the citations. Finally, we utilize this planning-cum-reasoning information to enhance the attribution-inclusive summary generation capability of an LLM of interest. The planning-cum-reasoning and the summarization modules are all carried in an in-context learning setting, thus requiring no LLM training. The In-Context Examples having the most similar attribution/reference layout to the test example in their summaries are selected by training a 2-layer Graph Attention Network (GAT), which incurs minimal training cost and time. ***MiDAS-PRo*** Framework is depicted in Fig. 1. ***MiDAS-PRo*** provides a sizeable improvement in summarization metrics and a massive improvement in clustering-based metrics when applied across different LLMs and varying number of in-context examples.

We make the following contributions in this paper - (1) Code LLMs are Few-Shot Planners and Reasoners for Multi-Document Summarization. To that effect, we propose ***MiDAS-PRo*** for multi-document summarization, where planning-cum-reasoning is carried out in 2 stages - *Hierarchical Reference Layout Tree (HRLT) Generation*, followed by *Sentence-Wise Entity Generation* (2) We introduce several clustering-based metrics to evaluate the attribution of references in the generated summary (3) Strate-

[*]This work was done during the internship at Adobe Research

gically organizing the references within the summary in ***MiDAS-PRo*** clearly indicates which specific source documents contribute to different parts of the summary, thus enhancing transparency and interpretability (4) To make the in-content examples more relevant, we utilize a novel GAT-based In-Context Example Selection Method for the proposed ***MiDAS-PRo*** (5) We also manually curate ***MiDAS***, a subset of 96 examples from the BigSurvey (LIU et al. 2022) dataset with reference attribution for evaluating ***MiDAS-PRo***.

2 Related Works

Multi-Document Summarization. PRIMERA (Xiao et al. 2022) introduces a novel pre-training objective specifically tailored for multi-document summarization, enabling the model to effectively connect and aggregate information across multiple documents. By leveraging efficient encoder-decoder transformers, PRIMERA simplifies the processing of concatenated input documents. Graph-based models (Liao, Lebanoff, and Liu 2018; Li et al. 2020; Pasunuru et al. 2021) have also been used for multi-document summarization. However, such models frequently depend on auxiliary information, such as AMR (Abstract Meaning Representation) or discourse structure, to construct an input graph. This reliance on additional models reduces their generalizability.

Code-based reasoning and planning using LLMs. Using code for reasoning and planning has shown promising results. For instance, code-based reasoning using the "Program of Thoughts" (PoT) approach (Chen et al. 2023a) enables LLMs to solve math problems by generating code to express complex reasoning procedures, which is then executed by a program interpreter, effectively separating reasoning from computation. Similarly, Gao et al. (2023) introduces Program-Aided Language models (PAL), which utilize LLMs to convert natural language problems into executable code, offloading the computation to a Python interpreter. This method enhances accuracy in several mathematical, symbolic, and algorithmic reasoning tasks. Madaan et al. (2022) demonstrates that framing structured commonsense reasoning tasks as code generation problems enables pre-trained code LLMs to outperform natural language models, even when the tasks don't involve code, across three varied structured reasoning tasks.

3 Problem Statement and Datasets

Given a collection of n documents, which are paper abstracts of interest - ($R^a_1, R^a_2, ..., R^a_n$), and a query intent Q^I, the objective is to generate the target multi-document abstractive summary S^T. Note that in our setting, S^T contains citations to $R^a_1, R^a_2, ..., R^a_n$.

3.1 Datasets

Multi-XScience. The Multi-XScience dataset (Lu, Dong, and Charlin 2020) is a large-scale dataset created to facilitate the task of multi-document summarization, particularly in the context of scientific articles. This dataset combines information from two primary sources: arXiv.org and the Microsoft Academic Graph (MAG). The construction of Multi-XScience involves multiple stages to ensure its robustness and utility. Initially, the LaTeX source of approximately 1.3 million arXiv papers is cleaned. These papers are then aligned with their references in MAG through a series of heuristics, followed by five iterations of data cleaning interspersed with human verification. Here, Q^I is the abstract of a query paper, and S^T is the related work paragraph of the query paper. The dataset consists of $30,369$ training, $5,066$ validation, and $5,093$ test instances. On average, documents in the dataset have a length of 778.08 tokens, summaries are 116.44 tokens long, and each document has 4.42 references. Multi-XScience stands out for its high degree of abstractiveness, reflected in the significant proportion of novel n-grams in its target summaries.

MiDAS. ***MiDAS*** (Multi-Document Attribution-inclusive Summarization) is a subset of BigSurvey-MDS dataset (LIU et al. 2022), **where we annotate additional reference attribution information and paper metadata that is not present in BigSurvey-MDS dataset**. BigSurvey-MDS is a large-scale dataset designed for the task of summarizing numerous academic papers on a single research topic. For ***MiDAS***, Q^I is the title of a survey research paper (annotated by us), and S^T is the introduction paragraph of the survey paper. BigSurvey-MDS is constructed using $4,478$ survey papers from arXiv.org and their associated reference papers' abstracts. The survey papers are selected for their comprehensive coverage of various research topics. The entire dataset is divided into training, validation, and test sets in the ratio of 8:1:1. BigSurvey-MDS has an average input document length of $11,893.1$ words and 450.1 sentences. The target summaries average $1,051.7$ words and 38.8 sentences. ***MiDAS*** samples 40 examples from the training and 56 examples from the test sets of BigSurvey-MDS.

