Building Intelligent and Reliable Summarization Systems

By

HAOPENG ZHANG
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

_____

Jiawei Zhang, Chair

_____

Xin Liu

_____

Hamed Pirsiavash

Committee in Charge

2024

Dedicated to Xiaoxuan, my anchor and my light.

In loving memory of my grandfather Xingya Wang.

Contents

Abstract

Data, in various formats, surrounds us everywhere in our daily lives, such as education, entertainment, and media. Living in the era of big data, the massive amount of web textual data has grown exponentially over the past decade. This leads to the problem of information overload, where an individual is exposed to more information than they could process. Thus, the need for an automatic text summarization (ATS) system emerges, which could transform this vast raw information into key points in the form of smaller, digestible pieces automatically.

ATS systems operate by extracting or generating a concise and readable summary while preserving salient information from the original documents. Developing intelligent systems that can produce concise, fluent, and reliable summaries has been a long-standing goal in natural language processing (NLP). Significant progress has been made in recent years, thanks to breakthroughs like pre-trained language models such as BERT [**19**] and GPT [**11**]. However, text summarization remains a complex and multifaceted task. Similar to the cognitive process humans undertake when crafting summaries, text summarization requires the machine to first semantically understand the contents of a document, then identify and extract salient information from the document, and finally generate an accurate and faithful summary.

This dissertation presents several distinct approaches to tackle the three critical steps of building ATS systems. Specifically, I first present my work to improve the modeling of long documents for extractive summarization. I introduce model HEGEL, a hypergraph neural network for long document summarization that captures high-order cross-sentence relations. HEGEL updates and learns effective sentence representations with hypergraph transformer layers and fuses different types of sentence dependencies, including latent topics, keywords, coreference, and section structure. Extensive experiments on two benchmark datasets demonstrate the effectiveness and efficiency of HEGEL in long document modeling and extractive summarization.

Then I move on to the holistic extraction of salient information from documents. To address the limitation of individual sentence label prediction in existing extractive summarization systems, I propose a novel paradigm for extractive summarization named DiffuSum. DiffuSum directly generates the desired summary sentence representations with diffusion models and extracts sentences based on sentence representation matching. Additionally, DiffuSum jointly optimizes a contrastive sentence encoder with a matching loss for sentence representation alignment and a multi-class contrastive loss for representation diversity. On the other hand, I also introduce a new holistic framework for unsupervised multi-document extractive summarization. The method incorporates the holistic beam search inference method associated with the holistic measurements, named Subset Representative Index (SRI). SRI balances the importance and diversity of a subset of sentences from the source documents and can be calculated in unsupervised and adaptive manners.

Next, I demonstrate my work on improving the quality and faithfulness of generated summaries. While text summarization systems have made significant progress in recent years, they typically generate summaries in one single step. However, the one-shot summarization setting is sometimes inadequate, as the generated summary may contain hallucinations or overlook essential details related to the reader's interests. To address this, I propose SummIt, an iterative text summarization framework based on large language models (LLMs) like ChatGPT. SummIt enables the model to refine the generated summary iteratively through self-evaluation and feedback, resembling humans' iterative process when drafting and revising summaries. Furthermore, I explore the potential benefits of integrating knowledge and topic extractors into the framework to enhance summary faithfulness and controllability. Both automatic evaluation and human studies are conducted on three benchmark summarization datasets to validate the effectiveness of the iterative refinements and to identify potential issues of over-correction.

Finally, as the emergence of large language models reshapes NLP research, I present a thorough evaluation of ChatGPT's performance on extractive summarization and compare

it with traditional fine-tuning methods on various benchmark datasets. The experimental analysis reveals that ChatGPT exhibits inferior extractive summarization performance in terms of ROUGE scores compared to existing supervised systems, while achieving higher performance based on LLM-based evaluation metrics. I also explore the effectiveness of in-context learning and chain-of-thought reasoning for enhancing its performance and propose an extract-then-generate pipeline with ChatGPT, which could yield significant performance improvements over abstractive baselines in terms of summary faithfulness. These observations highlight potential directions for enhancing ChatGPT's capabilities in faithful summarization using two-stage approaches.

In summary, by demonstrating and examining these systems and solutions, I aim to highlight the three critical yet challenging steps in building intelligent and reliable summarization systems, which are also crucial steps towards advancing the design of a more powerful and trustworthy AI assistant. I hope future research endeavors will continue to advance along these directions.

Acknowledgments

I've never considered myself a determined person, so when I decided to quit my electrical engineering Ph.D. in 2019, I couldn't have imagined that I'd end up completing a Ph.D. in NLP and writing this computer science dissertation by now. The journey, which started at Florida State University and concluded at UC Davis, has been challenging yet incredibly rewarding. I want to take this opportunity to express my deepest gratitude to the individuals who played a pivotal role in making this Ph.D. a reality.

First and foremost, I extend my heartfelt gratitude to my advisor, Dr. Jiawei Zhang, for his exceptional guidance, mentorship, and belief in my potential. Our paths crossed when I had just quit my ECE Ph.D., feeling lost and confused about my research journey. Jiawei always gave me the freedom to pursue research topics that truly interested me and exhibited boundless patience. He is a true researcher, serving as an exemplary role model from whom I continually learn. His guidance has been crucial in shaping this dissertation.

Next, I would also like to thank my dissertation committee members, Dr. Xin Liu and Dr. Hamed Pirsiavash, for their invaluable guidance. Additionally, I extend my gratitude to Dr. Ian Davidson and Dr. Joshua McCoy, who served on my qualification exam committee.

I had the privilege of completing four industry research internships throughout my Ph.D. journey, which added vibrant colors and enriched my experience. I am deeply grateful to all the mentors and colleagues I had the pleasure of meeting during these experiences. Notably, I want to express my special gratitude to Dr. Semih Yavuz, my mentor during my first-ever research internship at Salesforce Research. I was a newcomer to NLP at that time, unaware even of basic concepts like 'teacher forcing.' Semih patiently guided me, explaining every detail with responsiveness and kindness. His mentorship has been an invaluable source of learning, consistently benefiting me throughout my entire Ph.D. journey.

Ph.D. is a long, struggling, and lonely journey, but I was also fortunate to have crossed paths with so many brilliant friends along the way. Their support made the journey joyful and full of happy memories. I would like to thank Yuxiang, Lin, Xiao, and Zizhong from

our IFM Lab; Yixin, Bing, Bowen, Jiyang, and Yili from FSU; Zijian and Taiming from UC Davis; Ye and Man, whom I met at Salesforce; Hang, Xianjun, Pengshan, and Qing, whom I met at Tencent; Jin, Hayate, and Chen, whom I met at Megagons. Your presence has left an indelible mark on my journey, and I cherish all the memories we've shared. Notably, Xiao has also been a brilliant collaborator to me. We worked together on multiple projects, resulting in five publications. Wishing you all the best for the rest of your Ph.D. journey.

I am deeply indebted to my parents, Hong Wang and Dong Zhang, for their unwavering love, understanding, and support. They took care of and protected me to the best of their abilities, and their presence comforts me all the time. I haven't been able to return home since 2019, and I am really missing you. Hope to reunite and see you soon. I also would like to memorialize my grandfather, Xingya Wang, who passed away during my Ph.D. I can't return his unconditional love and care anymore, but he will remain cherished in my heart forever.

Lastly but most importantly, a special place in my heart is reserved for my beloved wife, Xiaoxuan Yang. You are my anchor, my sun, my soulmate, and the love of my life. We got married right before this Ph.D. journey, and I wholeheartedly acknowledge that I wouldn't have made it this far without your presence. You complete me, fill my heart, and illuminate my world. My soul never feels lonely after having you. Our beloved cat, Qiyue Zhang, has consistently been an angel, infusing this journey with joy.

Sincerely,

Haopeng Zhang

CHAPTER 1

Introduction

Living in the era of big data, the massive amount of textual data on the web has grown exponentially over the past decade with the advent of the Internet. This leads to the problem of information overload, where individuals are exposed to more information than they can effectively process. Information overload can subsequently lead to poor data retention, diminished mental energy, and decreased productivity.

To address this problem, Natural Language Processing (NLP) research has focused on building systems that can automatically analyze documents and assist users in digesting information. Text summarization is one core technique that aids users in navigating online documents efficiently by reducing their length and condensing them into short summaries. It refers to the process of distilling the most important information from a document (or a cluster of documents) to produce an abridged version.

Text summarization approaches could generally be categorized into different groups based on input and output formats, as illustrated in Figure 1.1.

- **Single-document vs. Multi-document vs. Query-focused:** Text summarization approaches could be divided into single-document summarization (SDS), multi-document summarization (MDS), and query-focused summarization (QFS) based on whether the input is a single document, a cluster of documents, or a document with user-specified queries.
- **Extractive vs. Abstractive vs. Hybrid:** Text summarization approaches could also be divided into extractive approaches, abstractive approaches, and hybrid approaches based on whether the output summary is created by extracting sentences

from the original documents, generated word by word from scratch with novel words, or a fusion of these two methods.

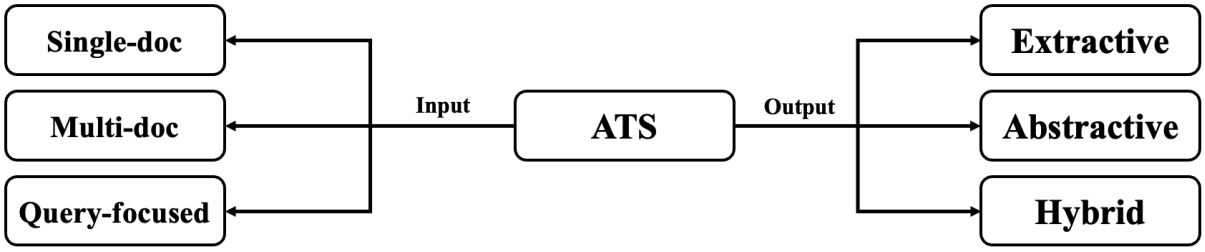

FIGURE 1.1. Categorization of text summarization approaches based on input and output formats.

Developing automatic text summarization (ATS) systems that can produce concise, fluent, and reliable summaries automatically has been a long-standing goal in NLP research. Significant progress has been made in recent years, thanks to breakthroughs like deep neural networks and pre-trained language models. However, text summarization remains a complex and multifaceted task. Similar to the cognitive process humans undertake when crafting summaries, automatic text summarization requires the machine to first semantically understand the contents of a document, then identify and extract salient information from the document, and finally generate an accurate and faithful summary.

The development progress of ATS has experienced several paradigm shifts with technological breakthroughs like term frequency-inverse document frequency (TF-IDF), deep neural networks, and Transformers [**105**]. As shown in Figure 1.2, the summarization paradigm could be generally categorized into four phases: statistical method phase, deep learning phase, pre-trained language model (PLM) phase, and large language model (LLM) phase. Here I will briefly introduce some representative work in each phase:

- **Statistical Methods:** In the early stages of summarization systems, the focus was primarily on extractive methods relying on statistical approaches. These included frequency-based methods, heuristic-based methods, graph-based methods, and cluster-based methods. Specifically, Maximal Marginal Relevance (MMR) was

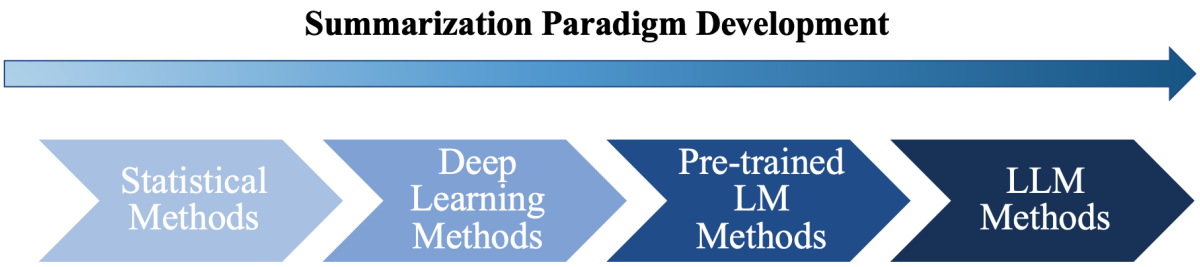

FIGURE 1.2. A roadmap illustrating the shift between summarization system paradigms.

introduced as a greedy approach to combine sentence relevance with information novelty [**12**]. Later, inspired by the PageRank algorithm [**10**], LexRank [**26**] and TextRank [**75**] were proposed to model the task of extractive summarization as identifying the most central nodes in a graph that represents the input document(s).

Furthermore, the sentence selection problem in extractive summarization was formulated as a constrained optimization problem to obtain a globally optimal solution. This was addressed with methods such as Integer Linear Programming (ILP) [**74**] or submodular optimization [**61**].

- **Deep Learning Methods:** Recent advances in deep neural networks have dramatically boosted progress in various tasks in NLP, especially when operating on large-scale text corpora. With the development of word embedding techniques [**76**, **84**], recent work typically represents text data as a sequence of tokens and learns effective sentence and document representations with deep neural network models, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks [**40**], and graph neural networks (GNNs).

  Specifically, researchers have applied RNNs [**77**], reinforcement learning(RL) [**80**], and graph neural networks [**118**] for extractive summarization. For abstractive summarization, RNNs [**78**] and pointer generation networks [**99**] have also been utilized.

- **Pre-trained Language Model Methods:** With the advent of Transformer-based architectures [**105**] and powerful large-scale pre-trained language models, ATS systems have witnessed a substantial leap in performance in the PLM phase. ATS systems in this phase commonly adapt PLM checkpoints that are trained self-supervisedly on large-scale corpora and continue to fine-tune the PLM on domain-specific training sets. BERT [**19**] marks the start of this phase and is the first widely adopted PLM that can be used for both extractive and abstractive summarization with the encoder-only architecture [**66**].

  Researchers in this phase emphasize abstractive summarization with encoder-decoder architecture models like T5 [**89**] and BART [**54**] as the backbone. In addition, researchers have also developed summarization-specific PLMs like PEGASUS [**131**], long-context PLMs like LED [**8**], and multi-document summarization PLMs like PRIMERA [**112**].
- **Large Language Model Methods:** Recently, the emergence of LLMs has reshaped both academic NLP research and industrial landscapes due to their remarkable capacity to understand, analyze, and generate texts. ATS system research now highly focuses on LLM-based abstractive summarization and is mostly built under zero-shot or few-shot settings instead of the fine-tuning step previously [**11**, **88**]. Researchers have also found that LLMs' summary outputs are preferred by human annotators despite lower automatic metric scores [**34**], and news summaries generated by LLMs are already similar to those created by humans [**133**].

In this dissertation, I will present several ATS approaches to address challenges in building intelligent and reliable summarization systems. This dissertation will address questions from different perspectives, covering both extractive and abstractive summarization, spanning from the deep learning paradigm phase to the LLM paradigm phase. Before diving into the details of what this dissertation is about, I will first introduce some preliminaries and problem formulations.

1.1. Background

This section delves into foundational concepts and the research landscape essential for understanding the development of modern ATS systems. This includes an overview of text summarization formulations, an introduction to language models, and the fundamentals of graph neural networks.

1.1.1. Text Summarization.

Text summarization is indeed one of the most important tasks in natural language processing. Its objective is to condense a piece of text while preserving its key information and main points. Here, we will cover the general problem formulation and common metrics used to evaluate summary quality:

- **Extractive Summarization:** Extractive summarization outputs a summary by identifying and directly extracting key sentences from the source document. Without loss of generality, we will formulate the summarization task for a single document here. Formally, given a document with n sentences as $D = \{s_1^d, s_2^d, ..., s_n^d\}$, extractive ATS system aims to form a $m(m \ll n)$ sentences summary $S = \{s_1^s, s_2^s, ..., s_m^s\}$ by directly extracting sentences from the source document.

  Most existing approaches formulate extractive summarization as a sequence labeling problem and assign each sentence a $\{0, 1\}$ label. Here, label 1 indicates that the sentence will be included in summary S, while label 0 indicates that the sentence is not salient and will be ignored. However, extractive ground-truth labels (ORACLE) are rarely available since most existing benchmark datasets use human-written summaries as gold summaries. Thus, it is very common to use a greedy algorithm to generate a sub-optimal ORACLE consisting of multiple sentences which maximize the ROUGE-2 score against the gold summary following [**77**]. Summaries created in the extractive manner are grammatically correct and faithful to the source document, but they could suffer from incoherence and redundancy problems.

- **Abstractive Summarization:** Abstractive summarization is typically formulated as a sequence-to-sequence problem, handled with the encoder-decoder neural architecture. Formally, abstractive ATS systems aim to generate a sequence of summary words S, conditioned on its corresponding document words D, by modeling the conditional probability distribution $p(S|D)$.

  Specifically, for the encoder-decoder architecture, an encoder is employed to encode the source document D into a sequence of continuous vector representations, from which a decoder then generates the summary sequence autoregressively. Most abstractive ATS systems are sequential models and use a teacher-forcing training strategy. Summaries created in the abstractive manner are more flexible with the use of novel words, but they could suffer from low fluency, hallucination, and grammar errors.

- **Summary Evaluation:** How to automatically evaluate the quality of model output summaries has been one of the most critical problems in ATS system designs. Existing summary evaluation metrics mostly rely on the summary's similarity to the reference gold summary (human-written) as criteria. The most popular metric is the ROUGE F-score [**60**], which measures the n-gram similarity between generated summaries and corresponding reference summaries. Specifically, ROUGE-1/2 scores refer to the unigram and bigram overlap and thus indicate summary informativeness. ROUGE-L score refers to the longest common sequence and thus indicates summary fluency. Researchers have also explored model-based evaluation metrics that compute token similarity using contextual embeddings like BERTscore [**132**]. Recently, researchers also explored LLM-based metrics, such as G-Eval [**64**]. It employs an LLM with chain-of-thoughts (CoT) and a form-filling paradigm to evaluate the quality of natural language generation (NLG) outputs. G-Eval has shown the highest correlation with human judgments compared to other summarization quality metrics.

Moreover, the faithfulness of generated summaries is critical for their real-world applications and interests. Researchers have also explored metrics to automatically evaluate summary faithfulness. FactCC [**51**] is a weakly supervised BERT-based model metric that verifies factual consistency through rule-based transformations applied to source document sentences. It shows a high correlation in assessing summary faithfulness with human judgments. DAE [**33**] decomposes entailment at the level of dependency arcs, examining the semantic relationships within the generated output and input. Rather than focusing on aggregate decisions, DAE measures the semantic relationship manifested by individual dependency arcs in the generated output supported by the input. Questeval [**98**] is a question answering based metric to measure summary faithfulness by how many generated questions could be answered by the summary.

1.1.2. Pre-trained Language Models.

The progress of recent ATS systems also highly relies on the advent of recent large-scale PLMs. The self-attention mechanism within the Transformer structure [**105**] brings parallelization in computation and higher learning efficiency, enabling model training on large-scale unlabeled corpora. PLMs learn universal language representations from unsupervised training, which provides a better model initialization to downstream tasks instead of learning a new task from scratch.