4 Proposed Approach

Our proposed approach ***MiDAS-PRo*** introduces a novel planning-cum-reasoning module using pre-trained Code LLMs in an in-context learning setting, which is carried out in 2 stages - (1) Hierarchical Reference Layout Tree (HRLT) Generation, where relative locations of references to source documents in the final summary are predicted to produce a hierarchical **plan** of the final summary. (2) Sentence-wise Entity Generation, where entities relevant to each sentence in the final summary are predicted to act as a **reasoning** scaffold for the final summary. (as shown in Fig. 1). The HRLT and sentence-wise entities so generated are added to a summarization-specific natural language prompt that is passed to a pre-trained natural language LLM to generate the final summary. Also, we use a novel GAT-based in-context example selection method (see Section 4.4).

4.1 Planning - HRLT Generation

To generate the final summary in a well-organized fashion, we first attempt at generating a layout of the references that are to appear in the final summary. The layout is hierarchical in nature, where we first to try to predict as to which references will be cited adjacent to each other/will appear in the same citation bracket, followed by predicting which

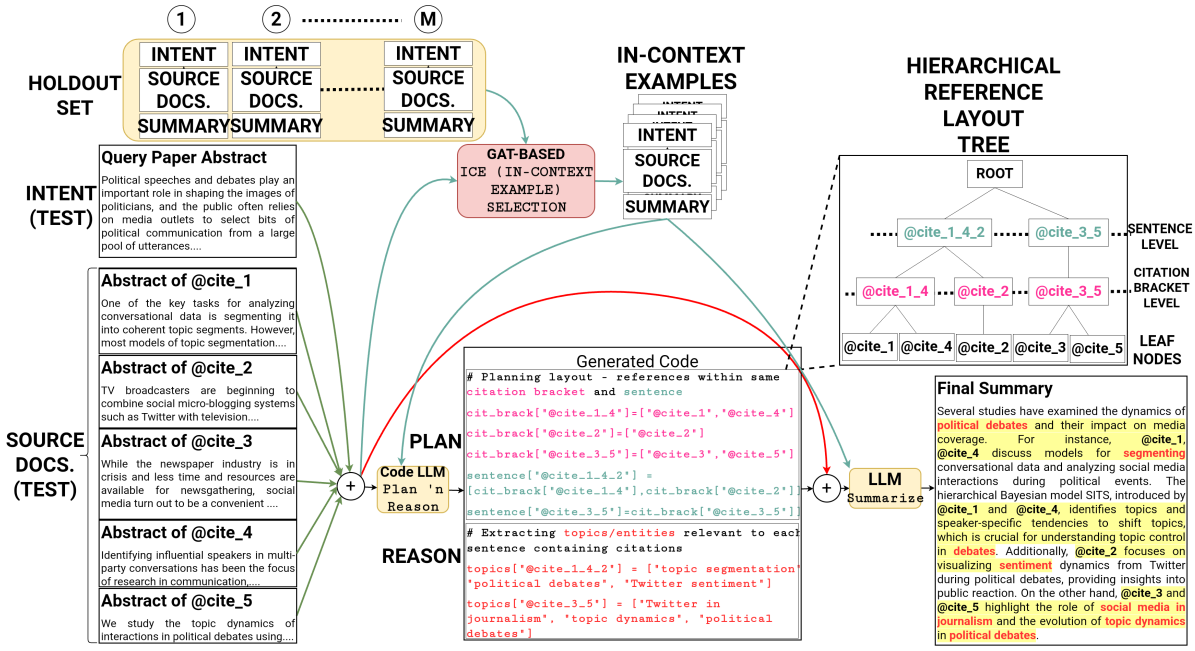

Figure 1: ***MiDAS-PRo*** Framework for Multi-Document Summarization

references will be cited in the same sentence. Such a hierarchical layout ensures that there is - (1) logical flow and association of content corresponding to the references (2) no citation in the summary that is not present in the set of source documents. We frame our problem of HRLT generation as a code completion problem similar to Madaan et al. (2022), which solves structured commonsense reasoning tasks via code completion[1]. The desired format of the generated code is shown in Fig. 1, where references within the same citation bracket are grouped together into lists to form the `cit_brack` dictionary. These groups are then further grouped based on whether the citations appear in the same sentence or not, thus populating the `sentence` dictionary. This code generation is carried out in an in-context learning setting (Brown et al. 2020) (described in Section 4.4) so that this desired format is obtained.

4.2 Reasoning - Sentence-Wise Entity Generation

Berezin and Batura (2022) has shown that named entity-based supervision improves abstractive text summarization performance. Similarly, we extract entities as a reasoning aid to improve the final summary so obtained. This is also carried out in an in-context learning setting (described in Section 4.4). For the in-context learning samples, we extract scientific entities from sentences in the ground truth summary containing references using scispaCy[2](Neumann et al. 2019). After extracting the entities, we convert it into the format of a code (as shown in Fig. 1), where we populate the dictionary `topics` sentence-wise. We name the dictionary as `topics` due to the entities extracted being topics relevant to the sentences in the final summary.

All in all, we generate the code for populating `citation_bracket`. `sentence`, and `topics` dictionaries in an in-context learning setting. These dictionaries would serve as planning, reasoning-based information when generating the summary, which is elaborated in Section 4.3. Structure of the 1-shot prompt is shown in Prompt 1.