Generally speaking, PLMs have three mainstream architectures: encoder-only, encoder-decoder, and decoder-only. Note that they all use the same self-attention layers as in the Transformers to encode word tokens. However, encoders are designed to learn embeddings from text data that can be used for various predictive modeling tasks such as classification. In contrast, decoders are designed to generate new texts autoregressively with the masking attention mechanism. Recent LLMs like the GPT family all use the decoder-only architecture since it is more parameter efficient and easier to scale up. Here, I will introduce some representative PLMs:

- **Encoder-only:** The encoder is responsible for understanding and extracting the relevant information from the input text. It can output continuous contextualized representations (embeddings) of the input text, which could be further manipulated for downstream tasks. Notable examples of encoder-only PLMs include BERT [**19**], RoBERTa [**67**], and sentenceBERT [**92**].
- **Encoder-decoder:** Encoder-decoder models are widely used for natural language processing tasks involving understanding input sequences and generating output sequences, such as text translation and summarization. They are effective at capturing the mapping between input and output sequences. Notable examples of encoder-decoder PLMs include T5 [**89**] and BART [**54**].
- **Decoder-only:** Decoder-only PLMs are autoregressive models that are widely used for generation tasks. The most notable models include the GPT family, which have shown remarkable performance in various benchmarks and are currently the most popular architecture for NLP in the era of LLMs. LLMs like GPT models learn emergent capabilities from large-scale pre-training and excel in a variety of NLP tasks with strong zero-shot and few-shot performance [**11**, **110**].

1.1.3. Graph for Summarization.

In early-stage NLP research, text data was often treated as a bag of tokens, such as bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF). With the development of Word Embedding techniques [**76**, **84**], recent work typically represents text data as a sequence of tokens and learns text representations with sequential deep learning frameworks such as RNNs and LSTMs.

On the other hand, textual data also contain rich structural information and could be represented by structures such as dependency parsing trees and constituency graphs. Graph models have been widely applied to extractive summarization due to their capability of modeling cross-sentence relations within a document. The sparse nature of graph structure also brings more scalability and flexibility.

The most common way of representing a document as a graph is to build sentence-level similarity graphs. Given a document $D = s_1^d, s_2^d, ..., s_n^d$, we can construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ stands for the node set and $\mathcal{E}$ represents edges between nodes. Each node $v_i \in \mathcal{V}$ in the sentence relation graph represents a corresponding sentence s_i^d in the document. The edge $e_{i,j} \in \mathcal{E}$ between node v_i and node v_j represents the semantic similarity between sentences s_i and s_j.

Without loss of generality, we use a pre-trained encoder to obtain initial node (sentence) representations as follows:

$$\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\} = \text{encoder}(\{s_1^d, s_2^d, ..., s_n^d\}). \tag{1.1}$$

We can then calculate the edge weight between two nodes (v_i, v_j) as the cosine similarity of their semantic representations $(\mathbf{h}_i, \mathbf{h}_j)$:

$$sim(\mathbf{h}_i, \mathbf{h}_j) = \frac{{\mathbf{h}_i}^\top \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\|}. \tag{1.2}$$

Rather than having a fully-connected graph, it's common to treat low similarity node pairs as disconnected. This approach emphasizes the connectivity of the graph and helps avoid noisy connection information. We can maintain a threshold $\theta \in [0, 1]$ for edge weights, such that edges with similarity scores smaller than θ are set to 0.

Unsupervised graph-based summarization methods rely on graph connectivity (node centrality) to score and rank sentences [**26**, **75**]. Recently, researchers have also explored supervised graph neural networks (GNNs) such as Graph Attention Network (GAT) [**106**] for the task of extractive summarization [**107**, **115**, **118**].

Specifically, GAT is an effective neural network architecture that operates on graph-structured data by leveraging masked self-attention layers. Given a constructed graph $\mathcal{G} =$

$(\mathcal{V}, \mathcal{E})$ with initial node representations $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|V|}\}$ and adjacency matrix $\mathbf{A}$, a GAT layer updates a node v_i with representation $\mathbf{h}_i$ to $\mathbf{h}'_i$ by:

$$
\begin{aligned}
e_{ij} &= \text{LeakyReLU}\left(\mathbf{W}_a \left[\mathbf{W}_{in}\mathbf{h}_i \| \mathbf{W}_{in}\mathbf{h}_j\right]\right), \\
\alpha_{ij} &= \frac{\exp\left(e_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(e_{ik}\right)}, \\
\mathbf{h}'_i &= \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}_v \mathbf{h}_j\right),
\end{aligned}
\tag{1.3}
$$

where $\mathcal{N}_i$ denotes the 1-hop neighbors of node v_i, α_{ij} denotes the attention weight between nodes $\mathbf{h}_i$ and $\mathbf{h}_j$, $\mathbf{W}_{in}, \mathbf{W}_a, \mathbf{W}_v$ are trainable weight matrices, and $\|$ denotes concatenation operation.

The single-head graph attention described above is further extended to multi-head attention, where T independent attention mechanisms are applied, and their outputs are concatenated as:

$$
\mathbf{h}'_i = \|_{t=1}^{T} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^t \mathbf{W}_h^k \mathbf{h}_j\right) \tag{1.4}
$$

In summary, this section covers the problem formulation of text summarization, the fundamentals of PLMs, and graph-based summarization. These topics are essential for better understanding this dissertation. I will discuss the main contributions of this dissertation and its outline next.

1.2. Contributions of this Dissertation

The overall research roadmap during my Ph.D. study is presented in Figure 1.3. As previously mentioned, this dissertation will mainly focus on various ATS approaches to address challenges in the three critical steps (document modeling, salient information extraction, and summary generation) of building intelligent and reliable summarization systems by answering the following research questions:

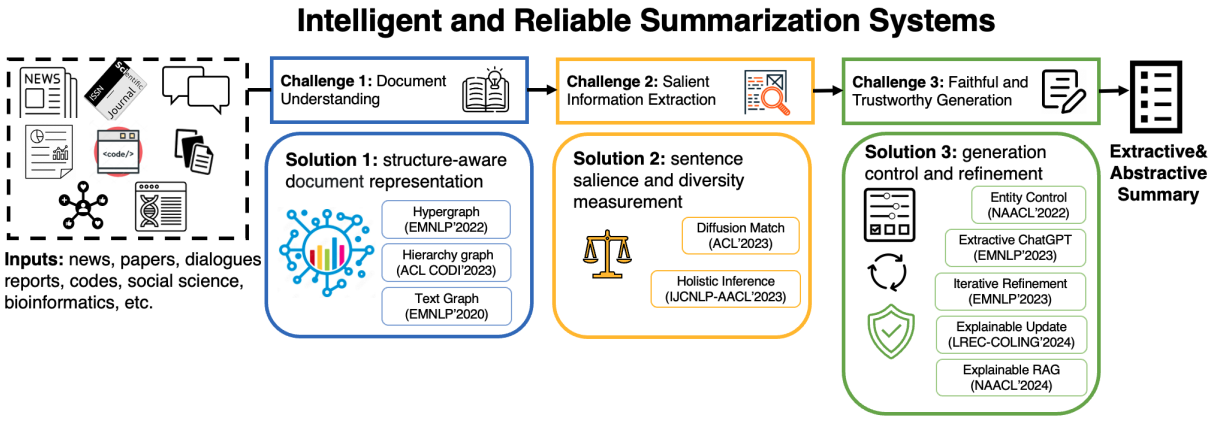

FIGURE 1.3. Overview of my Ph.D. research work to empower intelligent and reliable summarization along three research aspects: document understanding, salient information extraction, and faithful summary generation.

- How can we model the documents so that machines can understand their semantic content and inherent structures, especially for long documents?
- How can we enable machines to identify and extract the salient information from documents to form summaries?
- How can we ensure that the generated summaries are accurate and faithful?

This dissertation will address these questions that were previously under-investigated in text summarization research from different perspectives, covering both extractive and abstractive summarization. It will span from deep learning-based approaches to PLM-based approaches, and then to recent LLM-based ones. Specifically, the detailed contributions of this dissertation include:

- In Chapter 2, I proposed a hypergraph neural model for long document summarization. It is the first system to model high-order cross-sentence relations for extractive document summarization. This work was originally presented at the EMNLP 2022 conference [**123**].
- In Chapter 3, I proposed a generation-augmented framework to extract salient information from documents. The generation-augmented paradigm formulates extractive summarization as a task of vector matching and is the first attempt to apply continuous generative models for the extractive summarization task. This work was originally presented at the ACL 2023 conference [**125**].
- In Chapter 4, I proposed a holistic framework for multi-document extractive summarization. Our framework incorporates a holistic inference method for summary sentence extraction and a holistic measurement called the Subset Representative Index for balancing the importance and diversity of a subset of sentences. This work was originally presented at the IJCNLP-AACL 2023 conference [**121**].
- In Chapter 5, I proposed a novel framework for iterative text summarization with LLMs, enabling the iterative refinement of generated summaries by incorporating self-evaluation and feedback mechanisms. Additionally, I identified a potential issue of over-correction for LLM-based evaluation and editing. This work was originally presented at the EMNLP 2023 conference [**127**].
- In Chapter 6, I benchmarked the performance of ChatGPT for extractive summarization and investigated the effectiveness of in-context learning and chain-of-thought reasoning. I also demonstrated how to fuse extractive and abstractive summarization using an extract-then-generate pipeline with LLMs to further enhance the faithfulness of generated summaries. This work was also originally presented at the EMNLP 2023 conference [**126**].

In summary, by demonstrating and examining these systems and solutions, I aim to highlight the importance of structural document modeling for document comprehension,

holistic optimization for salient information extraction, and iterative refinement for faithful summary generation in the design of intelligent and reliable summarization systems.

1.3. Outline of this Dissertation

The rest of this dissertation is organized into six chapters. In the first few chapters, I present several approaches to address the three major steps (document modeling, salient information extraction, and summary generation) in building ATS systems. This includes long document modeling with structural learning (Chapter 2), salient information extraction with generation models (Chapter 3), summary salience and diversity balancing for multi-doc summarization (Chapter 4), and summary generation refinement with large language models (Chapter 5). I will then demonstrate how to fuse extractive and abstractive summarization with LLM in Chapter 6, before concluding in Chapter 7. More specifically, the topics covered in each chapter are briefly summarized as follows:

- In Chapter 2, I first present my work aimed at improving the modeling of long documents for extractive summarization. I introduce model HEGEL, a hypergraph neural network designed for long document summarization. HEGEL captures high-order cross-sentence relations by updating and learning effective sentence representations using hypergraph transformer layers. It also integrates various types of sentence dependencies, such as latent topics, keywords coreference, and section structure. Extensive experiments on two benchmark datasets demonstrate the effectiveness and efficiency of HEGEL in long document modeling and extractive summarization.
- In Chapter 3, I move on to the holistic extraction of salient information from documents. To overcome the limitations of individual sentence label prediction in existing extractive summarization systems, I proposed a novel paradigm named DiffuSum.

DiffuSum directly generates summary sentence representations using diffusion models and extracts sentences based on vector matching of these representations. Additionally, DiffuSum jointly optimizes a contrastive sentence encoder with a matching loss for aligning sentence representations and a multi-class contrastive loss for ensuring representation diversity. Experimental results demonstrate that DiffuSum achieves new state-of-the-art extractive summarization results on popular benchmarks. The strong performance of our framework highlights the significant potential of adapting generative models for extractive summarization.

- In Chapter 4, I propose a new holistic framework for unsupervised multi-document extractive summarization. The method incorporates the holistic beam search inference method associated with holistic measurements, named Subset Representative Index (SRI). SRI balances the importance and diversity of a subset of sentences from the source documents and can be calculated in unsupervised and adaptive manners. The proposed method outperforms strong baselines by a significant margin, as indicated by the resulting ROUGE scores and diversity measures. Our findings also suggest that diversity is essential for improving multi-document summary performance.
- In Chapter 5, I demonstrate my work on improving the quality and faithfulness of generated summaries. Despite significant progress in text summarization systems, they often generate summaries in one single step, which can lead to issues like hallucinations or overlooking essential details related to the reader's interests. To address this, I proposed SummIt, an iterative text summarization framework based on LLMs like ChatGPT. SummIt enables the model to refine the generated summary iteratively through self-evaluation and feedback, mirroring humans' iterative process when drafting and revising summaries. Additionally, I investigate the potential benefits of integrating knowledge and topic extractors into the framework to enhance summary faithfulness and controllability. Automatic evaluation and human studies

conducted on three benchmark summarization datasets validate the effectiveness of the iterative refinements and help identify potential issues of over-correction.

- In Chapter 6, I present a thorough evaluation of ChatGPT's performance on extractive summarization across various benchmark datasets. The experimental analysis reveals that ChatGPT demonstrates inferior extractive summarization performance in terms of ROUGE scores compared to existing supervised systems, while achieving higher performance based on LLM-based evaluation metrics. Additionally, I explore the effectiveness of in-context learning and chain-of-thought reasoning to enhance its performance. Moreover, I propose an extract-then-generate pipeline, which shows promising potential to yield significant improvements in terms of summary faithfulness. These observations highlight potential directions for enhancing ChatGPT's capabilities in faithful summarization through two-stage approaches.
- Finally, in Chapter 7, I summarize this dissertation and discuss future research directions in ATS systems design.

1.4. List of Publications

During my Ph.D. studies, I have made contributions to the following list of publications [**57**, **121**, **122**, **123**, **124**, **125**, **126**, **127**, **128**, **129**], some of which have been utilized to convey my research findings and are documented within this dissertation.

(1) Zizhong Li, **Haopeng Zhang**, and Jiawei Zhang. "Unveiling the Magic: Investigating Attention Distillation in Retrieval-augmented Generation." In Proceedings of the Association for Computational Linguistics: NAACL 2024. 2024.

(2) **Zhang, Haopeng**, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. "XATU: A Fine-grained Instruction-based Benchmark for Explainable Text Updates." In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). 2024.

(3) **Zhang, Haopeng**, Xiao Liu, and Jiawei Zhang. "SummIt: Iterative Text Summarization via ChatGPT." In Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 10644-10657. 2023.

(4) **Zhang, Haopeng**, Xiao Liu, and Jiawei Zhang. "Extractive Summarization via ChatGPT for Faithful Summary Generation." In Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 3270-3278. 2023.

(5) **Zhang, Haopeng**, Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Hongwei Wang, Jiawei Zhang, and Dong Yu. "Unsupervised Multi-document Summarization with Holistic Inference." In Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings), pp. 123-133. 2023.

(6) **Zhang, Haopeng**, Xiao Liu, and Jiawei Zhang. "DiffuSum: Generation Enhanced Extractive Summarization with Diffusion." In Findings of the Association for Computational Linguistics: ACL 2023, pp. 13089-13100. 2023.

(7) **Zhang, Haopeng**, Xiao Liu, and Jiawei Zhang. "Contrastive Hierarchical Discourse Graph for Scientific Document Summarization." In Proceedings of the 4th

Workshop on Computational Approaches to Discourse (CODI 2023), pp. 37-47. 2023.

(8) **Zhang, Haopeng**, Xiao Liu, and Jiawei Zhang. "HEGEL: Hypergraph Transformer for Long Document Summarization." In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 10167-10176. 2022.

(9) **Zhang, Haopeng**, Semih Yavuz, Wojciech Kryściński, Kazuma Hashimoto, and Yingbo Zhou. "Improving the Faithfulness of Abstractive Summarization via Entity Coverage Control." In Findings of the Association for Computational Linguistics: NAACL 2022, pp. 528-535. 2022.

(10) **Zhang, Haopeng**, and Jiawei Zhang. "Text Graph Transformer for Document Classification." In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8322-8327. 2020.

(11) Zhang, Jiawei, **Haopeng Zhang**, Congying Xia, and Li Sun. "Graph-bert: Only attention is needed for learning graph representations." arXiv preprint arXiv: 2001.05140 (2020).

CHAPTER 2

Leveraging Structures for Long Document Modeling

2.1. Introduction

Extractive summarization aims to condense a document while retaining its key information by directly selecting relevant sentences. Recent advancements in neural networks and large pre-trained language models [**19**, **54**] have led to promising results in news summarization, typically dealing with documents around 650 words long [**13**, **66**, **77**, **80**, **99**, **128**]. However, these models face challenges when applied to longer documents like scientific papers, which can range from 2000 to 7000 words in length, with expected summaries (abstracts) exceeding 200 words compared to the shorter summaries found in news headlines, typically around 40 words.

Extractive summarization of scientific papers poses significant challenges due to their long and structured nature. The extensive context makes it difficult for sequential models like RNNs to capture sentence-level long-distance dependencies and cross-sentence relations, which are crucial for extractive summarization. Additionally, the quadratic computation complexity of attention mechanisms in Transformer-based models [**105**] renders them impractical for long documents. Furthermore, long documents often cover diverse topics and contain richer structural information compared to short news articles, making it even more challenging for sequential models to capture.

As a result, researchers have turned to graph neural network (GNN) approaches to model cross-sentence relations. These methods typically represent a document as a sentence-level graph and frame extractive summarization as a node classification problem. Various approaches construct graphs from documents in different manners, including inter-sentence cosine similarity graphs [**22**, **26**], Rhetorical Structure Theory (RST) tree relation graphs [**115**],

approximate discourse graphs [**118**], topic-sentence graphs [**17**], and word-document heterogeneous graphs [**107**]. However, the usability of these approaches is often limited by two main aspects:

- These methods only model pairwise interactions between sentences, while interactions in natural language can be triadic, tetradic, or even of higher order [**21**]. Capturing high-order cross-sentence relations for extractive summarization remains an open question.
- These graph-based approaches rely on either semantic or discourse structure cross-sentence relations but are incapable of fusing sentence interactions from different perspectives.

Sentences within a document can interact in various ways, including embedding similarity, keyword coreference, topical modeling from a semantic perspective, and section or rhetorical structure from a discourse perspective. Capturing multi-type cross-sentence relations could enhance sentence representation learning and salience modeling. Figure 2.1 illustrates how different types of sentence interactions provide varying connectivity for document graph construction, encompassing both local and global context information.

To address the above issues, we propose **HEGEL** (**H**yp**E**r**G**raph transformer for **E**xtractive **L**ong document summarization), a graph-based model designed for summarizing long documents with rich discourse information. To better model high-order cross-sentence relations, we represent a document as a hypergraph, a generalization of graph structure where an edge can connect any number of vertices. We introduce three types of hyperedges to model sentence relations from different perspectives: section structure, latent topics, and keyword coreference. Additionally, we propose hypergraph transformer layers to update and learn effective sentence embeddings on hypergraph structures. We validate HEGEL through extensive experiments and analyses on two academic paper summarization benchmark datasets, demonstrating its effectiveness and efficiency. Our contributions are highlighted as follows:

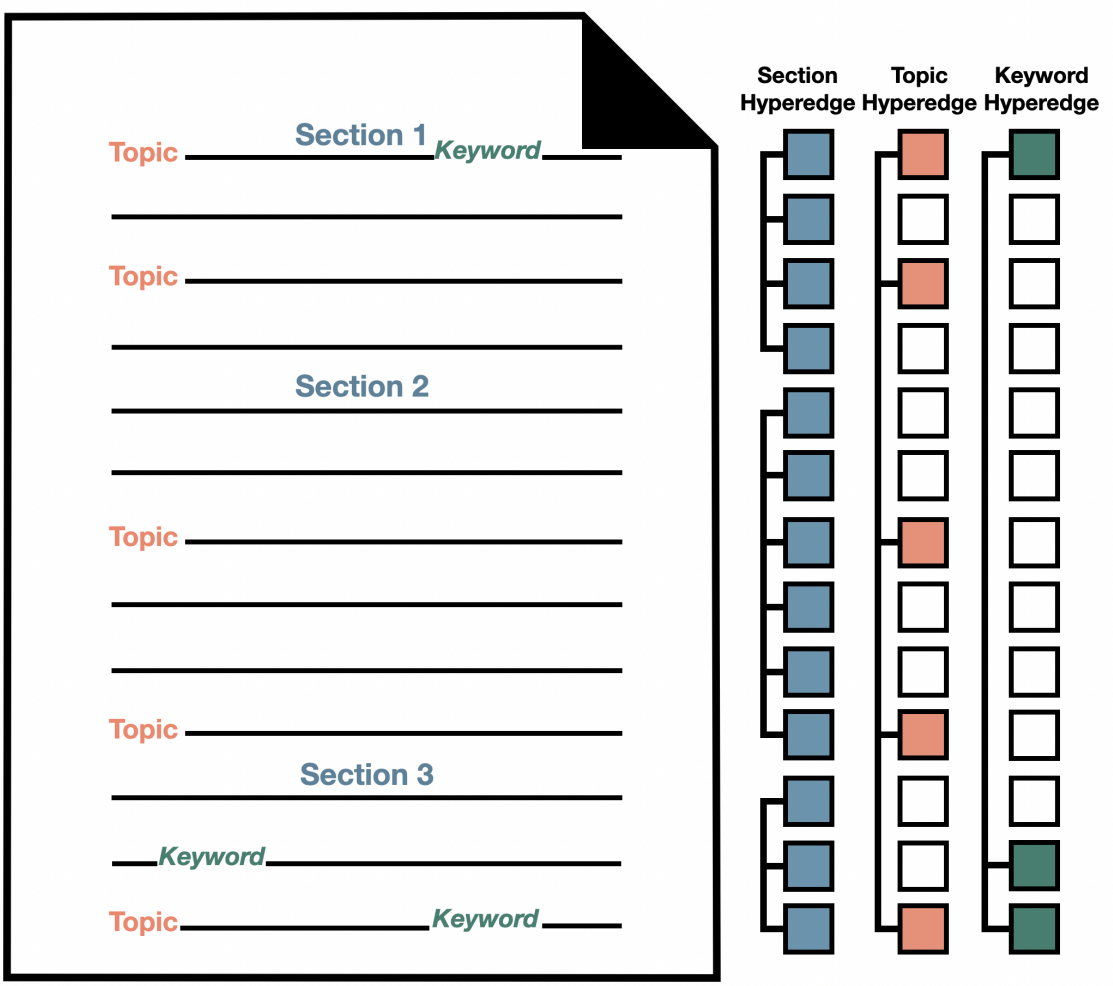

FIGURE 2.1. An illustration of modeling cross-sentence relations from section structure, latent topic, and keyword coreference perspectives.

- We propose a hypergraph neural model, named HEGEL, for long document summarization. To the best of our knowledge, we are the first to model high-order cross-sentence relations with hypergraphs for extractive document summarization.
- We propose three types of hyperedges (section, topic, and keyword) that capture sentence dependencies from different perspectives. Hypergraph transformer layers are then designed to update and learn effective sentence representations through message passing on the hypergraph.
- We validate HEGEL on two benchmark datasets (arXiv and PubMed), and the experimental results demonstrate its effectiveness over state-of-the-art baselines.

We also conduct ablation studies and qualitative analysis to further investigate the model's performance.

2.2. Related Work

2.2.1. Scientific Paper Summarization.

With promising progress in short news summarization, research interest in long-form documents such as academic papers has emerged. [**16**] proposed two benchmark datasets, ArXiv and PubMed, and employed a pointer generator network with a hierarchical encoder and discourse-aware decoder. [**113**] proposed an encoder-decoder model by incorporating global and local contexts. [**46**] introduced an unsupervised extractive approach to summarizing long scientific documents based on the Information Bottleneck principle. [**22**] devised an unsupervised ranking model by incorporating hierarchical graph representation and asymmetrical positional cues. Recently, [**95**] proposed applying a pre-trained language model with hierarchical structure information.

2.2.2. Graph based summarization.

Graph-based models have been utilized for extractive summarization to capture cross-sentence dependencies. Unsupervised graph summarization methods rely on graph connectivity to score and rank sentences [**22**, **87**, **135**]. Researchers also explore supervised graph neural networks for summarization. [**118**] applied Graph Convolutional Network (GCN) to the approximate discourse graph. [**115**] proposed applying GCN to structural discourse graphs based on RST trees and coreference mentions. [**17**] leveraged topical information by constructing topic-sentence graphs. Recently, [**107**] proposed constructing word-document heterogeneous graphs and using word nodes as intermediaries between sentences. [**45**] proposed using a multiplex graph to consider different sentence relations. Our work follows this line of work in developing novel graph neural networks for single-document extractive summarization. The main difference is that we construct a hypergraph from a document that

could capture high-order cross-sentence relations instead of pairwise relations, and fuse different types of sentence dependencies, including section structure, latent topics, and keyword coreference.

2.3. Method

In this section, we introduce HEGEL in great detail. We first present how to construct a hypergraph for a given long document. After encoding sentences into contextualized representations, we extract their section, latent topic, and keyword coreference relations and fuse them into a hypergraph. Then, our hypergraph transformer layer will update and learn sentence representations according to the hypergraph. Finally, HEGEL will score the salience of sentences based on the updated sentence representations to determine if the sentence should be included in the summary. The overall architecture of our model is shown in Figure 2.2.

2.3.1. Document as a Hypergraph.

A hypergraph is defined as a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ represents the set of nodes, and $\mathcal{E} = \{e_1, \ldots, e_m\}$ represents the set of hyperedges in the graph. Here each hyperedge e connects two or more nodes (i.e., $\sigma(e) \geq 2$). Specifically, we use the notations $v \in e$ and $v \notin e$ to denote whether node v is connected to hyperedge e or not in the graph G, respectively. The topological structure of hypergraph can also be represented by its incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$:

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{if } v_i \notin e_j \end{cases} \tag{2.1}$$

Given a document $D = \{s_1, s_2, ..., s_n\}$, each sentence s_i is represented by a corresponding node $v_i \in \mathcal{V}$. A Hyperedge e_j will be created if a subset of nodes $\mathcal{V}_j \subset \mathcal{V}$ share common semantic or structural information.

2.3.1.1. *Node Representation.*

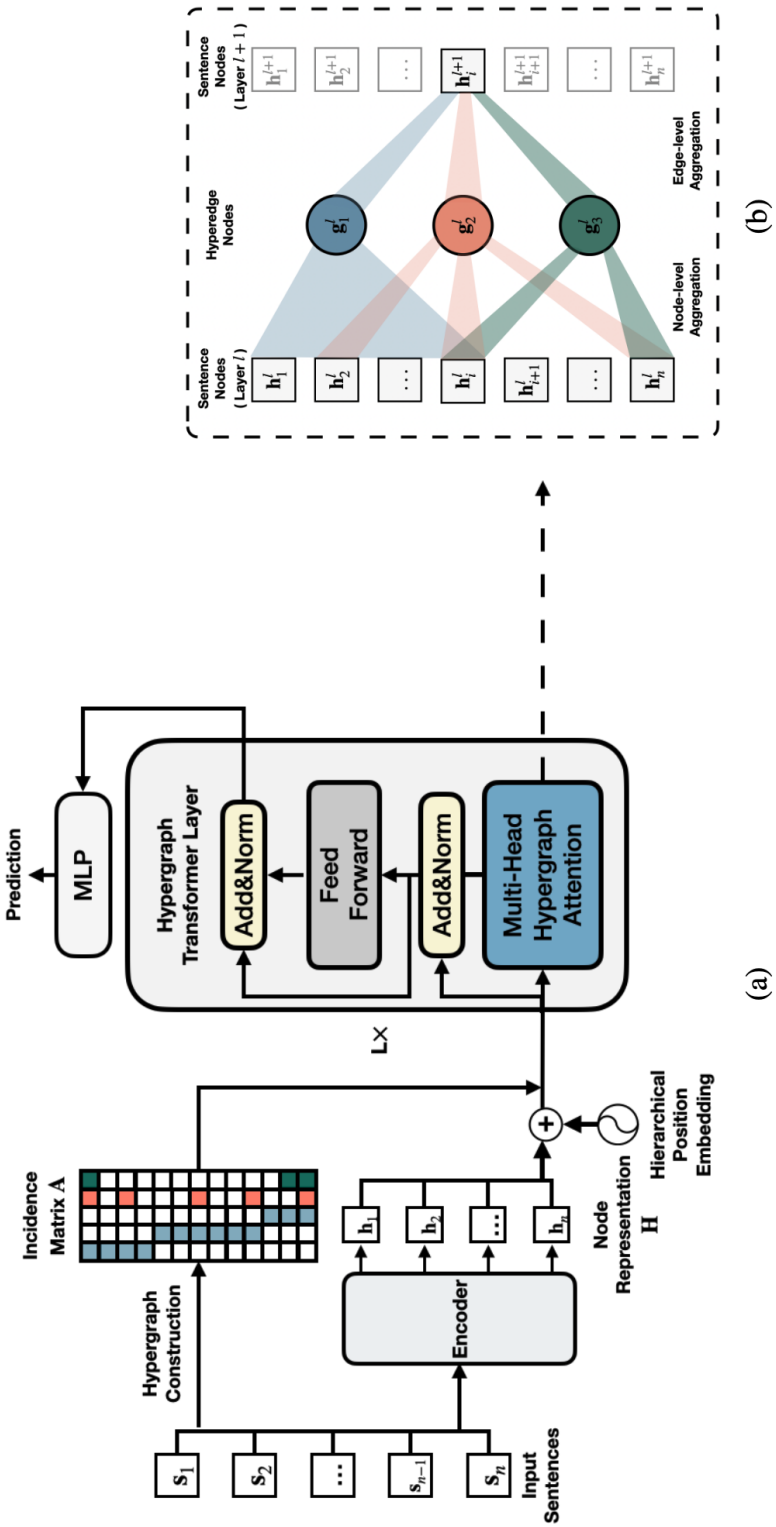

FIGURE 2.2. (a) The overall architecture of HEGEL. (b) Two-phase message passing mechanism in hypergraph transformer layer

We first adopt sentence-BERT [**92**] as the sentence encoder to embed the semantic meanings of sentences as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Note that the sentence-BERT is only used for initial sentence embedding, but not updated in HEGEL.

To preserve the sequential information, we also add positional encoding following Transformer [**105**]. We adopt the hierarchical position embedding [**95**], where the position of each sentence s_i can be represented as two parts: the section index of the sentence p_i^{sec}, and the sentence index in its corresponding section p_i^{sen}. The hierarchical position embedding (HPE) of sentence s_i can be calculated as:

$$\text{HPE}(s_i) = \gamma_1 \text{PE}(p_i^{sec}) + \gamma_2 \text{PE}(p_i^{sen}), \tag{2.2}$$

where γ_1, γ_2 are two hyperparameters to adjust the scale of positional encoding and $\text{PE}(\cdot)$ refers to the position encoding function:

$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d_{model}}), \tag{2.3}$$

$$\text{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}). \tag{2.4}$$

Then we can get the initial input node representations $\mathbf{H}^0 = \{\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0\}$, with vector $\mathbf{h}_i^0$ defined as:

$$\mathbf{h}_i^0 = \mathbf{x}_i + \text{HPE}(s_i) \tag{2.5}$$

2.3.1.2. *Hyperedge Construction.*

To effectively model multi-type cross-sentence relations in a long context, we propose the following three types of hyperedges. These hyperedges can capture high-order context information via multi-node connections and model both local and global context through document structures from different perspectives.

- **Section Hyperedges:** Scientific papers mostly follow a standard discourse structure describing the problem, methodology, experiments/results, and finally conclusions, so sentences within the same section tend to have the same semantic focus [**102**]. To capture the *local* sequential context, we build section hyperedges that consider each section as a hyperedge connecting all the sentences in that section. Section hyperedges could also address the incidence matrix sparsity issue and ensure all nodes of the graph are connected by at least one hyperedge. Assume a document has q sections, section hyperedge e_j^{sec} for the j-th section can be represented formally in its corresponding incidence matrix $\mathbf{A}_{sec} \in \mathbb{R}^{n \times q}$ as:

$$A_{ij}^{sec} = \begin{cases} 1, \text{ if } s_i \in e_j^{sec} \\ 0, \text{ if } s_i \notin e_j^{sec} \end{cases} \tag{2.6}$$

  where A_{ij}^{sec} denotes whether the i-th sentence is in the j-th section.

- **Topic Hyperedges:** Topical information has been demonstrated to be effective in capturing important content [**17**]. To leverage topical information of the document, we first apply the Latent Dirichlet Allocation (LDA) model [**9**] to extract the latent topic relationships between sentences and then construct the topic hyperedge. In addition, topic hyperedges could address the long-distance dependency problem by capturing *global* topical information of the document. After extracting p topics from LDA, we construct p corresponding topic hyperedges e_j^{topic}, represented by the entry A_{ij}^{topic} in the incidence matrix $\mathbf{A}_{topic} \in \mathbb{R}^{n \times p}$ as:

$$A_{ij}^{topic} = \begin{cases} 1, \text{ if } s_i \in e_j^{topic} \\ 0, \text{ if } s_i \notin e_j^{topic} \end{cases} \tag{2.7}$$

  where A_{ij}^{topic} denotes whether the i-th sentence belongs to the j-th latent topic.

- **Keyword Hyperedges:** Previous work finds that keywords compose the main body of the sentence, which are regarded as the indicators for important sentence

selection [**55**, **109**]. Keywords in the original sentence provide significant clues for the main points of the sentence. To utilize keyword information, we first extract keywords for academic papers with KeyBERT [**35**] and construct keyword hyperedges to link the sentences that contain the same keyword regardless of their sequential distance. Similar to topic hyperedges, keyword hyperedges also capture *global* context relations and thus, address the long-distance dependency problem. After extracting k keywords for a document, we construct k corresponding keyword hyperedges e_j^{kw}, represented in the incidence matrix $\mathbf{A}_{kw} \in \mathbb{R}^{n \times k}$ as:

$$A_{ij}^{kw} = \begin{cases} 1, \text{ if } s_i \in e_j^{kw} \\ 0, \text{ if } s_i \notin e_j^{kw}, \end{cases} \tag{2.8}$$

where $s_i \in e_j^{kw}$ means the i-th sentence contains the j-th keyword.

We finally fuse the three hyperedges by concatenation $\|$ and get the overall incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ as:

$$\mathbf{A} = \mathbf{A}_{sec} \| \mathbf{A}_{topic} \| \mathbf{A}_{kw}, \tag{2.9}$$

where its dimension $m = q + p + k$.

The initial input node representations $\mathbf{H}^0 = \{\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0\}$ and the overall hyperedge incidence matrix $\mathbf{A}$ will be fed into hypergraph transformer layers to learn effective sentence embeddings.

2.3.2. Hypergraph Transformer Layer.

The self-attention mechanism proposed in the Transformer [**105**] has demonstrated its effectiveness for learning text representation and graph representations [**21**, **106**, **119**, **129**, **130**]. To model cross-sentence relations and learn effective sentence (node) representations in hypergraphs, we propose the Hypergraph Transformer Layer, as in Figure 2.2.

2.3.2.1. *Hypergraph Attention.*

Given node representations $\mathbf{H}^0 = \{\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0\}$ and hyperedge incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, a l-layer hypergraph transformer computes hypergraph attention (HGA) and updates node representations $\mathbf{H}$ in an iterative manner as shown in Algorithm 1.

Specifically, in each iteration, we first obtain all m hyperedge representations $\{\mathbf{g}_1^l, \mathbf{g}_2^l, ..., \mathbf{g}_m^l\}$ as:

$$\mathbf{g}_j^l = \text{LeakyReLU}\left(\sum_{v_k \in e_j} \alpha_{jk} \mathbf{W}_h \mathbf{h}_k^{l-1}\right), \tag{2.10}$$

$$\begin{aligned} \alpha_{jk} &= \frac{\exp\left(\mathbf{w}_{ah}^{\mathrm{T}} \mathbf{u}_k\right)}{\sum_{v_p \in e_j} \exp\left(\mathbf{w}_{ah}^{\mathrm{T}} \mathbf{u}_p\right)}, \\ \mathbf{u}_k &= \text{LeakyReLU}\left(\mathbf{W}_h \mathbf{h}_k^{l-1}\right), \end{aligned} \tag{2.11}$$

where the superscript l denotes the model layer, matrices $\mathbf{W}_h, \mathbf{w}_{ah}$ are trainable weights and α_{jk} is the attention weight of node v_k in hyperedge e_j.

The second step is to update node representations $\mathbf{H}^{l-1}$ based on the updated hyperedge representations $\{\mathbf{g}_1^l, \mathbf{g}_2^l, ..., \mathbf{g}_m^l\}$ by:

$$\mathbf{h}_i^l = \text{LeakyReLU}\left(\sum_{v_i \in e_k} \beta_{ij} \mathbf{W}_e \mathbf{g}_k^l\right), \tag{2.12}$$

$$\begin{aligned} \beta_{ki} &= \frac{\exp\left(\mathbf{w}_{ae}^{\mathrm{T}} \mathbf{z}_k\right)}{\sum_{v_i \in e_q} \exp\left(\mathbf{w}_{ae}^{\mathrm{T}} \mathbf{z}_i\right)}, \\ \mathbf{z}_k &= \text{LeakyReLU}\left(\left[\mathbf{W}_e \mathbf{g}_k^l \| \mathbf{W}_h \mathbf{h}_i^{l-1}\right]\right), \end{aligned} \tag{2.13}$$

where $\mathbf{h}_i^l$ is the representation of node v_i, $\mathbf{W}_e, \mathbf{w}_{ae}$ are trainable weights, and β_{ki} is the attention weight of hyperedge e_k that connects node v_i. $\|$ here is the concatenation operation. In this way, information of different granularities and types can be fully exploited through the hypergraph attention message passing processes.

Multi-Head Hypergraph Attention: As in the vanilla Transformer, we also extend hypergraph attention (HGA) into multi-head hypergraph attention (MH-HGA) to expand the model's representation subspaces, represented as:

$$\begin{aligned} \text{MH-HGA}(\mathbf{H}, \mathbf{A}) &= \sigma(\mathbf{W}_O \|_{i=1}^{h} \text{head}_i), \\ \text{head}_i &= \text{HGA}_i(\mathbf{H}, \mathbf{A}), \end{aligned} \tag{2.14}$$

where $\text{HGA}(\cdot)$ denotes hypergraph attention, σ is the activation function, $\mathbf{W}_O$ is the multi-head weight, and $\|$ denotes concatenation.