4.3 Generating the Final Summary

To generate the final summary, the code (generated in case of the test samples and extracted in case of the in-context samples) is converted into natural language. This planning and reasoning information is appended to the intent and source documents in the in-context examples (which contain ground truth summaries at each example's end) and the test example at hand to create the complete input prompt (see the prompt structure in Prompt 2).

4.4 In-Context Example Selection using Graph-Attention Network (GAT)

The LLM is provided with task-specific examples and instructions within the same session, without updating its parameters, thus enriching the output given by the LLM. This means that the examples used in the ICL prompt should be relevant and helpful to the test example for which the task is to be solved by the LLM. For this, we use an vector embedding-based similarity approach to pick examples from a training set that have the most similar embeddings to that of the test example. We introduce a novel method to learn a single embedding to represent the collection of documents to be summarized.

[1]However, unlike Madaan et al. (2022), we use code completion to solve an intermediate task in ***MiDAS-PRo***

[2]We use the `en_ner_bionlp13cg_md` model from https://allenai.github.io/scispacy/, which is a NER model trained on the BIONLP13CG (Pyysalo et al. 2015) corpus

Algorithm 1: Structure of 1-shot Prompt used Planning and Reasoning in ***MiDAS-PRo*** for Multi-XScience

```
##Example 1:
def main():
    # Given is a dictionary of paper abstracts cited as references in the related work of a query paper, and query paper abstract-
    reference_paper_abstracts = dict()
    reference_paper_abstracts["@cite_30"]="<@cite_30 ABSTRACT>"
    reference_paper_abstracts["@cite_19"]="<@cite_19 ABSTRACT>"
    query_paper_abstract = "<ABSTRACT OF QUERY RESEARCH PAPER>"
def hierarchical_clustering():
    # Hierarchical Clustering of references within Related Work Section of query paper
    cit_brack = {} # contains lists as values that show how references are grouped within same citation bracket in Related Work of query paper
    sentence = {} # contains lists, each list contains references in a sentence in Related Work of query paper
    topics = {} # contains lists as values, each list contains topics/ entities relevant to a sentence
    cit_brack["@cite_19_30"] = ["@cite_19", "@cite_30"]
    sentence["@cite_19_30"] = [cit_brack["@cite_19_30"]]
    topics["@cite_19_30"] = ["automobiles", "economic models"]
##Example 2:
def main():
    # Given is a dictionary of paper abstracts cited as references in the related work of a query paper, and query paper abstract-
    reference_paper_abstracts = dict()
    reference_paper_abstracts["@cite_9"]="<@cite_9 ABSTRACT>"
    reference_paper_abstracts["@cite_15"]="<@cite_15 ABSTRACT>"
    query_paper_abstract = "<ABSTRACT OF QUERY RESEARCH PAPER>"
def hierarchical_clustering():
    # Hierarchical Clustering of references within Related Work Section of query paper
    cit_brack = {} # contains lists as values that show how references are grouped within same citation bracket in Related Work of query paper
    sentence = {} # contains lists, each list contains references in a sentence in Related Work of query paper
    topics = {} # contains lists as values, each list contains topics/ entities relevant to a sentence
```

Algorithm 2: Structure of 1-shot Prompt used for generating summary in ***MiDAS-PRo*** for Multi-XScience

```
##Example 1:
Paper Abstracts:
@cite_30 : <ABSTRACT OF @cite_30 PAPER>
@cite_19 : <ABSTRACT OF @cite_19 PAPER>
Query Paper Abstract:
<ABSTRACT OF QUERY RESEARCH PAPER>
Generate related work section.
@cite_19, @cite_30 are in same citation bracket (i.e., they are right next to each other) within Related work. @cite_19, @cite_30 are in same sentence within Related work - contains following entities - <TOPICS IN SENTENCE>.
Answer: <RELATED WORK>
##Example 2:
Paper Abstracts:
@cite_9 : <ABSTRACT OF @cite_9 PAPER>
@cite_15 : <ABSTRACT OF @cite_15 PAPER>
Query Paper Abstract:
<ABSTRACT OF QUERY RESEARCH PAPER>
Generate related work section.
@cite_9, @cite_15 are in same citation bracket (i.e., they are right next to each other) within Related work. @cite_9, @cite_15 are in same sentence within Related work - contains following entities - <TOPICS IN SENTENCE>.
Answer:
```

First, we create a star graph $\mathcal{G}$, where the central node v_0 (connected to all other nodes of the graph) represents the intent, and the other nodes $v_1, v_2, \ldots, v_N$ represent the N source documents. Let h_i denote the embedding of node v_i for $i = 0,\ 1,\ \ldots,\ N$. We initialize the central node embedding h_0 with the intent representation and other node embeddings h_i (for $i = 1,\ \ldots,\ N$) with embeddings of the respective source documents using Sentence-BERT (Reimers and Gurevych 2019) transformer encoder. We then implement a Graph Attention Network (GAT) (Veličković et al. 2018) to update the central node embedding h_0. The attention mechanism assigns weights α_{0i} to each edge connecting central node v_0 with a source document node v_i as follows:

$$\alpha_{0i} = \frac{exp\big(LeakyReLU\left(a^T\left[Wh_0||Wh_i\right]\right)\big)}{\sum_{j=1}^{N} exp(LeakyReLU\left(a^T\left[Wh_0||Wh_j\right]\right))} \quad (1)$$

, where a is the attention vector, W is the weight matrix, and $||$ denotes concatenation.