Algorithm 1: $\text{MH-HGA}_{head}(\mathbf{H}, \mathbf{A})$

input : node representation $\mathbf{H}^{l-1} \in \mathbb{R}^{n \times d}$,
incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$
output: updated representation $\mathbf{H}^{l} \in \mathbb{R}^{n \times d}$

for *head* $= 1, 2, ..., h$ **do**
  `// update hyperedges from nodes`
  for $j = 1, 2, ..., m$ **do**
    for *node* $v_k \in e_j$ **do**
      compute attention α_{jk} with Eq. 2.11;
      update hyperedge representation $\mathbf{g}_j^l$ with Eq. 2.10;
    end
  end
  `// update node representations`
  for $i = 1, 2, ..., n$ **do**
    for *hyperedge that* $v_i \in e_k$ **do**
      compute attention β_{ki} with Eq. 2.13; update node representation $\mathbf{h}_i^l$ with Eq. 2.12;
    end
  end
end

2.3.2.2. *Hypergraph Transformer.*

After obtaining the multi-head attention, we also introduce the feed-forward blocks (FFN) with residual connection and layer normalization (LN) like in Transformer. We formally characterize the Hypergraph Transformer layer as below:

$$\begin{aligned} \mathbf{H}'^{(l)} &= \text{LN}(\text{MH-HGA}(\mathbf{H}^{l-1}, \mathbf{A}) + \mathbf{H}^{l-1}) \\ \mathbf{H}^{l} &= \text{LN}(\text{FFN}(\mathbf{H}'^{(l)}) + \mathbf{H}'^{(l)} \end{aligned} \tag{2.15}$$

2.3.3. Training Objective.

After passing L hypergraph transformer layers, we obtain the final sentence node representations $\mathbf{H}^L = \{\mathbf{h}_1^L, \mathbf{h}_2^L, ..., \mathbf{h}_n^L\}$. We then add a multi-layer perceptron(MLP) followed by a sigmoid activation function indicating the confidence score for selecting each sentence. Formally, the predicted confidence score $\hat{y}_i$ for sentence s_i is:

$$\begin{aligned} \mathbf{z}_i &= \text{LeakyReLU}(\mathbf{W}_{p1}\mathbf{h}_i^L), \\ \hat{y}_i &= \text{sigmoid}(\mathbf{W}_{p2}\mathbf{z}_i), \end{aligned} \tag{2.16}$$

where $\mathbf{W}_{p1}, \mathbf{W}_{p2}$ are trainable parameters.

Compared with the sentence ground truth label y_i, we train HEGEL in an end-to-end manner and optimize with binary cross-entropy loss as:

$$\mathcal{L} = -\frac{1}{N \cdot N_d} \sum_{d=1}^{N} \sum_{i=1}^{N_d} (y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)), \tag{2.17}$$

where N denotes the number of training instances in the training set, and N_d denotes the number of sentences in the document.

2.4. Experiment

This section presents experimental details on two benchmarked academic paper summarization datasets. We compare our proposed model with state-of-the-art baselines and conduct detailed analysis to validate the effectiveness of HEGEL.

	Arxiv	PubMed
# train	201,427	112,291
# validation	6,431	6,402
# test	6,436	6,449
avg. document length	4,938	3,016
avg. summary length	203	220

TABLE 2.1. Detailed statistics of the PubMed and Arxiv datasets including the train/validation/test split, the average length of documents and summary (in words).

2.4.1. Experiment Setup.

Datasets: Scientific papers are an example of long documents with a section discourse structure. Here we validate HEGEL on two benchmark scientific paper summarization datasets: ArXiv and PubMed [**16**]. PubMed contains academic papers from the biomedical domain, while ArXiv contains papers from different scientific domains. We use the original train, validation, and testing splits as in [**16**]. The detailed statistics of the datasets are shown in Table 2.1.

Compared Baselines: We conduct a systematic comparison with state-of-the-art baseline approaches as follows:

- Unsupervised methods: LEAD that selects the first few sentences as a summary; graph-based methods LexRank [**26**], PACSUM [**135**], and HIPORANK [**22**].
- Neural extractive models: encoder-decoder based model Cheng&Lapata [**13**] and SummaRuNNer [**77**]; local and global context model ExtSum-LG [**113**] and its variant RdLoss/MMR [**114**]; transformer-based models SentCLF, SentPTR [**86**], and HiStruct+ [**95**].

- Neural abstractive models: pointer network PGN [**99**], hierarchical attention model DiscourseAware [**16**], transformer-based model TLM-I+E [**86**], and divide-and-conquer method DANGER [**31**].

2.4.2. Implementation Details.

We use pre-trained sentence-BERT [**92**] checkpoint *all-mpnet-base-v2* as the encoder for initial sentence representations. The embedding dimension is 768, and the input layer dimension is 1024. In our experiment, we stack two layers of hypergraph transformer, and each has 8 attention heads with a hidden dimension of 128. The output layer's hidden dimension is set to 4096. We generate at most 100 topics for each document and filter out the topic and keyword hyperedges that connect less than 5 sentence nodes or greater than 25 sentence nodes. For position encodings, we set the rescale weights γ_1 and γ_2 to 0.001.

The model is optimized with Adam optimizer [**70**] with a learning rate of 0.0001 and a dropout rate of 0.3. We train the model on an RTX A6000 GPU for 20 epochs and validate after each epoch using ROUGE-1 F-score to choose checkpoints. Early stopping is employed to select the best model with the patience of 3.

Following the standard-setting, we use ROUGE F-scores [**60**] for performance evaluation. Specifically, ROUGE-1/2 scores measure summary informativeness, and the ROUGE-L score measures summary fluency. Following prior work [**78**], we construct extractive ground truth (ORACLE) by greedily optimizing the ROUGE score on the gold-standard abstracts for extractive summary labeling.

2.4.3. Experiment Results.

The performance of HEGEL and baseline methods on the ArXiv and PubMed datasets is shown in Table 2.2. The first block lists the extractive ground truth ORACLE and the unsupervised methods. The second block includes recent extractive summarization models, and the third contains state-of-the-art abstractive methods.

The LEAD method exhibits limited performance on scientific paper summarization compared to its strong performance on short news summarization datasets like CNN/Daily

Models	PubMed			ArXiv		
	R-1	R-2	R-L	R-1	R-2	R-L
ORACLE	55.05	27.48	49.11	53.88	23.05	46.54
LEAD	35.63	12.28	25.17	33.66	8.94	22.19
LexRank (2004)	39.19	13.89	34.59	33.85	10.73	28.99
PACSUM (2019)	39.79	14.00	36.09	38.57	10.93	34.33
HIPORANK (2021)	43.58	17.00	39.31	39.34	12.56	34.89
Cheng&Lapata (2016)	43.89	18.53	30.17	42.24	15.97	27.88
SummaRuNNer (2016)	43.89	18.78	30.36	42.81	16.52	28.23
ExtSum-LG (2019)	44.85	19.70	31.43	43.62	17.36	29.14
SentCLF (2020)	45.01	19.91	41.16	34.01	8.71	30.41
SentPTR (2020)	43.30	17.92	39.47	42.32	15.63	38.06
ExtSum-LG+RdLoss (2021)	45.30	20.42	40.95	44.01	17.79	39.09
ExtSum-LG+MMR (2021)	45.39	20.37	40.99	43.87	17.50	38.97
HiStruct+ (2022)	46.59	20.39	42.11	45.22	17.67	**40.16**
PGN (2017)	35.86	10.22	29.69	32.06	9.04	25.16
DiscourseAware (2018)	38.93	15.37	35.21	35.80	11.05	31.80
TLM-I+E (2020)	42.13	16.27	39.21	41.62	14.69	38.03
DANCER-LSTM (2020)	44.09	17.69	40.27	41.87	15.92	37.61
DANCER-RUM (2020)	43.98	17.65	40.25	42.70	16.54	38.44
HEGEL (ours)	**47.13**	**21.00**	**42.18**	**46.41**	**18.17**	39.89

TABLE 2.2. Experimental Results on PubMed and Arxiv datasets.

Mail [**37**] and New York Times [**96**]. This phenomenon indicates that academic papers have less positional bias than news articles, and the ground truth sentences are distributed more evenly.

For graph-based unsupervised baselines, HIPORANK [**22**] achieves state-of-the-art performance, which can even compete with some supervised methods. This demonstrates the significance of incorporating discourse structural information when modeling cross-sentence relations for long documents.

In general, neural extractive methods perform better than abstractive methods due to the extended context. Among extractive baselines, transformer-based methods like SentPTR and

HiStruct+ show substantial performance gains, demonstrating the effectiveness of the attention mechanism. HiStruct+ achieves strong performance by injecting inherent hierarchical structures into large pre-trained language models like Longformer. In contrast, our model HEGEL relies solely on hypergraph transformer layers for sentence representation learning and requires no pre-trained knowledge.

As shown in Table 2.2, HEGEL outperforms state-of-the-art extractive and abstractive baselines on both datasets. The superior performance of HEGEL highlights the capability of hypergraphs in modeling high-order cross-sentence relations and the importance of fusing both semantic and structural information. We conduct an extensive ablation study and performance analysis next.

2.5. Analysis

2.5.1. Ablation Study.

Model	ROUGE-1	ROUGE-2	ROUGE-L
full HEGEL	**47.13**	**21.00**	**42.18**
w/o Position	46.86	20.05	41.91
w/o Keyword	46.92	20.71	42.03
w/o Topic	46.35	20.30	41.48
w/o Section	45.63	19.30	40.71

TABLE 2.3. Ablation study results on the PubMed dataset.

We first analyze the influence of different components of HEGEL. Table 2.3 shows the experimental results of removing each type of hyperedge and the hierarchical position encoding of HEGEL on the PubMed dataset. As shown in the second row, removing the hierarchical position embedding hurts the model performance, indicating the importance of injecting sequential order information. Regarding hyperedges (rows 3-5), we can see that all three types of hyperedges (section, keyword, and topic) help boost the overall model performance. Specifically, the performance drops most when the section hyperedges are removed. The

hypergraph becomes sparse, harming its connectivity. This indicates that the section hyperedges, which contain local context information, play an essential role in the information aggregation process. Note that although we only discuss three types of hyperedges (section, keyword, and topic) in this work, it is easy to extend our model with hyperedges from other perspectives like syntactic for future work.

2.5.2. Hyperedge Analysis.

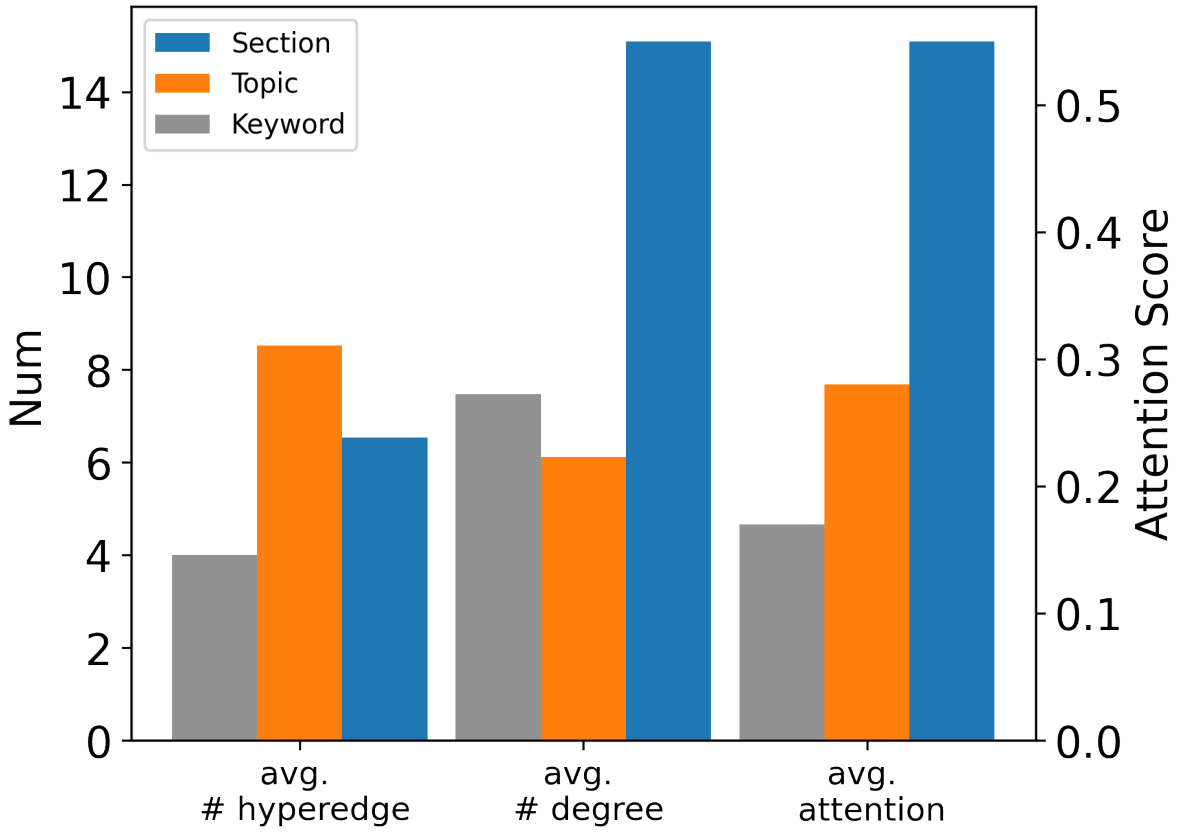

FIGURE 2.3. Average attention distribution over three types of hyperedges on PubMed dataset.

We also explore the hyperedge pattern to further understand the performance of HEGEL. As shown in Figure 2.3, we observe that topic hyperedges are the most frequent on average, while section hyperedges have the largest degree (number of connected nodes). In terms of cross-attention over the predicted sentence nodes, HEGEL allocates more than half of the attention to section hyperedges and the least attention to keyword edges. These results are

consistent with the earlier ablation study, indicating that local section context information plays a more critical role in long document summarization.

2.5.3. Embedding Analysis.

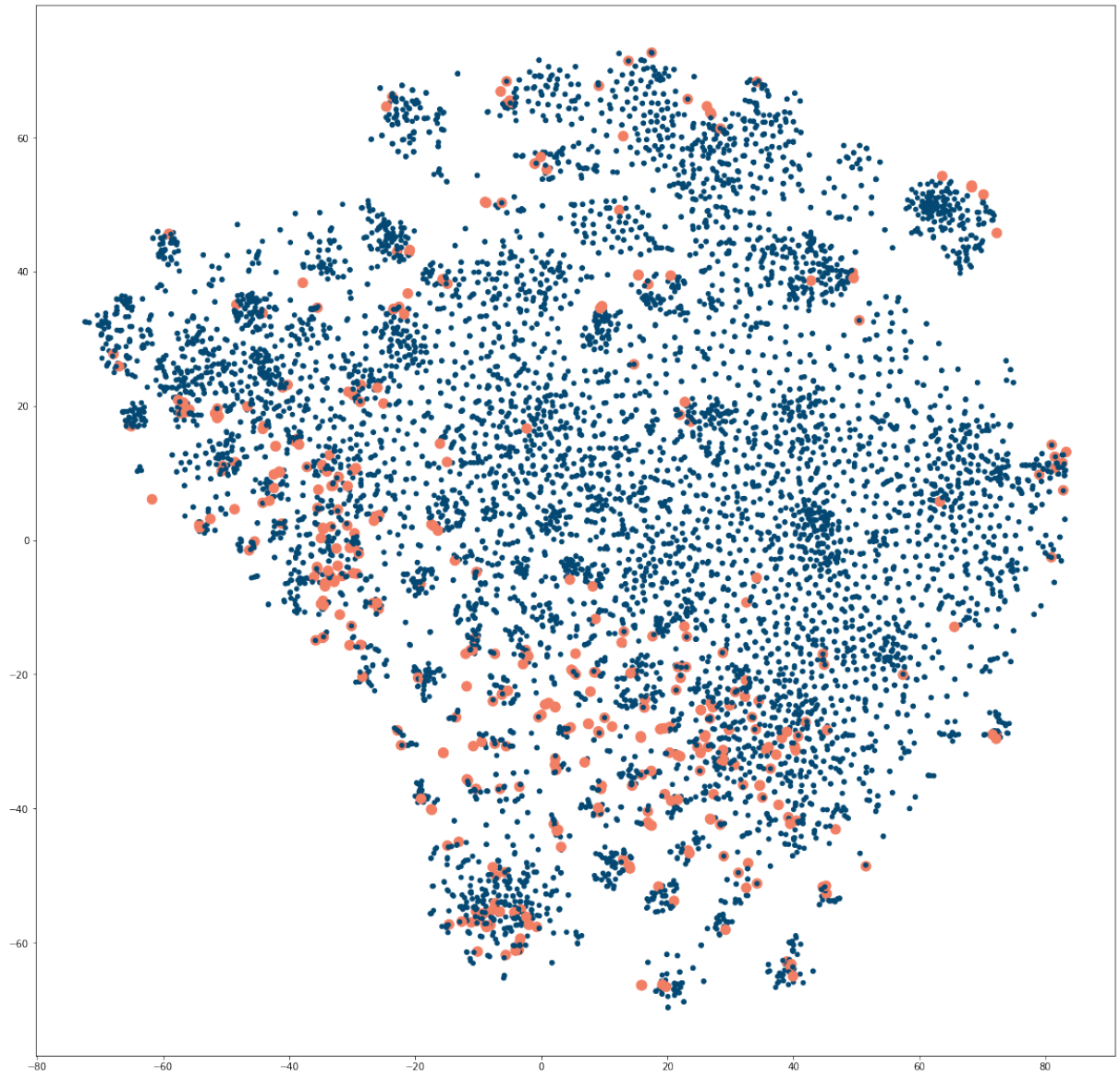

FIGURE 2.4. Visualization of sentence nodes embeddings for 100 documents from PubMed test set.

To explore the sentence embeddings learned by HEGEL, we present a visualization of the output sentence node embeddings from the last hypergraph transformer layer. We employ T-SNE [**103**] to reduce each node's dimension to 2, as shown in Figure 2.4. The orange dots

represent the ground truth sentences, while the blue dots represent the non-ground truth sentences. We can observe some clustering effects of the ground truth nodes, which tend to appear in the bottom-left zone of the plot. These results indicate that HEGEL learns effective sentence embeddings as indicators for salient sentence selection.

2.5.4. Case Study.

Here, we also provide an example output summary from HEGEL in Table 2.4. We can observe that the selected sentences span a long distance in the original document but are thematically related according to the latent topic and keyword coreference. As a result, HEGEL effectively captures high-order cross-sentence relations through multi-type hyperedges and selects these salient sentences based on learned high-order representations.

[Method] Phylogenetic analyses of partial middle east respiratory syndrome coronavirus genomic sequences for **viruses** detected in dromedaries imported from oman to united arab emirates, may 2015. (**Section 1**)
[Information] Additional information regarding 2 persons with asymptomatic merscov **infection** and other persons tested in the study. (**Section 2**)
[Information] Our findings provide further evidence that asymptomatic human **infections** can be caused by zoonotic transmission. (**Section 2**)
[Method] Merscov genomic sequences determined in this study are similar to those of **viruses** detected in 2015 in patients in saudi arabia and south korea with hospital - acquired **infections**. (**Section 3**)
[Information] The **infected** dromedaries were imported from oman , which suggests that **viruses** from this clade are circulating on the arabian peninsula. (**Section 4**)

TABLE 2.4. An example output summary of HEGEL. Topics are marked in orange, keywords are marked in green, and sections are marked in blue.

2.6. Conclusion

Overall, this chapter presents HEGEL for long document summarization. HEGEL represents a document as a hypergraph to address the long dependency issue and captures higher-order cross-sentence relations through multi-type hyperedges. The strong performance of HEGEL demonstrates the importance of modeling high-order sentence interactions and fusing semantic and structural information for future research in long document extractive summarization.

CHAPTER 3

Augmenting Salient Information Extraction with Generation Model

3.1. Introduction

Document summarization aims to compress text material while retaining its most salient information, playing a critical role in managing the growing amount of publicly available text data. Automatic text summarization approaches can be divided into two streams: abstractive and extractive summarization. Abstractive methods [**6**, **55**, **78**] aim to produce flexible and less redundant summaries by generating new phrases and sentences not explicitly present in the source text. However, they often face challenges in generating ungrammatical or even nonfactual content [**51**]. In contrast, extractive summarization forms a summary by directly extracting sentences from the source document. Thus, the extracted summaries are grammatically accurate and faithful representations of the original text.

We focus on extractive summarization in this work. Extractive summarization is commonly formulated as a sequence labeling problem, where the task is to predict a binary label (0 or 1) for each sentence, indicating whether the sentence should be included in the summary [**66**, **77**, **138**]. While sequence labeling approaches excel at predicting individual sentence labels, generative models offer increased flexibility and have shown success in attending to the entirety of the input context. Recent works have successfully applied generative models to various token-level sequence labeling tasks [**4**, **24**, **116**]. However, the application of generative models to sentence-level tasks like extractive summarization has not been explored extensively.

Recently, continuous diffusion models have achieved great success in the vision and audio domains [**38**, **39**, **50**, **94**]. Researchers have also attempted to apply diffusion models for

text generation by converting discrete tokens to continuous embeddings and mapping from the embedding space to words using a rounding method [**32**, **56**, **101**, **120**]. However, these approaches are not directly applicable for sentence-level tasks like summarization for several reasons:

(1) Summarization typically involves longer input contexts and larger generation lengths (around 3-6 sentences), while the existing token-level diffusion-LM models are primarily designed for short generation tasks like text simplification and question generation. Their performance tends to degrade significantly when generating longer sequences.

(2) The word embeddings generated by these models might be indistinguishable, leading to ambiguous and hallucinated generations.

(3) The rounding step in existing diffusion models can be less efficient and dramatically slow down the inference process.

To address the aforementioned issues, we propose a novel extractive summarization paradigm called **DiffuSum**. DiffuSum leverages transformer-based diffusion models to generate the desired summary sentence representations and extracts summaries based on sentence representation matching. Instead of generating text word by word, DiffuSum directly generates continuous representations for each summary sentence, allowing it to process much longer text. DiffuSum operates at the summary level, as the transformer-based diffusion architecture generates all summary sentence representations simultaneously.

Moreover, DiffuSum incorporates a contrastive sentence encoding module with a matching loss for sentence representation alignment and a multi-class contrastive loss [**48**] for representation diversity. DiffuSum jointly optimizes the **sentence encoding module** and the **diffusion generation module**, and extracts sentences by representation matching without any rounding step. We validate DiffuSum through extensive experiments on three benchmark datasets. The experimental results demonstrate that DiffuSum achieves comparable or even better performance than state-of-the-art systems relying on pre-trained language

models. Additionally, DiffuSum exhibits strong adaptation ability based on cross-dataset evaluation results.

We highlight our contributions of DiffuSum as follows:

- We propose a novel generation-augmented paradigm for extractive summarization with diffusion models. DiffuSum directly generates the desired summary sentence representations and then extracts sentences based on representation matching. To the best of our knowledge, this is the first attempt to apply diffusion models for the extractive summarization task.
- We introduce a contrastive sentence encoding module with a matching loss for representation alignment and a multi-class contrastive loss for representation diversity.
- We conduct extensive experiments and analysis on three benchmark summarization datasets to validate the effectiveness of DiffuSum. DiffuSum achieves new extractive state-of-art results on CNN/DailyMail dataset with ROUGE scores of 44.83/22.56/40.56.

3.2. Related Work

3.2.1. Extractive Summarization.

Recent advances in deep neural networks have dramatically boosted progress in extractive summarization systems. Existing extractive summarization systems encompass a wide range of approaches. Most works formulate the task as a sequence classification problem and utilize sequential neural models with various encoders such as recurrent neural networks [**13**, **78**] and pre-trained language models [**25**, **66**, **126**].

Another group of work formulates extractive summarization as a node classification problem and applies graph neural networks to model inter-sentence dependencies [**107**, **115**, **123**]. These formulations are sentence-level methods that make individual predictions for each sentence.

Recently, [**136**] observed that a summary consisting of sentences with the highest scores is not necessarily the best. As a result, summary-level formulations like text matching [**2**, **136**] and reinforcement learning [**6**, **80**] have been proposed.

Our proposed framework, DiffuSum, is also a novel summary-level extractive system with generation augmentation. Instead of sequentially labeling sentences, DiffuSum directly generates the desired summary sentence representations with diffusion models and extracts sentences by representation matching.

3.2.2. Diffusion Models for Text.

Continuous diffusion models were first introduced in [**100**] and have achieved significant success in continuous domain generations like image, video, and audio [**39**, **50**, **94**]. However, few works have applied continuous diffusion models to text data due to their inherently discrete nature.

Among the initial attempts, Diffusion-LM [**56**] first adapts continuous diffusion models for text by adding an embedding step and a rounding step, and designing a training objective to learn the embedding. DiffuSeq [**32**] proposes a diffusion model designed for sequence-to-sequence (seq2seq) text generation tasks by adding partial noise during the forward process and conditional denoising during the reverse process. CDCD [**20**] is proposed for text modeling and machine translation based on variance-exploding stochastic differential equations (SDEs) on token embeddings. SeqDiffuSeq [**120**] also proposes an encoder-decoder diffusion model architecture for conditional generation by combining self-conditioning and adaptive noise schedule techniques. However, these works mainly focus on generating token-level embeddings for short text generation (less than 128 tokens).

To adapt diffusion models to longer sequences like summaries, our proposed approach, DiffuSum, directly generates summary sentence embeddings with a partial denoising framework. Additionally, DiffuSum jointly optimizes the diffusion model with a contrastive sentence encoding module instead of using a static embedding matrix.

3.3. Preliminary

3.3.1. Continuous Diffusion Models.

The continuous diffusion model [**38**] is a probabilistic model containing two Markov chains: the forward and the backward process.

- **Forward Process:** Given a data point sampled from a real-world data distribution $\mathbf{x}_0 \sim q(x)$, the forward process gradually corrupts $\mathbf{x}_0$ into a standard Gaussian distribution prior $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Each step of the forward process gradually interpolates Gaussian noise to the sample, represented as:

$$q(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t+1}; \sqrt{1-\beta_t}\mathbf{x}_t, \beta_t\mathbf{I}\right), \tag{3.1}$$

  where $\beta_t \in (0, 1)$ adjusts the scale of the variance.

- **Reverse Process:** The reverse process starts from $\mathbf{x}_T \sim \mathcal{N}(0, I)$ and learns a parametric distribution $p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$ to invert the diffusion process of Eq. 3.1 gradually. Each step of the reverse process is defined as:

$$p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_\theta^2\left(t\right)\mathbf{I}\right), \tag{3.2}$$

  where $\mu_\theta\left(\mathbf{x}_t, t\right)$ and $\sigma_\theta^2(t)$ are learnable means and variances predicted by neural networks.

While there exists a tractable variational lower-bound (VLB) on $\log p_\theta\left(\mathbf{x}_0\right)$, [**38**] simplifies the loss function of continuous diffusion to:

$$\mathcal{L}_{\text{simple}} = \sum_{t=1}^{T} \left\| \mathbf{x}_0 - \tilde{f}_\theta\left(\mathbf{x}_t, t\right) \right\|^2, \tag{3.3}$$

where $\tilde{f}_\theta\left(\mathbf{x}_t, t\right)$ is the reconstructed $\mathbf{x}_0$ at step t.

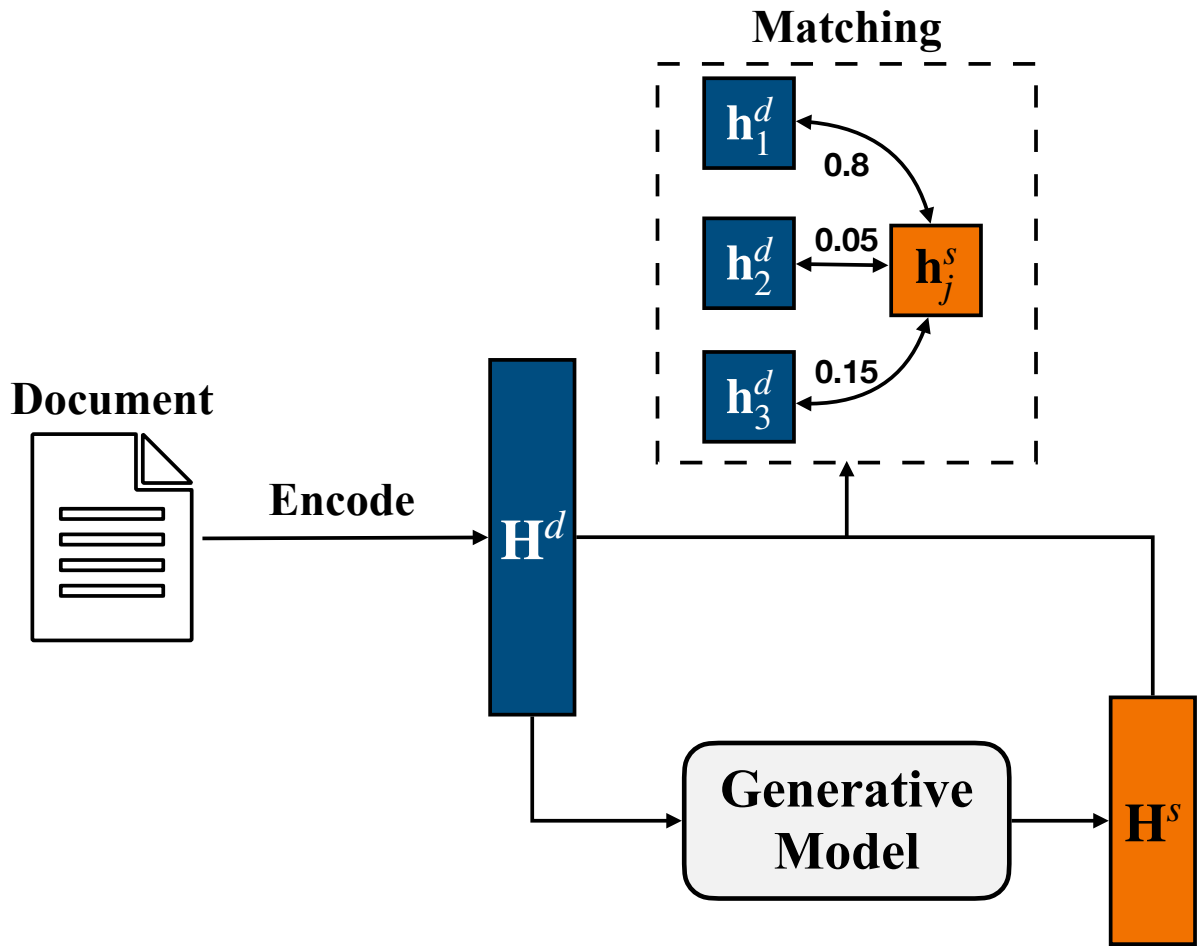

FIGURE 3.1. The proposed generation-enhanced extractive summarization framework. The model first conditionally generates desired summary embeddings and then extracts sentences based on representation matching.

3.3.2. Problem Formulation.

Given a document with n sentences as $D = \{s_1^d, s_2^d, ..., s_n^d\}$, extractive summarization system aims to form a $m(m \ll n)$ sentences summary $S = \{s_1^s, s_2^s, ..., s_m^s\}$ by directly extracting sentences from the source document. Most existing work formulates it as sequence labeling and gives each sentence a $\{0, 1\}$ label, where label 1 indicates that the sentence will be included in summary S. Since extractive ground-truth labels (ORACLE) are not available for human-written gold summary, it is common to use a greedy algorithm to generate an ORACLE consisting of multiple sentences which maximize the ROUGE-2 score against the gold summary following [**77**].

In contrast, we propose a summary-level framework with generative model augmentation as shown in Figure 3.1. Formally, we train a diffusion model with the *reverse process* $p_\theta(\tilde{\mathbf{H}}^s_{t-1}|\tilde{\mathbf{H}}^s_t, \mathbf{H}^d)$ to directly generate the desired summary sentence representations $\tilde{\mathbf{H}}^s_{t-1} = [\tilde{\mathbf{h}}^s_1, \tilde{\mathbf{h}}^s_2, ..., \tilde{\mathbf{h}}^s_m] \in \mathbb{R}^{m\times h}$, where $\tilde{\mathbf{h}}^s_j$ is the vector representing the j-th summary sentence at diffusion step $t-1$. The model then extracts summary sentences based on the matching between the generated summary sentence representations after T reverse steps $\tilde{\mathbf{H}}^s_0 = [\tilde{\mathbf{h}}^s_1, \tilde{\mathbf{h}}^s_2, ..., \tilde{\mathbf{h}}^s_m]$ and the document sentence embeddings $\mathbf{H}^d = [\mathbf{h}^d_1, \mathbf{h}^d_2, ..., \mathbf{h}^d_n]$. The matching score for the j-th sentence in the output s^s_j with the document is defined as:

$$\tilde{\mathbf{y}}_j = \text{softmax}(\tilde{\mathbf{h}}^s_j \cdot \mathbf{H}^{d^T}). \tag{3.4}$$

Here we use dot product as the similarity measurement and then extract the sentence with the highest matching score for each generated summary sentence.

Our approach operates on the summary level by generating all summary sentence representations simultaneously. We adopt continuous diffusion models for sentence embedding generation in this context.

3.4. Method

In this section, we introduce the detailed design of DiffuSum. DiffuSum consists of two major modules: a sentence encoding module and a diffusion module, which will be introduced in Section 3.4.1 and Section 3.4.2, respectively. After that, we explain how we optimize our model and conduct inference in Section 3.4.3. The overall model architecture of DiffuSum is also illustrated in Figure 3.2.

3.4.1. Sentence Encoding Module.

In order to generate desired summary sentence embeddings, we first build a contrastive sentence encoding module to transfer discrete text inputs $D = \{s^d_1, s^d_2, ..., s^d_n\}$ into continuous vector representations $\mathbf{H}^d = [\mathbf{h}^d_1, \mathbf{h}^d_2, ..., \mathbf{h}^d_n] \in \mathbb{R}^{n\times h}$, where h is the dimension of the encoded sentence representations.

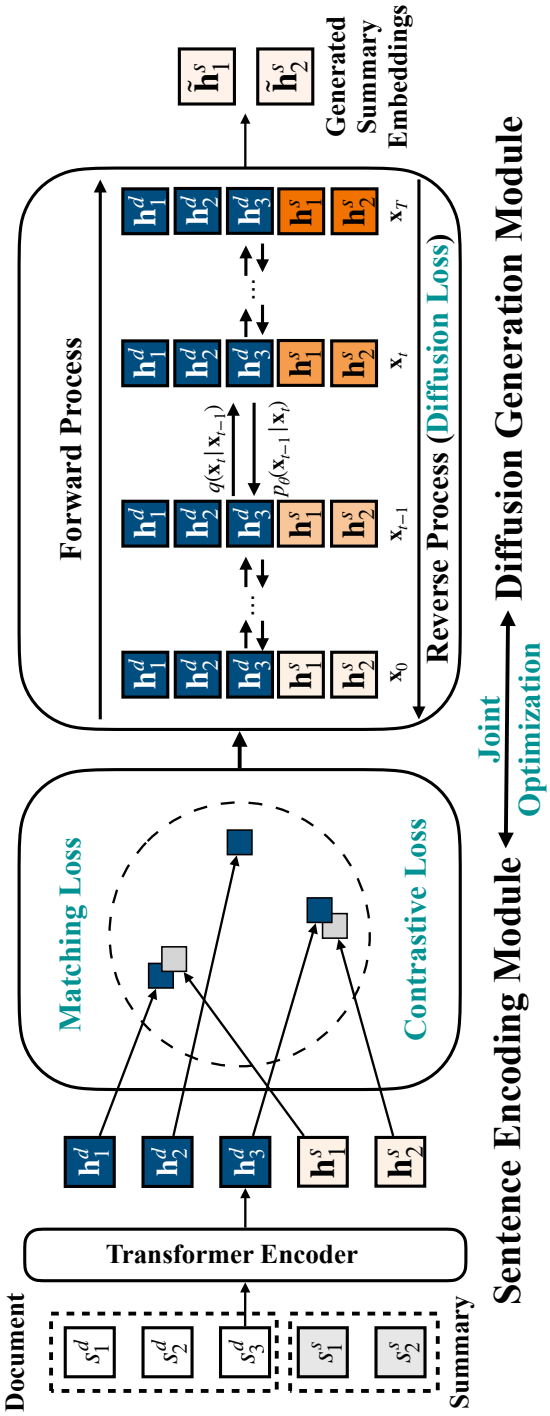

FIGURE 3.2. The overall architecture of DiffuSum. The input document is passed to the sentence encoding module and the diffusion generation module. DiffuSum will generate the desired summary sentence representations for inference.

Specifically, we first obtain the initial representations of sentences $\mathbf{E}^d = [\mathbf{e}_1^d, \mathbf{e}_2^d, ..., \mathbf{e}_n^d]$ with Sentence-BERT [**92**]. Note that the Sentence-BERT is only used for initial sentence embedding, but is not updated during training. The initial representations are then fed into a stacked transformer layer followed by a projection layer to obtain contextualized sentence representations $\mathbf{h}_i^d$:

$$\mathbf{h}_i^d = \text{MLP}(\text{Transformer}(\mathbf{e}_i^d)). \tag{3.5}$$

The same encoding process is applied to the summary sentences $S = \{s_1^s, s_2^s, ..., s_m^s\}$ to obtain encoded summary sentence representations $\mathbf{H}^s = [\mathbf{h}_1^s, \mathbf{h}_2^s, ..., \mathbf{h}_m^s] \in \mathbb{R}^{m \times h}$. The encoded document sentence representations $\mathbf{H}^d$ and summary sentence representations $\mathbf{H}^s$ are then concatenated as $\mathbf{H}^{in} = \mathbf{H}^d \| \mathbf{H}^s \in \mathbb{R}^{(n+m) \times h}$ and will be passed to the diffusion generation module.