The central node embedding h'_0 is then updated as: $h'_0 = \sigma\left(\sum_{i=1}^{N} \alpha_{0i} W h_i\right)$, where σ is a non-linear sigmoid activation function. h'_0 encapsulates both the overall content of the source documents and the intent.

Fig. 2 shows the training pipeline for training the GAT. Pairs of document collections are sampled from the training set. For each pair of collections A and B, the collection embeddings h'^A_0 and h'^B_0 are generated by passing the corresponding star graphs $\mathcal{G}^A$ and $\mathcal{G}^B$ as inputs to the GAT

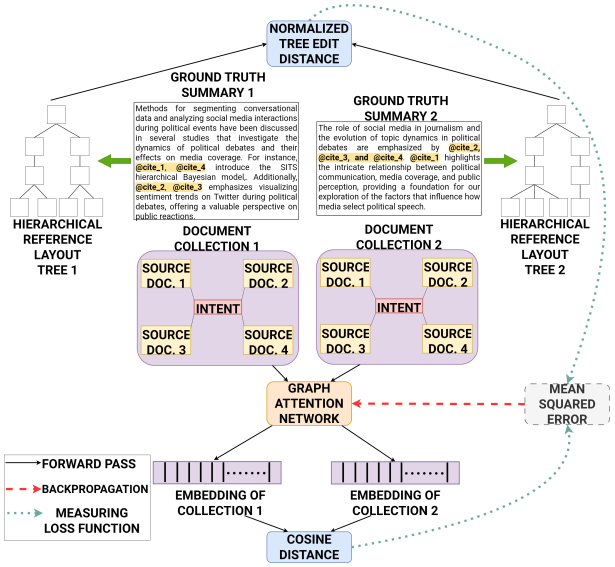

Figure 2: Training GAT module for ICE Selection

Model respectively, with θ as the GAT's trainable parameters, and the HRLTs $\mathcal{T}^A$ and $\mathcal{T}^B$ are extracted using the corresponding ground truth summaries. The mean squared error $\mathcal{L}_\theta$ between the normalized tree similarity $sim(\mathcal{T}^A, \mathcal{T}^B)$ (which is **1 - normalized tree edit distance** between the two trees) and the cosine similarity between the two collection embeddings $cos(h_0'^A, h_0'^B)$ is backpropagated to update the GAT parameters, so that the trained GAT Model is able to mimic the tree similarity metric when the HRLT is not directly available (similar to Poesia et al. (2022)). Formally, suppose $\mathcal{D}$ is a dataset of collection embeddings and associated HRLTs, with $\mathcal{D}_i = (h_0'^i, \mathcal{T}^i)$, the loss is -

$$\mathcal{L}_\theta := \mathbb{E}_{i,j\sim\mathcal{D}} \left[cos(h_0'^i, h_0'^j) - sim(\mathcal{T}^i, \mathcal{T}^j) \right]^2 \quad (2)$$

After training the GAT Model, document collections from the holdout set with embeddings closest to the test document collection's embedding are chosen as in-context examples.

5 Experiments and Results

5.1 Experimental Setup

For in-context example selection, we apply a 2-layer GAT model. The first layer consists of 8 attention heads computing 8 features (resulting in a total of 64 features) followed by a ReLU (Nair and Hinton 2010) non-linearity, while the second layer consists of a single attention head for computing 128 features, followed by a sigmoid activation function. A dropout (Srivastava et al. 2014) of $p = 0.6$ is applied to both the layers. This model is trained on $143,405$ pairs of document collections, and validated on $17,925$ pairs of document collections. We utilize a batch size of 32 on the training nodes to minimize the mean-squared error using the Adam SGD optimizer (Kingma and Ba 2014) with a learning rate of 0.02. We use an early stopping strategy on the validation loss with a patience of 100 epochs. Also, the GAT model contains 57,920 trainable parameters and 57,929 FLOPS, and hence, the amount of training compute is negligible.

We generate the intermediate code for planning and reasoning using GPT-4o (OpenAI 2024) due to its proficiency in understanding and generating code. To generate the final summary after that, we use an LLM that was pretrained on natural language instructions (e.g. Mistral-7B-Instruct (Jiang et al. 2023), Llama-3-8B-Instruct, Llama-3-70B-Instruct (Dubey et al. 2024), GPT-4o). We create a "holdout" set (which is a subset of the training set) for selecting in-context samples. For Multi-XScience, the holdout set is populated by first randomly selecting 1000 samples from the training set. The HRLT (see Fig. 1 for an example) is extracted for each training sample, and the top 50 samples are chosen as the holdout set based on the average number of children of the nodes present in the second ("sentence") and third ("citation_bracket") levels of the tree. The choice behind this heuristic is to populate the holdout set with samples that have a high number of citations grouped within citation brackets and sentences, thus providing surplus supervision in ***MiDAS-PRo***. For ***MiDAS***, 40 training set samples are used as the holdout set. All experiments using open-source models are carried out on 4 A100 GPUs, each having a VRAM of 80GB.

5.2 Baselines

We compare ***MiDAS-PRo*** applied on a natural language LLM with baselines using the same LLM and number of in-context examples without utilizing extra code-based planning and reasoning information. However, baselines differ in the way the in-context examples are chosen from holdout set, and are described as follows - (1) **Random** - In-context examples are randomly sampled from holdout set for each test sample (2) **BM25** - In-context examples are retrieved using the sparse retrieval method BM25 (Robertson, Zaragoza et al. 2009) applied on concatenated source documents and intent of all document collections present in the holdout set. For a K-shot setting ($K \geq 1$), top K samples from the holdout set based on the relevance score obtained using BM25 are selected (3) **SciBERT** - SciBERT (Beltagy, Lo, and Cohan 2019) is used to encode intents for holdout set samples and the test sample (documents in document collection are not included while encoding due to limited sequence length of 512 tokens), and encoded embeddings of samples from holdout set that are most similar to that of the test sample are selected based on cosine similarity. We also compare ***MiDAS-PRo*** with other Multi-Doc Summarization baselines such as PEGASUS (Zhang et al. 2020), BART (Lewis et al. 2020), and PRIMERA (Xiao et al. 2022).

5.3 Automated Evaluation Metrics

To evaluate content of the generated summary, we evaluate ***MiDAS-PRo*** against baselines using conventional text generation metrics such as lexical metrics of BLEU (Papineni et al. 2002) and ROUGE-L (Lin 2004) which match surface-level word sequences between predicted and ground truth summaries, and semantic similarity metric of BERTScore (Zhang* et al. 2020) which matches actual meaning between predicted and ground truth summaries.

Natural Lang LLM	ICE Selection	Summarization Evaluation Metrics			Clustering-based Metrics					
		ROUGE-L	BLEU	BERTScore	NMI(CB)	AMI(CB)	NMI(S)	AMI(S)	AVG NMI	AVG AMI
Mistral-7B-Instruct	Random	0.1589	0.0798	0.8464	0.305	0.095	0.317	0.136	0.311	0.1155
	BM25	0.1584	0.0802	0.8452	0.334	0.139	0.344	0.177	0.339	0.158
	SciBERT	0.162	0.0908	0.847	0.317	0.136	0.354	0.208	0.3355	0.172
	MiDAS-PRo	**0.1646** (+1.6%)	**0.1023** (+12.7%)	**0.851** (+0.5%)	**0.355** (+6.3%)	**0.184** (+32.4%)	**0.372** (+5.1%)	**0.235** (+13%)	**0.3635** (+7.2%)	**0.2095** (+21.8%)
Llama-3-8B-Instruct	Random	0.1506	0.0708	0.844	0.3	0.099	0.309	0.139	0.304	0.119
	BM25	0.1492	0.0717	0.8422	0.379	0.21	0.376	0.231	0.378	0.22
	SciBERT	0.1512	0.0731	0.8435	0.353	0.175	0.362	0.214	0.358	0.194
	MiDAS-PRo	**0.1542** (+2%)	**0.0917** (+25.4%)	**0.8461** (+0.2%)	**0.42** (+10.8%)	**0.32** (+52.4%)	**0.422** (+12.2%)	**0.341** (+47.6%)	**0.421** (+11.5%)	**0.33** (+49.9%)
Llama-3-70B-Instruct	Random	0.1571	0.068	0.8473	0.258	0.024	0.239	0.031	0.2485	0.0275
	BM25	0.1551	0.0583	0.8432	0.254	0.019	0.237	0.027	0.2455	0.023
	SciBERT	0.1565	0.0645	0.8459	0.262	0.031	0.24	0.034	0.251	0.0325
	MiDAS-PRo	**0.159** (+1.2%)	**0.0889** (+30.7%)	**0.8492** (+0.2%)	**0.281** (+7.2%)	**0.092** (+196.8%)	**0.27** (+12.5%)	**0.106** (+211.8%)	**0.2755** (+9.8%)	**0.099** (+204.6%)
GPT-4o	Random	0.1495	0.0748	0.8507	0.271	0.049	0.238	0.043	0.254	0.046
	BM25	0.1518	0.0839	0.852	0.292	0.092	0.262	0.089	0.277	0.09
	SciBERT	0.152	0.0792	0.8516	0.287	0.078	0.26	0.08	0.274	0.079
	MiDAS-PRo	**0.1556** (+2.4%)	**0.0993** (+18.4%)	**0.8535** (+0.2%)	**0.391** (+33.9%)	**0.263** (+185.9%)	**0.354** (+35.1%)	**0.245** (+175.3%)	**0.372** (+34.5%)	**0.254** (+180.7%)

Table 1: 1-Shot Results of Natural Language LLMs combined with different ICE Selection Methods compared to ***MiDAS-PRo*** on Multi-XScience. Underlined values correspond to metrics where ***MiDAS-PRo*** gives significant improvement ($p < 0.05$)

ICE Selection	Summarization Evaluation Metrics			Clustering-based Metrics					
	ROUGE-L	BLEU	BERTScore	NMI(CB)	AMI(CB)	NMI(S)	AMI(S)	AVG NMI	AVG AMI
Random	**0.1519**	0.0535	0.8116	**0.542**	**0.097**	**0.448**	**0.07**	**0.495**	**0.084**
BM25	0.151	0.0555	0.8113	0.504	0.04	*0.423*	0.03	*0.464*	0.035
SciBERT	0.1511	*0.0571*	*0.8119*	0.512	0.068	0.417	0.026	*0.464*	0.047
MiDAS-PRo	*0.1516*	**0.0692**	**0.8125**	*0.514*	*0.093*	0.414	*0.065*	*0.464*	*0.079*

Table 2: 1-Shot In-Context Results of GPT-4o combined with different In-Context Example (ICE) Selection Methods compared to ***MiDAS-PRo*** on ***MiDAS***. Results in **bold** and *italic* are the best and the second-best results respectively

To evaluate organization of references to source documents in the summary (i.e. attribution organization), we regard the grouping of citations within citation bracket/adjacent citations and sentences as two levels of clustering, and apply the following metrics that measure mutual information between predicted and ground truth clusterings at two levels - Normalized Mutual Information (NMI) (Danon et al. 2005), which normalizes the Mutual Information (MI) score, scaling results between 0 (no mutual information) and 1 (perfect correlation), and Adjusted Mutual Information (AMI) (Vinh, Epps, and Bailey 2009), which adjusts MI score to account for chance.We represent NMI and AMI at citation bracket level as NMI(CB) and AMI(CB), and at sentence level as NMI(S) and AMI(S) respectively.