To ensure the sentence encoding module produces accurate and distinguishable representations, we introduce a matching loss $\mathcal{L}_{\text{match}}$ and a multi-class supervised contrastive loss $\mathcal{L}_{\text{contra}}$ to optimize the module, which is defined as follows.:

Matching Loss We first introduce a matching loss to ensure an accurate matching between the encoded document and summary sentence representations. Formally, for the j-th encoded summary sentence representation $\mathbf{h}_j^s$, we generate its encoding matching scores $\hat{\mathbf{y}}_j$ by computing the dot product with document representations followed by a softmax function:

$$\hat{\mathbf{y}}_j = \text{softmax}(\mathbf{h}_j^s \cdot \mathbf{H}^{d^{\mathrm{T}}}). \tag{3.6}$$

Then we have the encoding matching loss $\mathcal{L}_{\text{match}}$ as the cross-entropy between our encoding matching score $\hat{\mathbf{y}}_j$ and the ground truth extractive summarization label (ORACLE) $\mathbf{y}_j$:

$$\mathcal{L}_{\text{match}} = \sum_{j=1}^{m} \text{CrossEntropy}\left(\mathbf{y}_j, \hat{\mathbf{y}}_j\right). \tag{3.7}$$

Contrastive Loss The sentence encoding module also needs to ensure the encoded summary sentence embeddings $[\mathbf{h}_1^s, \mathbf{h}_2^s, ..., \mathbf{h}_m^s]$ are diverse and distinguishable. Thus, we introduce the multi-class supervised contrastive loss [**48**] to push the summary sentence representation closer to its corresponding document sentence representation while keeping it away from other sentence embeddings.

Given the sentence contextual representations $\mathbf{H}^{in} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{n+m}] \in \mathbb{R}^{(n+m)\times h}$, the contrastive label $\mathbf{y}^c$ is defined as:

$$y_p^c = \begin{cases} q, \text{ if } p \leq n \text{ and } s_p^d = s_q^s \\ q, \text{ if } p = n + q \\ 0, \text{ otherwise} \end{cases}, \tag{3.8}$$

where $q \in \{1, 2, \cdots, m\}$ and y_p^c is the p-th element of $\mathbf{y}^c$. The contrastive loss $\mathcal{L}_{\text{contra}}$ is defined as:

$$\begin{aligned} \mathcal{L}_{\text{contra}} &= \frac{-1}{2N_{y_p^c} - 1} \sum_{p=1}^{n+m} \mathcal{L}_{\text{contra}}^p, \\ \mathcal{L}_{\text{contra}}^p &= \sum_{\substack{q=1;q\neq p;\\ y_q^c = y_p^c}}^{n+m} \log \frac{\exp\left(\mathbf{h}_p \cdot \mathbf{h}_q^T / \tau\right)}{\sum_{k=1;p\neq k}^{n+m} \exp\left(\mathbf{h}_p \cdot \mathbf{h}_k^T / \tau\right)}, \end{aligned} \tag{3.9}$$

where $N_{y_p^c}$ is the total number of sentences in the document that have the same label y_p^c ($N_{y_p^c} = 2$ in our case) and τ is a temperature hyperparameter.

The overall optimizing objective for the sentence encoding module $\mathcal{L}_{\text{se}}$ is defined as:

$$\mathcal{L}_{\text{se}} = \mathcal{L}_{\text{match}} + \gamma \mathcal{L}_{\text{contra}}, \tag{3.10}$$

where γ is a rescaling factor that adjusts the diversity of the sentence representations.

3.4.2. Diffusion Generation Module.

After obtaining the input encoding $\mathbf{H}^{in} = \mathbf{H}^d \| \mathbf{H}^s$, we adopt the continuous diffusion model to generate desired summary sentence embeddings conditionally. As described in

Section 3.3.1, our diffusion generation module adds Gaussian noise gradually through the forward process and fits a stacked Transformer to invert the diffusion in the reverse process.

We first perform one-step Markov transition $q(\mathbf{x}_0|\mathbf{H}^{in}) = \mathcal{N}(\mathbf{H}^{in}, \beta_0\mathbf{I})$ for the starting state $\mathbf{x}_0 = \mathbf{x}_0^d||\mathbf{x}_0^s$. Note that the initial Markov transition is applied to both document and summary sentence embeddings.

We then start the forward process by gradually injecting partial noise to summary embeddings $\mathbf{x}^s$ and leaving document embeddings unchanged $\mathbf{x}^d$ similar to [**32**]. This enables the diffusion model to generate conditionally on the source document. At step t of the forward process $q(\mathbf{x}_t^s|\mathbf{x}_{t-1})$, the noised representations is $\mathbf{x}_t$:

$$\mathbf{x}_t = \mathbf{x}_0^d||\mathcal{N}\left(\mathbf{x}_t^s; \sqrt{1-\beta_t}\mathbf{x}_{t-1}^s, \beta_t\mathbf{I}\right), \tag{3.11}$$

where $t \in \{1, 2, \cdots, T\}$ for a total of T diffusion steps and $||$ represents concatenation.

Once the partially noised representations are acquired, we conduct the backward process to remove the noise of summary representations given the condition of the sentence representations of the previous step:

$$p_\theta\left(\mathbf{x}_{t-1}^s|\mathbf{x}_t\right) = \mathcal{N}\left(\mathbf{x}_{t-1}^s; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_\theta^2\left(t\right)\mathbf{I}\right), \tag{3.12}$$

where $\mu_\theta(\cdot)$ and $\sigma_\theta^2(\cdot)$ are parameterized models (stacked Transformer in our case) to predict the mean and standard variation at diffusion step $t-1$. The final output of the diffusion module is the generated summary sentence representations after T reverse steps $\tilde{\mathbf{H}}_0^s = [\tilde{\mathbf{h}}_1^s, \tilde{\mathbf{h}}_2^s, ..., \tilde{\mathbf{h}}_m^s]$. We optimize the diffusion generation module with diffusion loss $\mathcal{L}_{\text{diffusion}}$ defined as:

$$\begin{aligned}\mathcal{L}_{\text{diffusion}} = &\sum_{t=2}^{T}\left\|\mathbf{x}_0 - \tilde{f}_\theta\left(\mathbf{x}_t, t\right)\right\|^2 + \\ &\left\|\mathbf{H}^{in} - \tilde{f}_\theta\left(\mathbf{x}_1, 1\right)\right\|^2 + \mathcal{R}\left(\mathbf{x}_0\right),\end{aligned} \tag{3.13}$$

where $\tilde{f}_\theta(\mathbf{x}_t, t)$ is the reconstructed $\mathbf{x}_0$ at step t and $\mathcal{R}(\mathbf{x}_0)$ is a L-2 regularization term.

3.4.3. Optimization and Inference.

We jointly optimize the sentence encoding module and the diffusion generation module in an end-to-end manner. The overall training loss of DiffuSum can be represented as:

$$\mathcal{L} = \mathcal{L}_{\text{se}} + \eta \mathcal{L}_{\text{diffusion}} \tag{3.14}$$

where η is a balancing factor of sentence encoding module loss $\mathcal{L}_{\text{se}}$ and diffusion generation module loss $\mathcal{L}_{\text{diffusion}}$.

For inference, DiffuSum first obtains encoded document representations $\mathbf{H}^d$, followed by a one-step Markov transition $q(\mathbf{x}_0^d|\mathbf{H}^d)$. Then we random sample m Gaussian noise embeddings as initial summary sentence representations $\mathbf{x}_T^s \in \mathbb{R}^{m \times h}$ and concatenate it with document representations to get the input $\mathbf{x}_T^{in} = \mathbf{x}_0^d || \mathbf{x}_T^s$ for diffusion step T. Then DiffuSum applies the learned reverse process (generation process) to remove the Gaussian noise iteratively and get the output summary sentence representations $\tilde{\mathbf{H}}_0^s = [\tilde{\mathbf{h}}_1^s, \tilde{\mathbf{h}}_2^s, ..., \tilde{\mathbf{h}}_m^s]$.

DiffuSum then calculates the matching between the generated summary representation $\tilde{\mathbf{h}}_i^s$ and the document representation $\mathbf{H}^d$ to obtain prediction label $\tilde{\mathbf{y}}_i^{pred}$ as in Eq. 3.4. We extract the sentence with the highest score for each generated summary sentence representation and form the summary.

3.5. Experiment

3.5.1. Experimental Setup.

Datasets: We conduct experiments on three benchmark summarization datasets: CNN/DailyMail, XSum, and PubMed. The detailed statistics of each dataset are shown in Table 3.1.

CNN/DailyMail [**37**] is the most widely adopted summarization dataset, containing news articles paired with corresponding human-written news highlights as summaries. In this work, we utilize the non-anonymized version of the dataset and adhere to the common

training, validation, and testing splits, which consist of 287,084/13,367/11,489 examples, respectively.

XSum [**79**] is a one-sentence news summarization dataset where all summaries are professionally written by the original authors of the documents. In this work, we adhere to the common training, validation, and testing splits, which consist of 204,045/11,332/11,334 examples, respectively.

PubMed [**16**] is a scientific paper summarization dataset of long documents. We follow the setting in [**136**] and use the introduction section as the article and the abstract section as the summary. The training/validation/testing split is (83,233/4,946/5,025).

Dataset	**Domain**	**Doc #words**	**Sum #words**	**#Ext**
CNN/DM	News	766.1	58.2	3
XSum	News	430.2	23.3	2
PubMed	Paper	444	209.5	6

TABLE 3.1. Statistics of the experimental datasets. Doc # words and Sum # words refer to the average word number in the source document and summary. # Ext refers to the number of sentences to extract.

Baselines: We compare DiffuSum with strong sentence-level baseline methods:

- Vanilla Transformer [**105**]
- Hierarchical encoder model HIBERT [**134**]
- PNBERT [**137**] that combines LSTM-Pointer with pre-trained BERT
- BERT-based extractive model BERTSum [**66**]
- BERTEXT [**6**] that augments BERT with reinforcement learning

We also compare DiffuSum with state-of-the-art summary-level approaches:

- contrastive Learning based re-ranking framework COLO [**2**]
- summary-level two-stage text matching framework MATCHSUM [**136**].

3.5.2. Implementation Details.

We use Sentence-BERT [**92**] checkpoint *all-mpnet-base-v2* for initial sentence representations. The dimension of the sentence representations h is set to 128. We use an 8-layer Transformer with 12 attention heads in our sentence encoding module and a 12-layer Transformer with 12 attention heads in the diffusion generation module. The hidden size of the model is set to 768, and temperature τ is set to 0.07. The scaling factors γ and η are set to 0.001 and 100, where γ is searched in the range of $[0.0001, 1]$ and η is searched within the range of $[10, 1000]$. We set the diffusion steps T to 500. Effects of hyperparameter T and h are discussed in section 3.6.2.

DiffuSum has a total of 13 million parameters and is optimized with AdamW optimizer [**70**] with a learning rate of $1e^{-5}$ and a dropout rate of 0.1. We train the model for 10 epochs and validate the performance by the average of ROUGE-1 and ROUGE-2 F-1 scores on the validation set.

Following the standard setting, we evaluate model performance with ROUGE[1] scores [**60**]. Specifically, ROUGE-1/2 scores measure summary informativeness, and the ROUGE-L score measures summary fluency. Single-run results are presented in the following sections with the default random seed of 101.

3.5.3. Experiment Results.

Results on CNN/DailyMail: Experimental results on the CNN/DailyMail dataset are presented in Table 3.2. The first block in the table comprises the extractive ground truth ORACLE (upper bound) and LEAD, which selects the first few sentences as a summary. The second block includes recent strong one-stage extractive baseline methods, along with our proposed model DiffuSum. The third section contains two-stage baseline methods that pre-select salient sentences. We adhere to the same setting and display the results of DiffuSum with the same pre-selection for a fair comparison.

[1]ROUGE: https://pypi.org/project/rouge-score/

Model	R-1	R-2	R-L
LEAD	40.43	17.62	36.67
ORACLE	52.59	31.23	48.87
One-stage Systems			
Transformer (2017)	40.90	18.02	37.17
HIBERT* (2019)	42.37	19.95	38.83
PNBERT* (2019)	42.69	19.60	38.85
BERTEXT* (2019)	42.76	19.87	39.11
BERTSum* (2019)	43.85	20.34	39.90
$\text{COLO}^*_{\text{Ext}}$ (2023)	44.58	21.25	40.65
DiffuSum (ours)	**44.62**	**22.51**	**40.34**
Two-stage Systems			
MATCHSUM*(BERT) (2020)	44.22	20.62	40.38
MATCHSUM*(Roberta)	44.41	20.86	40.55
DiffuSum (ours)	**44.83**	**22.56**	**40.56**

TABLE 3.2. Experimental results on CNN/DailyMail dataset. Models using pre-trained language models are marked with*.

According to the results, DiffuSum achieves new state-of-the-art performance under both one-stage and two-stage settings, particularly demonstrating a significant improvement in the ROUGE-2 score. The superior performance of DiffuSum highlights the effectiveness of our generation-augmented framework and underscores the great potential of applying diffusion models in text representation generation.

It's worth noting that most baseline methods incorporate pre-trained language model components, whereas our proposed framework, DiffuSum, trains Transformers from scratch and contains no pre-trained knowledge. We believe DiffuSum could achieve even better performance if combined with pre-trained knowledge, which we leave as a direction for future work.

Additionally, we observe that summary-level methods generally outperform sentence-level methods, indicating the necessity to address the inherent gap between them.

Model	PubMed			XSum		
	R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
ORACLE	45.12	20.33	40.19	25.62	7.62	18.72
LEAD	37.58	12.22	33.44	14.40	1.46	10.59
BERTSUM	41.05	14.88	36.57	22.86	4.48	17.16
MatchSUM	41.21	14.91	36.75	**24.86**	4.66	**18.41**
DiffuSum	**41.40**	**15.55**	**37.48**	24.00	**5.44**	18.01

TABLE 3.3. Experimental Results on PubMed and XSum datasets.

Results on XSum and PubMed: We also evaluate DiffuSum on PubMed and XSum datasets, representing datasets of different domains and summary lengths as shown in Table 3.3.

For datasets with longer summaries such as PubMed, DiffuSum demonstrates remarkably strong performance and surpasses state-of-the-art baselines. This strong performance attests to the capability of our model to handle longer input contexts and complex generations effectively.

Moreover, our summary-level setting proves advantageous for datasets with longer summaries by considering dependencies among summary sentences. This holistic approach enables DiffuSum to capture the intricate relationships between sentences and produce more informative and coherent summaries.

For datasets with shorter summaries such as XSum, DiffuSum achieves comparable performance to state-of-the-art approaches, with a notably higher ROUGE-2 score. Short-summary datasets tend to be simpler for matching-based methods like MatchSum since the candidate pool is much smaller. Despite this, DiffuSum still demonstrates its effectiveness in capturing the essence of the input documents and generating high-quality summaries.

Overall, DiffuSum achieves a comparable or even better performance compared to pre-trained language model-based baseline methods. These results underscore the effectiveness of DiffuSum across summarization datasets with varying lengths.

3.6. Analysis

We further analyze the performance of DiffuSum by conducting an ablation study in Section 3.6.1 and a hyperparameter sensitivity study in Section 3.6.2. Additionally, we validate the cross-domain generalization capability in Section 3.6.3 and visualize the generated summary sentence representations in Section 3.6.4.

3.6.1. Ablation Study.

Model	R-1	R-2	R-L
DiffuSum	**44.83**	**22.56**	**40.56**
w/o Sentence-BERT	43.53	21.63	40.23
w/o ORACLE	39.19	17.12	34.38
w/o Contrastive Loss	44.57	22.35	40.34

TABLE 3.4. Ablation study results on CNN/DailyMail dataset.

To understand the strong performance of DiffuSum, we conduct an ablation study by removing model components of the sentence encoding module and present the results in Table 3.4.

The second row shows a performance drop when replacing the initial sentence representation from Sentence-BERT to BERT-base encoder [**19**]. This drop indicates that sentence-level information is necessary for the success of DiffuSum. The third row shows that replacing the ORACLE with abstractive reference summaries degrades performance.

Regarding the sentence encoding loss, both the matching loss and contrastive loss benefit the overall model performance according to rows 4 and 5. The matching loss is critical to the model, as the performance drops dramatically by more than 40% without it.

Model	R-1	R-2	R-L
DiffuSum(T=500, h=128)	**44.83**	**22.56**	**40.58**
DiffuSum(T=500, h=64)	43.36	21.27	39.89
DiffuSum(T=500, h=256)	44.53	22.49	40.27
DiffuSum(T=50, h=128)	42.60	19.71	38.96
DiffuSum(T=100, h=128)	44.61	22.24	40.32
DiffuSum(T=1000, h=128)	44.65	22.36	40.37
DiffuSum(T=2000, h=128)	44.64	22.37	40.40

TABLE 3.5. The performance of DiffuSum with different hyperparameter settings on CNN/DM dataset.

These results demonstrate the importance of jointly training a sentence encoder that produces accurate and diverse sentence representations with the generation module.

3.6.2. Hyperparameter Sensitivity.

We also investigate the influence of two important hyperparameters of our diffusion generation module: diffusion steps T and the sentence representations dimension h, as shown in Table 3.5.

The first row represents our best model, while the second block shows the performance of DiffuSum with different sentence representation dimensions. We observe a significant performance drop when setting the dimension to 64, indicating severe information loss when shrinking the sentence dimension too much. Additionally, there is a slight performance drop when the dimension is set to 256, suggesting that a dimension that is too large may introduce more noise.

In the third block, we examine the influence of diffusion steps. We find that the model performance initially increases with more diffusion steps, but then starts to decrease and oscillate if steps continue to increase. We argue that too few diffusion steps may not fully remove the injected noise in the forward pass, while too many steps may introduce too much noise for the model to recover effectively.

3.6.3. Cross-dataset Evaluation.

Train \ Test	CNN/DM	XSum	PubMed
CNN/DM	44.83/22.56	21.35/3.85	39.83(-1.57)/13.25
XSum	42.85/21.37	24.0/5.44	38.71(-2.69)/12.93

TABLE 3.6. ROUGE-1 and ROUGE-2 results for cross-dataset evaluation.

We also observe that DiffuSum demonstrates a strong cross-dataset adaptation ability. As depicted in Table 3.6, the model trained on the news domain datasets (CNN/DM and XSum) achieves comparable performance (only 1.57 and 2.69 ROUGE-1 drops) when directly evaluated on the scientific paper domain.

These cross-dataset results highlight the robustness of our generation-augmented framework and suggest the potential to build a generalized extractive summarization system capable of effectively summarizing diverse types of documents across different domains.

3.6.4. Representation Analysis.

We also analyze the quality of the generated sentence representations. We employ T-SNE [**103**] to reduce the dimensionality of the sentence representations to 2, and visualize both the encoded sentence representations and the generated summary sentence representations in Figure 3.3.

In the figure, the blue dots represent non-summary sentences, while the red dots represent summary sentences (ORACLE) obtained from our sentence encoding module. The green dots represent summary sentence representations reconstructed by our diffusion generation module. We observe that most of the ORACLE sentences cluster on the right side of the plot. This indicates that our contrastive encoder can effectively distinguish ORACLE sentences from non-summary sentences.

Additionally, we notice that the sentence representations generated by the diffusion module (green) are closely clustered around the original summary representations (red). This

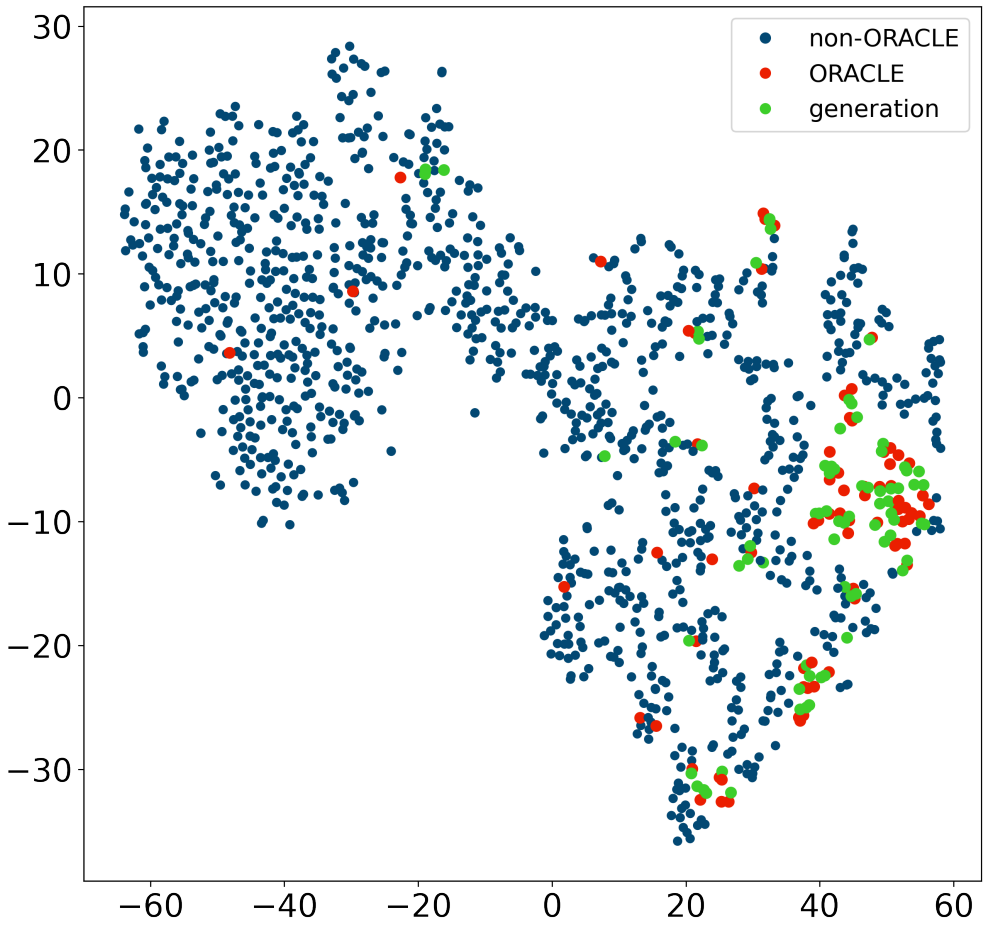

FIGURE 3.3. T-SNE visualization of sentence embeddings from 25 CNN/DM dataset documents.

finding demonstrates that our diffusion generation module is proficient in reconstructing sentence representations from random Gaussian noise.

3.7. Conclusion

This chapter introduces a novel paradigm for extractive summarization with generation augmentation. Rather than sequentially labeling sentences, DiffuSum directly generates the desired summary sentence representations using diffusion models and subsequently extracts summary sentences based on representation matching. Experimental results on three benchmark datasets demonstrate the effectiveness of DiffuSum.

This work represents the first attempt to adapt diffusion models for summarization tasks. Future research directions could explore various applications of continuous diffusion models in both extractive and abstractive summarization tasks, further advancing the state of the art in automatic text summarization.

CHAPTER 4

Balancing Summary Salience and Diversity

4.1. Introduction

Multi-document summarization (**MDS**) is one of the essential tools for obtaining core information from a collection of documents written on the same topic. It seeks to find the main ideas from multiple sources with diversified messages. Despite recent advances in MDS system designs [**75**, **112**], three major challenges hinder its development:

First, existing extractive multi-document summarization systems rely on optimization with individual scoring, which becomes sub-optimal when extracting multiple summary sentences [**136**]. A typical individual system scores each candidate summary with only measurements of the newly added sentences during inference. In contrast, the holistic system simultaneously measures all summary sentences and the relations among them. Despite recent efforts in holistic methods on a single document [**2**, **136**], how to extract sentences holistically for multi-document summarization remains open. In this work, we propose an inference method that holistically optimizes the extractive summary under the multi-document setting.

Second, multi-document summarization naturally contains excessively redundant information [**53**]. An ideal summary should provide important information with diversified perspectives [**81**]. In Figure 4.1, we show a salient and diversified summary versus a salient but redundant summary. A salient and diversified summary often covers the information thoroughly, while a salient but redundant summary is usually incomplete. Different from existing approaches [**15**, **114**] for limiting the repetitions, we introduce **S**ubset **R**epresentative **I**ndex (**SRI**), a holistically balanced measurement between importance and diversity for extractive multi-document summarization.

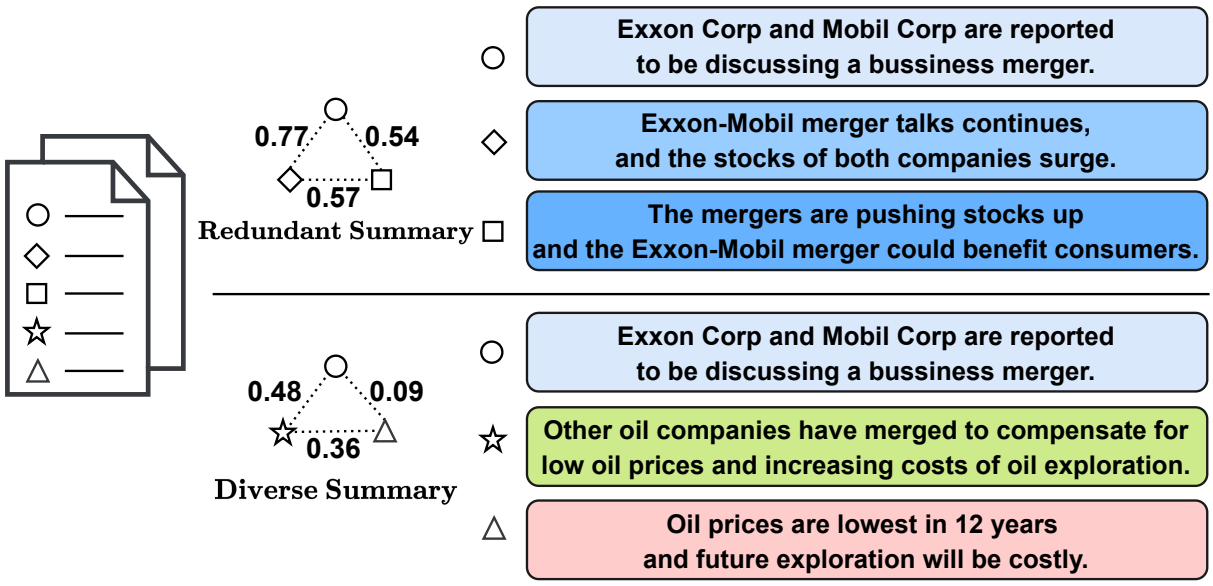

FIGURE 4.1. An example of a diverse summary vs. a redundant summary. Sentences in the redundant summary have higher semantic similarity than a diverse summary.

Finally, recent deep learning-based supervised summarization methods are data-driven and require a massive number of high-quality summaries in the training data. Nevertheless, hiring humans to write summaries is always expensive, time-consuming, and thus hard to scale up. This problem becomes more severe for multi-document summarization since it requires more effort to read more documents. Therefore, existing multi-document summarization datasets are either small-scale [**18**, **82**] or created by acquiring data from the Internet with automatic alignments [**3**, **27**] that could be erroneous. Here, we propose an unsupervised multi-document summarization method to tackle the low-resource issue. It can further benefit unsupervised multi-document summarization, with the adaptive setting using large-scale high-quality single-document summarization data (e.g., CNN/DailyMail [**37**]).

In this chapter, we present a novel framework for unsupervised extractive multi-document summarization, aiming to holistically select the extractive summary sentences. The framework contains the holistic beam search inference method associated with holistic measurements named **SRI** (**S**ubset **R**epresentative **I**ndex). The SRI is designed as a holistic measurement for balancing the importance of individual sentences and the diversity among sentences

within a set. To address data sparsity, we propose to calculate SRI in both unsupervised and adaptive manners. Unsupervised SRI relies on centrality from graph-based methods [**26**, **75**] for subset importance measurement, while adaptive SRI uses BERT [**19**] fine-tuned on the single-document summarization (SDS) corpus for sentence importance measurement. Our method shows performance improvements in both the summary informativeness and diversity scores, indicating our approach can achieve better coverage of documents while maintaining the gist information of multi-documents. We highlight the contributions of our work as follows:

- We propose a novel holistic framework for multi-document extractive summarization. Our framework incorporates a holistic inference method for summary sentence extraction and a holistic measurement called the Subset Representative Index (SRI) to balance the importance and diversity of a subset of sentences.
- We propose two unsupervised methods to measure SRI, using graph-based centrality or adapting from a single-document corpus
- We conduct extensive experiments on several benchmark datasets, and the results demonstrate the effectiveness of our paradigm under both unsupervised and adaptive settings. Our findings suggest that effectively modeling sentence importance and pairwise sentence similarity is crucial for extracting diverse summaries and improving summarization performance.

4.2. Related Works

4.2.1. Multi-document Summarization.

Traditional non-neural approaches to multi-document summarization have been both extractive [**12**, **26**, **75**] and abstractive [**30**]. Recent neural MDS systems rely on Transformer-based encoder-decoder models to process the integrated long documents with hierarchical inter-text attention [**27**, **65**], or attention across representations of different granularity [**44**]. This work focuses on unsupervised MDS scenarios where gold reference summaries are unavailable. Prior unsupervised MDS systems are mostly graph-based [**26**]. Similar to our

adaptive setting, [**53**] proposed adapting the encoder-decoder framework from a single document corpus, but our work focuses on the extractive summarization setting with holistic inference.

4.2.2. Sentence Importance Measurements.

Most works formulate extractive summarization as a sequence classification problem and use sequential neural models with different encoders like recurrent neural networks [**13**, **78**] and pre-trained language models [**66**]. The prediction probabilities are treated as the importance measurement of sentences. On the other hand, unsupervised graph-based methods calculate the importance of sentences with node centrality and rank them for the summaries, including TextRank [**75**], LexRank [**26**], PacSum [**135**], and its variants [**58**]. Recent researches [**107**, **115**, **123**] have explored Graph Neural Networks to obtain better representations for each sentence. Graph methods have merits in considering implicit document structure and adapting regardless of the input length.

4.2.3. Redundancy.

Considering only the importance of sentences for the summary leads to repeated information, and resolving redundant contents is an essential problem in the extractive summarization system. Traditional methods to tackle redundancy rely on discrete optimization problems like Maximal Marginal Relevance (MMR) [**12**], Determinantal Point Process (DPP) [**52**], and submodular selection [**61**]. rigram blocking is introduced to explicitly reduce redundancy by avoiding sentences that share a 3-gram with the previously added one [**66**]. [**83**] first adopted trigram blocking in decoding for abstractive summarization. [**72**] proposed sentence filtering and beam search methods for extractive summarization sentence selection. [**138**] propose a model jointly learning to score and select sentences inspired by MMR. [**114**] conducted a systematic study of redundancy in long documents.

4.3. Method

This section provides a detailed description of our proposed holistic MDS summarization framework. We first explain how we formulate the MDS problem holistically. The overall architecture of our holistic framework is shown in Figure 4.2, which includes holistic inference methods for summary sentence extraction and a new holistic measurement, the Subset Representative Index (SRI).

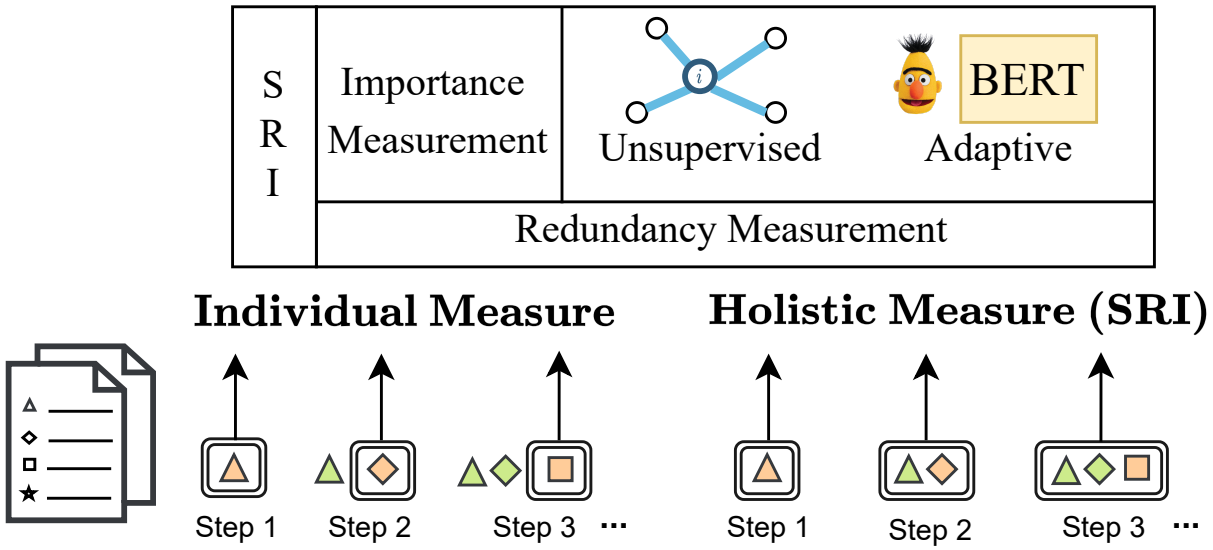

FIGURE 4.2. Illustration of the proposed holistic framework for multi-document summarization. The individual inference only resorts to each candidate, while the holistic inference is based on all candidates. Orange and green indicate newly added sentences and already added ones to the summary, respectively.

4.3.1. Problem Formulation.

Multi-document summarization typically takes a collection of n documents $\mathcal{D} = \{D^{(1)}, \ldots, D^{(n)}\}$ as inputs. Each document contains a varying number of sentences $D^{(i)} = \{s_0^{(i)}, \ldots, s_{l_i}^{(i)}\}$, where l_i is the number of sentences in the i-th document. Let $\mathcal{S}$ be the collection of all sentences, i.e. $\mathcal{S} = D^{(1)} \cup \cdots \cup D^{(n)}$. Additionally, let $e_{i,j}$ denote the similarity score between sentence s_i and sentence s_j. Our goal is to select a representative subset of sentences $\mathcal{S}' \subset \mathcal{S}$ that maximizes the total importance of the subset while minimizing the redundancy within sentences in the subset at the same time.

4.3.2. Holistic Inference.

Most existing approaches for unsupervised extractive summarization formulate it as an individual sentence ranking problem. They first calculate a measurement $\mathcal{M}(s_i)$ (e.g., sentence importance) for each sentence $s_i \in \mathcal{S}$ and rank all sentences in $\mathcal{S}$ accordingly. For summary inference, they directly use an individual greedy method that adds one sentence with the highest ranking at a time until the desired total number of summary sentences is reached.

In contrast, a holistic summarization method should evaluate a subset of sentences $\mathcal{M}(\mathcal{S}')$ as a whole, then select the best subset $\mathcal{S}'$. This setting formulates the holistic summary inference into a best subset selection problem, which has exponential time complexity.

To address the exponential time complexity issue, we propose several holistic inference methods for summary sentence extraction. These methods optimize subsets of sentences using subset measurements, as opposed to the individual greedy inference method. We describe the different variants of the proposed method as follows:

Holistic Greedy Method: The most straightforward way to address the exponential time complexity issue is to adopt a greedy approach. Similar to the individual greedy method, the holistic greedy method also adds one sentence at a time. However, it picks the sentence using a subset measurement that takes into account the previously selected sentences. Formally, at each step, the method selects the sentence that maximizes the following objective:

$$\text{argmax}_{s_i \in \mathcal{S} \setminus \mathcal{S}'} \mathcal{M}(\mathcal{S}' \cup \{s_i\}), \tag{4.1}$$

where $\mathcal{S}'$ represents the previously selected sentences.

Holistic Exhaustive Search: It is a brute-force method that considers every possible subset with the desired number of sentences. However, due to the exponential computation

time, it is necessary to first filter out low-importance candidates using $\mathcal{M}(\{s_i\})$ to reduce the search space.

Holistic Beam Inference: We also propose Holistic Beam Inference, which balances the trade-off between search space size and efficiency. It is a more advanced holistic inference method that adapts the beam-search decoding algorithm. We illustrate the algorithm in Algorithm 2. At each step, it considers the top-k candidate subsets, which enlarges the search space and therefore has a higher chance of finding a better subset solution compared to the holistic greedy method. Meanwhile, the algorithm has linear time complexity, making it more efficient than the holistic exhaustive search method.

Algorithm 2: Holistic Beam Inference

Input: set of sentences $\mathcal{S}$, Measurement $\mathcal{M}(\cdot)$
Parameter: # summary sentences $N < |\mathcal{S}|$, beam size k
Output: the selected subset $\mathcal{S}'$

The candidate set $\mathcal{C} \leftarrow \{\varnothing\}$
for N times **do**
  The beam set $\mathcal{C}' \leftarrow \{\varnothing\}$
  for $\mathcal{X} \in \mathcal{C}$ **do**
    $\mathcal{X}' \leftarrow \text{arg-top-}k\text{-max}_{s \in \mathcal{S} \setminus \mathcal{X}} \mathcal{M}(\mathcal{X} \cup \{s\})$
    for $x \in \mathcal{X}'$ **do**
      Add $\mathcal{X} \cup \{x\}$ to $\mathcal{C}'$
    end for
  end for
  $\mathcal{C} \leftarrow \text{arg-top-}k\text{-max}_{\mathcal{X} \in \mathcal{C}'} \mathcal{M}(\mathcal{X})$
end for
return $\text{argmax}_{\mathcal{X} \in \mathcal{C}} \mathcal{M}(\mathcal{X})$

4.3.3. Subset Representative Index.

To complement the holistic inference methods, we propose a new subset measurement, Subset Representative Index (SRI), denoted as $\mathcal{M}(\mathcal{S}')$. It balances the importance measurement $\mathcal{I}(\mathcal{S}')$ and redundancy measurement $\mathcal{R}(\mathcal{S}')$.