5.4 1-shot Results

Table 1 shows 1-shot results using 4 LLMs, along with performance improvement using ***MiDAS-PRo*** over best performing baseline corresponding to each metric on Multi-XScience. We infer - (1) ***MiDAS-PRo*** applied on 4 LLMs gives improvement over all baseline methods on all summarization and clustering-based metrics. Improvements are atleast 1% in summarization on average, and atleast 7% and 21% in average NMI and AMI Scores respectively (2) Percentage improvements in average summarization and AMI scores increase with model size for open-source models (3) ***MiDAS-PRo*** when applied on GPT-4o performs best semantically as per BERTScore (4) Interestingly, using ***MiDAS-PRo*** on Mistral-7B-Instruct and Llama-3-8B-Instruct gives best summarization and clustering-based metrics respectively, even though both models are smaller than Llama-3-70B and have a fraction of context window of GPT-4o. This shows that increasing number of model parameters and context window size does not enhance multi-doc summarization for scientific articles, underlining difficulty of the task.

Table 2 shows 1-shot results using GPT-4o compared to baselines on ***MiDAS*** (as ***MiDAS*** contains long-context samples, which needs a model with a long context window). ***MiDAS-PRo*** gives best BLEU and BERTScore, and second-best ROUGE-L and average clustering-based metrics.

5.5 Few-shot results with GPT-4o

Table 3 shows K-Shot In-Context Results ($K = 1, 3, 5, 10$) of GPT-4o (GPT-4o has long context window of $128,000$ tokens, which could accommodate multiple long input samples) combined with different In-Context Example Selection Methods on Multi-XScience. We can infer that - (1) ***MiDAS-PRo*** performs better than all baselines w.r.t all metrics across varying number of in-context examples (2) The average summarization and clustering-based metrics keep improving as the number of in-context examples keeps increasing for ***MiDAS-PRo*** and most baselines, as the amount of supervision increases (3) Percentage improvement in summarization metrics increases when number of ICEs increases from 1 to 5 (4) Percentage improvement in avg. clustering-based metrics generally decreases with increase in number of ICEs (5) BM25 performs best among all baselines for almost all metrics across different number of in-context examples.

No. of ICE	ICE Selection	Summarization Evaluation Metrics			Clustering-based Metrics					
		ROUGE-L	BLEU	BERTScore	NMI(CB)	AMI(CB)	NMI(S)	AMI(S)	AVG NMI	AVG AMI
1	Random	0.1495	0.0748	0.8507	0.271	0.049	0.238	0.043	0.254	0.046
	BM25	0.1518	0.0839	0.852	0.292	0.092	0.262	0.089	0.277	0.09
	SciBERT	0.152	0.0792	0.8516	0.287	0.078	0.26	0.08	0.274	0.079
	MiDAS-PRo	**0.1556 (+2.4%)**	**0.0993 (+18.4%)**	**0.8535 (+0.2%)**	**0.391 (+33.9%)**	**0.263 (+185.9%)**	**0.354 (+35.1%)**	**0.245 (+175.3%)**	**0.372 (+34.5%)**	**0.254 (+180.7%)**
3	Random	0.1505	0.0766	0.8514	0.274	0.051	0.243	0.049	0.258	0.05
	BM25	0.1526	0.087	0.8525	0.302	0.105	0.277	0.107	0.29	0.106
	SciBERT	0.1522	0.0826	0.8522	0.29	0.086	0.264	0.09	0.277	0.088
	MiDAS-PRo	**0.1578 (+3.4%)**	**0.1081 (+24.2%)**	**0.855 (+0.3%)**	**0.392 (+29.8%)**	**0.277 (+163.8%)**	**0.359 (+29.6%)**	**0.257 (+140.2%)**	**0.376 (+29.7%)**	**0.267 (+151.9%)**
5	Random	0.151	0.0769	0.8514	0.272	0.05	0.243	0.047	0.258	0.048
	BM25	0.1531	0.0882	0.8526	0.31	0.115	0.283	0.119	0.296	0.117
	SciBERT	0.1531	0.0848	0.8525	0.2987	0.1011	0.2748	0.1047	0.2868	0.1029
	MiDAS-PRo	**0.1587 (+3.7%)**	**0.1105 (+25.3%)**	**0.8554 (+0.3%)**	**0.3944 (+27.2%)**	**0.283 (+146.1%)**	**0.3575 (+26.3%)**	**0.2597 (+118.2%)**	**0.376 (+27%)**	**0.2714 (+132%)**
10	Random	0.1515	0.0782	0.8517	0.277	0.055	0.253	0.061	0.265	0.058
	BM25	0.1543	0.089	0.853	0.311	0.119	0.286	0.123	0.298	0.121
	SciBERT	0.1541	0.087	0.8529	0.3	0.104	0.273	0.109	0.286	0.106
	MiDAS-PRo	**0.1588 (+2.9%)**	**0.1113 (+25.1%)**	**0.8555 (+0.3%)**	**0.3896 (+25.3%)**	**0.2767 (+132.5%)**	**0.3766 (+31.7%)**	**0.2788 (+126.7%)**	**0.3831 (+28.6%)**	**0.2778 (+129.6%)**

Table 3: K-Shot In-Context Results ($K = 1, 3, 5, 10$) of GPT-4o combined with different ICE Selection Methods compared to ***MiDAS-PRo*** on Multi-XScience. ***MiDAS-PRo*** gives significant improvement ($p < 0.05$) compared to baselines

Variant	Summarization Evaluation Metrics			Clustering-based Metrics					
	ROUGE-L	BLEU	BERTScore	NMI(CB)	AMI(CB)	NMI(S)	AMI(S)	AVG NMI	AVG AMI
ONLY-CB	0.153	0.0894	0.8459	0.407	0.301	0.404	0.317	0.406	0.309
ONLY-SENT	**0.1546**	0.09	**0.8466**	0.382	0.274	0.394	0.306	0.388	0.29
ONLY-PLAN	0.154	0.0913	0.8462	0.411	0.307	0.411	0.329	0.411	0.318
SENT+REASON	**0.1546**	0.09	**0.8466**	0.389	0.281	0.39	0.302	0.39	0.292
MiDAS-PRo	0.1542	**0.0917**	0.8461	**0.42**	**0.32**	**0.422**	**0.341**	**0.421**	**0.33**

Table 4: Ablation Analysis of ***MiDAS-PRo*** when applied on Llama-3-8B-Instruct (GPT-4o is used for generating the code)

5.6 Comparison with other baselines

Table 5 shows that ***MiDAS-PRo*** outperforms 3 0-shot and 10-shot fine-tuned baselines when using similar number of ICE, and is comparable to 100-shot fine-tuned baselines.

	No. of ICE/training samples		
Model	**0/1***	**10**	**100**
PEGASUS	0.1504	0.1549	0.1589
BART	0.0861	0.1490	0.1645
PRIMERA	0.1409	0.1529	0.1684
MiDAS-PRo	**0.1556**	**0.1588**	-

* ***MiDAS-PRo*** needs $\geq$ 1 ICE, compared with 0-shot baselines

Table 5: Comparison of ***MiDAS-PRo*** (with GPT-4o backbone) with other Multi-Doc Summarization Baselines.

5.7 Ablation Analysis

We compare ***MiDAS-PRo*** with - (1) **ONLY-CB.** generating only citation bracket level info. of HRLT without Reasoning (2) **ONLY-SENT.** generating only sentence-level info. of HRLT without Reasoning (3) **ONLY-PLAN.** Planning without Reasoning (4) **SENT + REASON.** generating only sentence-level info. of HRLT with Reasoning. Table 4 shows ablations when ***MiDAS-PRo*** is applied on LlaMA-3-8B-Instruct with GPT-4o for generating code. We infer ***MiDAS-PRo*** outperforms variants in BLEU, avg. summarization scores, all clustering-based metrics, underlining importance of hierarchical planning, reasoning in ***MiDAS-PRo***.

	Relevance	Fluency	Informativeness	Attr. Corr.	Non-Red.	Avg. Score
Random	4.5	**4.67**	4.42	4.42	**4.92**	4.58
BM25	4.75	4.5	**4.67**	**4.5**	4.75	4.63
SciBERT	4.58	**4.67**	4.5	4.33	4.75	4.57
MiDAS-PRo	**4.83**	**4.67**	*4.58*	**4.5**	*4.83*	**4.68**

Table 6: Human Evaluation of ***MiDAS-PRo***

5.8 Human Evaluation

To perform fine-grained evaluation, we ask a scientific expert to evaluate summaries generated by ***MiDAS-PRo*** and 3 baselines for 12 random Multi-XScience Test Samples, on 5 aspects - (1) **Relevance** - how relevant is summary to query abstract? (2) **Fluency** - how good is content organization and flow of summary? (3) **Informativeness** - how good is coverage of reference abstracts' content within summary? (4) **Attribution Correctness** - how much of summary content corresponding to a reference is derived from reference abstract (5) **Non-Redundancy** - Is there less repetition in summary?

Table 6 shows - ***MiDAS-PRo*** has best Relevance, Fluency, Attribution Correctness, and average score.

6 Discussion

This work shows effectiveness of using LLMs for planning and reasoning in multi-document summarization with attribution. Our approach ***MiDAS-PRo*** leverages planning and reasoning to address challenges of synthesizing information from multiple sources. Experiments validate superiority of ***MiDAS-PRo***. Notably, code-based reasoning with LLMs enhances interpretability of generated summaries.

References

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. Hong Kong, China: Association for Computational Linguistics.

Berezin, S.; and Batura, T. 2022. Named Entity Inclusion in Abstractive Text Summarization. In Cohan, A.; Feigenblat, G.; Freitag, D.; Ghosal, T.; Herrmannova, D.; Knoth, P.; Lo, K.; Mayr, P.; Shmueli-Scheuer, M.; de Waard, A.; and Wang, L. L., eds., *Proceedings of the Third Workshop on Scholarly Document Processing*, 158–162. Gyeongju, Republic of Korea: Association for Computational Linguistics.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2023a. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Transactions on Machine Learning Research*.

Chen, X.; Chan, Z.; Gao, S.; Yu, M.-H.; Zhao, D.; and Yan, R. 2019. Learning towards Abstractive Timeline Summarization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 4939–4945. International Joint Conferences on Artificial Intelligence Organization.

Chen, X.; Li, M.; Gao, S.; Chan, Z.; Zhao, D.; Gao, X.; Zhang, X.; and Yan, R. 2023b. Follow the Timeline! Generating an Abstractive and Extractive Timeline Summary in Chronological Order. *ACM Trans. Inf. Syst.*, 41(1).

Danon, L.; Diaz-Guilera, A.; Duch, J.; and Arenas, A. 2005. Comparing community structure identification. *Journal of statistical mechanics: Theory and experiment*, 2005(09): P09008.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

El-Kassas, W. S.; Salama, C. R.; Rafea, A. A.; and Mohamed, H. K. 2021. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165: 113679.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, 10764–10799. PMLR.

Grail, Q.; Perez, J.; and Gaussier, E. 2021. Globalizing BERT-based transformer architectures for long document summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1792–1810.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jin, H.; Zhang, Y.; Meng, D.; Wang, J.; and Tan, J. 2024. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kurisinkel, L. J.; and Chen, N. F. 2023. LLM Based Multi-Document Summarization Exploiting Main-Event Biased Monotone Submodular Content Extraction. *arXiv preprint arXiv:2310.03414*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Li, W.; Xiao, X.; Liu, J.; Wu, H.; Wang, H.; and Du, J. 2020. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6232–6243. Online: Association for Computational Linguistics.

Liao, K.; Lebanoff, L.; and Liu, F. 2018. Abstract Meaning Representation for Multi-Document Summarization. In Bender, E. M.; Derczynski, L.; and Isabelle, P., eds., *Proceedings of the 27th International Conference on Computational Linguistics*, 1178–1190. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

LIU, S.; Cao, J.; Yang, R.; and Wen, Z. 2022. Generating a Structured Summary of Numerous Academic Papers: Dataset and Method. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 4259–4265. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Lu, Y.; Dong, Y.; and Charlin, L. 2020. Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8068–8074. Online: Association for Computational Linguistics.

Madaan, A.; Zhou, S.; Alon, U.; Yang, Y.; and Neubig, G. 2022. Language Models of Code are Few-Shot Commonsense Learners. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1384–1403. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Neumann, M.; King, D.; Beltagy, I.; and Ammar, W. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In Demner-Fushman, D.; Cohen, K. B.; Ananiadou, S.; and Tsujii, J., eds., *Proceedings of the 18th BioNLP Workshop and Shared Task*, 319–327. Florence, Italy: Association for Computational Linguistics.

OpenAI. 2024. Hello, GPT-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-08-07.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Isabelle, P.; Charniak, E.; and Lin, D., eds., *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Pasunuru, R.; Liu, M.; Bansal, M.; Ravi, S.; and Dreyer, M. 2021. Efficiently Summarizing Text and Graph Encodings of Multi-Document Clusters. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4768–4779. Online: Association for Computational Linguistics.

Poesia, G.; Polozov, A.; Le, V.; Tiwari, A.; Soares, G.; Meek, C.; and Gulwani, S. 2022. Synchromesh: Reliable Code Generation from Pre-trained Language Models. In *International Conference on Learning Representations*.

Pyysalo, S.; Ohta, T.; Rak, R.; Rowley, A.; Chun, H.-W.; Jung, S.-J.; Choi, S.-P.; Tsujii, J.; and Ananiadou, S. 2015. Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013. *BMC bioinformatics*, 16: 1–19.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Shi, Z.; Gao, S.; Zhang, Z.; Chen, X.; Chen, Z.; Ren, P.; and Ren, Z. 2023. Towards a Unified Framework for Reference Retrieval and Related Work Generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Song, J.; Chim, J.; Tsakalidis, A.; Ive, J.; Atzil-Slonim, D.; and Liakata, M. 2024. Clinically meaningful timeline summarisation in social media for mental health monitoring. *arXiv preprint arXiv:2401.16240*.

Song, L.; Zhang, Y.; Wang, Z.; and Gildea, D. 2018. A graph-to-sequence model for AMR-to-text generation. *arXiv preprint arXiv:1805.02473*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Vinh, N. X.; Epps, J.; and Bailey, J. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 1073–1080. New York, NY, USA: Association for Computing Machinery. ISBN 9781605585161.

Wang, P.; Li, S.; Liu, S.; Tang, J.; and Wang, T. 2023. Plan and generate: Explicit and implicit variational augmentation for multi-document summarization of scientific articles. *Information Processing & Management*, 60(4): 103409.

Wiseman, S.; Shieber, S. M.; and Rush, A. M. 2018. Learning Neural Templates for Text Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3174–3187.

Xiao, W.; Beltagy, I.; Carenini, G.; and Cohan, A. 2022. PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5245–5263. Dublin, Ireland: Association for Computational Linguistics.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, 11328–11339. PMLR.

Zhang*, T.; Kishore*, V.; Wu*, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Zhang, Y.; Wang, X.; Xu, Z.; Yu, Q.; Yuille, A.; and Xu, D. 2020. When radiology report generation meets knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12910–12917.

Zhang, Z.; Elfardy, H.; Dreyer, M.; Small, K.; Ji, H.; and Bansal, M. 2023. Enhancing Multi-Document Summarization with Cross-Document Graph-based Information Extraction. In Vlachos, A.; and Augenstein, I., eds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 1696–1707. Dubrovnik, Croatia: Association for Computational Linguistics.