An ideal extractive summary should select the most representative subset from a collection of the input sentences, maximizing the total non-redundant salient information passed to the

user. SRI is a holistic subset measurement that balances the importance and redundancy of a subset of sentences from the source documents. Formally, we define SRI as below:

$$\mathcal{M}(\mathcal{S}') = \mathcal{I}(\mathcal{S}') - \lambda \cdot \mathcal{R}(\mathcal{S}'), \tag{4.2}$$

where $\mathcal{I}(\mathcal{S}')$ measures the informativeness of a set of sentences, and $\mathcal{R}(\mathcal{S}')$ measures the redundancy within the set. The parameter λ is used to control the weight of the redundancy in the overall SRI score. We detail the methods for measuring the set importance and redundancy in an unsupervised manner as follows:

Graph-Based Importance Measurement: To measure the importance of sentences, we use a graph-based approach. We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where node $v_i \in \mathcal{V}$ represents sentence $s_i \in \mathcal{S}$, and edge $e_{i,j} \in \mathcal{E}$ represents the similarity between sentence s_i and s_j. Our proposed approach for sentence similarity score employs a combination of two methods: TF-IDF and Sentence-BERT [**92**]. TF-IDF is used to encode sentences with surface-form similarity, while Sentence-BERT is used to encode sentences with semantic similarity:

$$e_{i,j} = \alpha \cdot \mathbf{c}_i^\top \mathbf{c}_j + (1 - \alpha) \cdot \mathbf{r}_i^\top \mathbf{r}_j, \tag{4.3}$$

where $\mathbf{c}_i$, $\mathbf{c}_j$, $\mathbf{r}_i$ and $\mathbf{r}_j$ are the corresponding TF-IDF features and sentence embeddings for the i-th and j-th sentences, respectively. The weight term $\alpha \in [0, 1]$ is a configurable hyperparameter to balance between statistical similarity and contextualized similarity.

Inspired from [**26**, **75**], we define the importance of a sentence as its node centrality in the graph, which is calculated as the sum of the weights of edges connected to the node representing this sentence:

$$\mathcal{I}(s_i) = \sum_{s_j \in \mathcal{S} \setminus s_i} e_{i,j}. \tag{4.4}$$

Similarly, the importance of a subset of sentences is defined as the total weights between the subgraph and the remaining graph:

$$\begin{aligned}\mathcal{I}(\mathcal{S}') =& \frac{1}{|\mathcal{S}| - |\mathcal{S}'|} \sum_{s_i \in \mathcal{S}', s_j \in \mathcal{S} \setminus \mathcal{S}'} e_{i,j} \\ \approx& \frac{1}{|\mathcal{S}|} \sum_{s_i \in \mathcal{S}', s_j \in \mathcal{S} \setminus \mathcal{S}'} e_{i,j}.\end{aligned} \tag{4.5}$$

Since $|\mathcal{S}'|$ is usually far smaller than $|\mathcal{S}|$ in summarization tasks, we can approximate the denominator by using $|\mathcal{S}|$ directly. This way, the subset importance only takes into account the relationship of the subset with the remaining sentences, rather than considering dependencies within the subset.

Adaptive Importance Measurement: In spite of the data sparsity issue in MDS, the Single Document Summarization (SDS) task has abundant high-quality labeled data [**16**, **37**, **79**]. We propose a method called adaptive importance measurement, which adapts SDS data for MDS importance measurement. This method utilizes the labeled data from SDS to train a model for predicting the importance of sentences in MDS.

In the adaptive setting, we fine-tune BERT [**19**] to serve as a sentence importance scorer on SDS datasets and then adapt the fine-tuned model to the target MDS datasets. Specifically, we first calculate the normalized salience of a sentence as:

$$\begin{aligned} f\left(s_i\right) &= \mathbf{v}^{\top} \tanh\left(\mathbf{W}_1 \mathbf{r}_i\right), \\ \text{salience}\left(s_i\right) &= \frac{f\left(s_i\right)}{\sum_{s_j \in D} f\left(s_j\right)}, \end{aligned} \tag{4.6}$$

where $\mathbf{W}$ is a trainable weight, and $\mathbf{r}_i$ is the contextualized representation of sentence s_i. Then, we fine-tune BERT to minimize the following loss:

$$\begin{aligned} R\left(s_i\right) &= \text{softmax}\left(\text{ROUGE}\left(s_i\right)\right), \\ \mathcal{L} &= -\sum_{D} \sum_{s_i \in D} R\left(s_i\right) \log \text{salience}(s_i). \end{aligned} \tag{4.7}$$

The fine-tuned BERT can be directly adapted to the MDS datasets and calculate the adaptive importance measurement for sentences.

Redundancy Measurement: The redundancy measurement for a subset of sentences $\mathcal{S}'$ is defined as the total similarity score of each sentence with its most similar counterpart. This measurement captures the degree of overlap between the sentences in the subset, indicating the level of redundancy present in the selected sentences:

$$\mathcal{R}(\mathcal{S}') = \sum_{s_i \in \mathcal{S}'} \max_{s_j \in \mathcal{S}' \setminus \{s_i\}} e_{i,j}. \tag{4.8}$$

Overall, we can calculate the Subset Representative Index (SRI) in both unsupervised and adaptive manners. Our holistic framework extracts summaries as a whole with the holistic inference method, which is guided by SRI to measure the importance and redundancy of a subset of sentences. This approach allows us to balance the importance and redundancy of a summary, making it more informative and coherent.

4.4. Experiments

In this section, we provide details on our experimental setup, including the datasets, evaluation metrics, baselines, and implementation details (Section 4.4.1). We then present the results of our model on benchmark MDS datasets in both unsupervised (Section 4.4.2) and adaptive (Section 4.4.3) settings.

4.4.1. Experimental Setting.

Dataset: We evaluate our unsupervised method on benchmark multi-document summarization datasets. Particularly, we use MultiNews [**27**], WikiSum [**63**], DUC-04 [**82**], and TAC-11 [**18**] datasets.

MultiNews is collected from a diverse set of news articles on newser.com. It is a large-scale dataset containing reference summaries written by professional editors. WikiSum is another large-scale dataset that provides documents and summaries from Wikipedia webpages, where

the documents come from the reference webpages of Wikipedia articles and top-10 Google searches, and the summaries are the lead section of the Wikipedia articles. We use the top-40 high-ranked paragraphs for the document inputs following [**65**]. For summary extraction, we use the average number of reference sentences: 10 and 5, respectively, on MultiNews and WikiSum.

For the DUC and TAC datasets, the task is to generate a succinct summary of up to 100 words from a set of 10 news articles. We report results on DUC-04 and TAC-11, which are standard test sets used in previous studies [**15**,**41**]. DUC-03 and TAC-08/09/10 are used for the validation set to tune hyperparameters. For the adaptive setting, we fine-tune BERT on a single document summarization dataset CNN/DailyMail [**37**] and directly adapt to MDS test sets. Table 4.1 shows the statistics of the datasets in detail.

Dataset	# test	# ref.	avg.w/doc	avg.w/sum
DUC-04	50	4	4,636	109.6
TAC-11	44	4	4,696	99.7
Multi-News	5,622	1	2,104	264.7
Wikisum	38,205	1	2,800	139.4
CNNDM(SDS)	11,489	1	766.1	58.2

TABLE 4.1. Detailed statistics of four multi-document datasets. #test denotes the number of document clusters in the test set, #ref denotes the number of reference summaries, avg.word(doc) denotes the average number of words in the source document cluster, avg.word(sum) denotes the average number of words in the ground truth summary.

Evaluation Metrics: The extracted summaries are evaluated against human reference summaries using ROUGE [**60**] with options `-n 2 -m -w 1.2 -c 95 -r 1000 -l 100` for DUC/TAC for the summarization quality. We report ROUGE-1, ROUGE-2, ROUGE-SU4, and ROUGE-L[1] that respectively measure the overlap of unigrams, bigrams, skip bigrams with a maximum distance of 4 words, and the longest common sequence between extracted summary and reference summary. To align with previous works, we report R-1, R-2, R-L

[1]Due to some legacy issues, some baselines report the original ROUGE-L, others report ROUGE-Lsum.

for Multinews and Wikisum datasets, and R-1, R-2, R-SU4 for DUC and TAC datasets. For all baseline methods, we report ROUGE results from their original papers if available or use results reported in [**14**, **62**]. We also report the measure of diversity for the generated summaries by calculating a unique n-gram ratio [**85**, **114**] defined as:

$$\text{uniq } n\text{-gram ratio} = \frac{\#\text{ uniq-}n\text{-gram}}{\#n\text{-gram}} \tag{4.9}$$

Baselines: We compare our methods with the following strong unsupervised summarization baselines:

- MMR [**12**] combines query relevance with information novelty in the context of summarization.
- LexRank [**26**] computes sentence importance based on eigenvector centrality in a graph representation of sentences.
- TextRank [**75**] adopts PageRank [**10**] to compute node centrality recursively based on a Markov chain model.
- SumBasic [**104**] is an extractive approach assuming words frequently occurring in a document cluster are more likely to be included in the summary.
- KL-Sum [**36**] uses a greedy approach to add a sentence to the summary to minimize the KL divergence.
- PRIMERA [**112**] is a pyramid-based pre-trained model for MDS that achieves state-of-the-art performance. We compare it under its zero-shot setting.

Implementation Details: We run all experiments with 88 Intel(R) Xeon(R) CPUs. We combine the surface indicator based on TF-IDF and contextualized embeddings. We treat each document cluster as a corpus and each sentence as a document when calculating the TF-IDF scores. We employ the pre-trained sentence-transformer [**92**] and extract sentence representations using a checkpoint of 'all-mpnet-base-v2'.

The graph edges with low similarity are treated as disconnected to emphasize the connectivity of the graph and avoid noisy edge connections. We keep a threshold $\tilde{e}$ for edge weights

such that edges with similarity scores smaller than $\tilde{e}$ will be set to 0. Here $\tilde{e}$ is controlled by a hyper-parameter to be tuned according to datasets. The final representation of edge weight between two sentences (s_i, s_j) is

$$e_{i,j} = \max(\text{sim}(s_i, s_j) - \tilde{e}, 0), \tag{4.10}$$

where $\tilde{e} = \min(e) + \theta\ (\max(e) - \min(e))$ is the threshold controlled by hyper-parameter θ. For exhaustive search, we filter out the sentences with low centrality and only keep the top 15 sentences at inference.

All hyper-parameters are tuned on validation sets on MultiNews and WikiSum and training sets on DUC and TAC. The best parameters are selected based on the highest R-1 score. More specific, for the balancing factor λ in SRI, we use $\{2^{-13}, 2^{-7}, 2^{-4}, 2^{-6}\}$ on DUC, TAC, MultiNews and WikiSum dataset. For α that weighted the contributions of TF-IDF and contextualized sentence similarity, we use 0.9 on News domain datasets and 0.8 on the WikiSum dataset. The edge weight threshold θ is $\{0, 0, 0.1, 0.1\}$ for DUC, TAC, MultiNews and WikiSum. As for beam search, we use beam size $\{4, 4, 4, 3\}$ on the corresponding datasets.

4.4.2. Unsupervised Summarization Results.

The unsupervised summarization results on four benchmark MDS datasets are shown in Table 4.2.

The summarization performance of our method outperforms strong unsupervised baselines. Note that MultiNews and WikiSum datasets provide abundant training samples and contain shorter input than the DUC or TAC datasets. Our method performs better than the pre-trained model, *PRIMERA*, with a zero-shot setting. Compared to the baseline (Sentence Greedy) that extracts sentences solely based on importance, balancing diversity with SRI boosts performance by a large margin.

For the DUC-04 and TAC-11 datasets, our proposed methods outperform unsupervised baselines by a large margin. It demonstrates that balancing the summary informativeness and diversity during the sentence extraction process is crucial for better summary quality.

System	*DUC-04*			*TAC-11*			*MultiNews*			*WikiSum*		
	R-1	**R-2**	**R-SU**	**R-1**	**R-2**	**R-SU**	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L***
Unsupervised Systems												
LEAD	30.77	8.27	7.35	32.88	7.84	11.46	39.41	11.77	14.51	37.63	14.75	33.76
MMR (1998)	30.14	4.55	8.16	31.43	6.14	11.16	38.77	11.98	12.91	31.22	10.24	22.48
LexRank (2004)	34.44	7.11	11.19	33.10	7.50	11.13	38.27	12.70	13.20	36.12	11.67	22.52
TextRank (2004)	33.16	6.13	10.16	33.24	7.62	11.27	38.44	13.10	13.50	23.66	7.79	21.23
SumBasic (2007)	29.48	4.25	8.64	31.58	6.06	10.06	-	-	-	-	-	-
KLSumm (2009)	31.04	6.03	10.23	31.23	7.07	10.56	-	-	-	-	-	-
PRIMERA (2022)	35.10	7.20	17.90	-	-	-	42.00	13.60	20.80	28.00	8.00	18.00
Individual. Greedy	34.81	7.85	11.37	34.42	8.10	11.25	40.48	13.49	16.14	37.24	10.29	32.77
Our Methods												
SRI+beam	**36.84**	**8.37**	12.28	**35.37**	**8.49**	**11.73**	**44.22**	**14.63**	18.61	38.94	15.23	34.12
SRI+exh	36.70	**8.37**	**12.31**	35.19	8.31	11.34	43.16	14.58	18.00	**39.26**	**16.15**	**34.19**

TABLE 4.2. ROUGE-F1 scores on four datasets under the unsupervised setting. Best unsupervised results are bold. For a fair comparison, we report R-L on Multinews and R-Lsum [**99**] for WikiSum and limit summaries to 100 words on DUC-04 and TAC-11. R-L are marked with * if reporting ROUGE-Lsum numbers.

Note that the input length of DUC/TAC datasets is extremely long, spanning an average of 180 sentences. These long inputs easily exceed the input capacity of transformer-based models, possibly resulting in information loss from documents. The proposed methods, on the other hand, process documents regardless of the input length or formats (SDS or MDS). Also, our unsupervised methods have the advantage of processing datasets with small training data. The superior performances on datasets with different input lengths and low-resource data illustrate the effectiveness of our methods. To further verify the model performance, we also conduct a human evaluation by experts on a scale of 5. The results shown in Table 4.3 also prove that our method outputs better summaries in an unsupervised setting.

Method	**Fluent**	**Informative**	**Faithful**	**Overall**
MMR	3.2	3.5	**4.7**	3.2
PRIMERA	**4.3**	2.5	3.3	3.3
SRI	3.8	**4.3**	**4.7**	**4.0**

TABLE 4.3. Human evaluation results on a scale of 1-5.

System	*DUC-04*			*MultiNews*		
	R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Adaptive Systems						
BART(2019)	24.1	4.0	15.3	27.3	6.2	15.1
BART (CNNDM)	29.4	6.1	16.2	36.7	8.3	17.2
PEGASUS (2020)	32.7	7.4	17.6	32.0	10.1	16.7
PEGASUS(CNNDM)	34.2	7.5	17.4	35.1	11.9	18.2
LED(2020)	16.6	3.0	12.0	17.3	3.7	10.4
PRIMERA (2022)	35.1	7.2	17.9	42.0	13.6	20.8
Our Systems						
SRI+beam (graph)	36.8	8.4	16.4	44.2	**14.6**	18.6
SRI+beam (CNNDM)	**36.9**	**8.6**	**18.5**	**44.6**	14.3	**21.1**

TABLE 4.4. ROUGE-F1 results on DUC-04 and Multinews datasets under the adaptive setting. Models adapted from CNN/DailyMail dataset are marked in the bracket.

4.4.3. Adptive Summarization Results.

The experimental results under the adaptive setting are shown in Table 4.4. Compared to large pre-trained generation models (BART) and other task-specific pre-trained summarization models (PEGASUS, PRIMERA), our framework shows strong performance when adapting from a single document summarization dataset. We also notice that fine-tuning on a single document summarization corpus improves the performance of all pre-trained models, but still, our framework achieves the best results under the adaptive setting.

4.5. Analysis

4.5.1. Summary Diversity.

Other than summary quality, we also test the effectiveness of our Subset Representative Index (SRI) in terms of the diversity of the output summaries. We present the unique n-gram ratios of output summaries under unsupervised and adaptive settings and the reference summary on the TAC-11 dataset in Figure 4.3. According to the results, our framework is

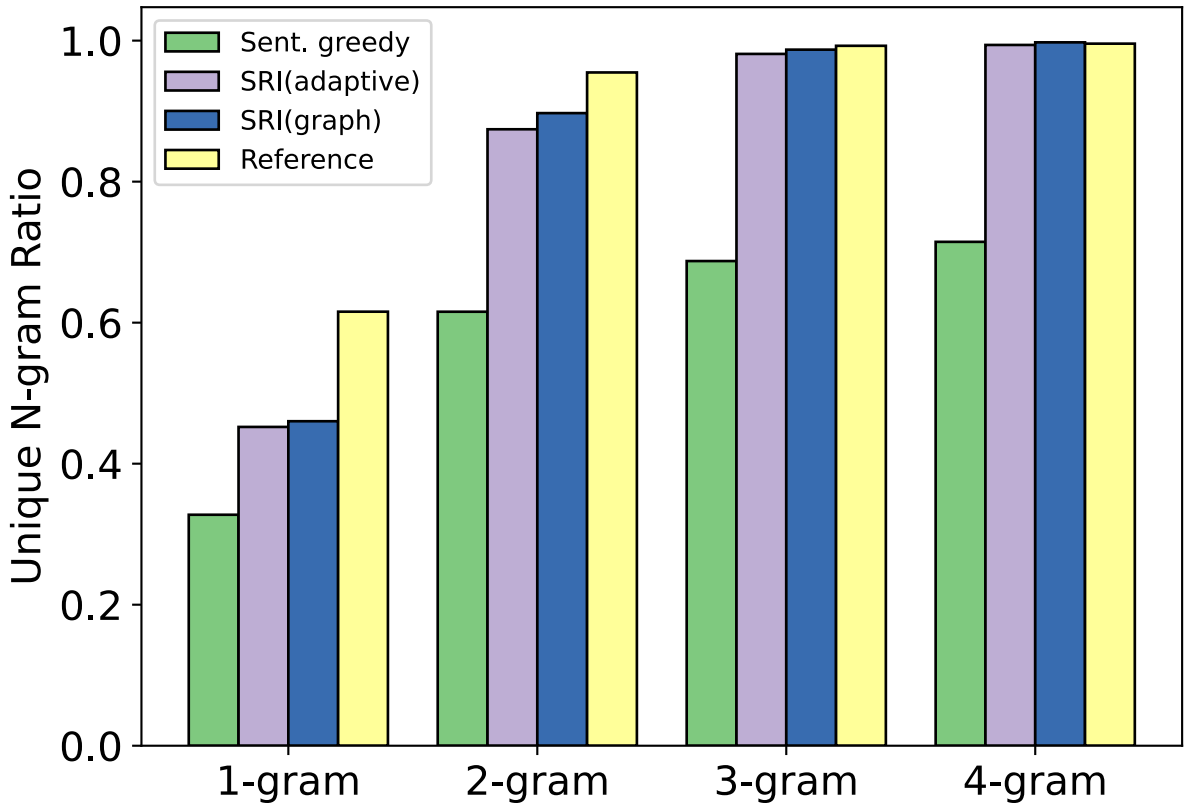

FIGURE 4.3. Unique n-gram ratios ($n = 1, 2, 3, 4$) of the output summary by different methods on TAC-11.

extremely effective in reducing summary redundancy and increasing summary diversity under both unsupervised and adaptive settings.

Compared to the ROUGE-F1 results, holistic inference with importance-diversity balancing measurement SRI increases both summary quality and diversity at the same time. The results suggest that considering summary diversity is beneficial in extractive summarization, especially in redundant cases like MDS and long document summarization. Our findings also verify the crucial role of effective modeling of sentence importance and similarity.

4.5.2. Hyperparameter Sensitivity.

To test the robustness of our proposed approaches, we study the hyperparameter sensitivity of our proposed methods. The results are shown in Figure 4.4. The first plot shows the impact of the balancing factor λ in SRI. The second plot shows the impact of α, which

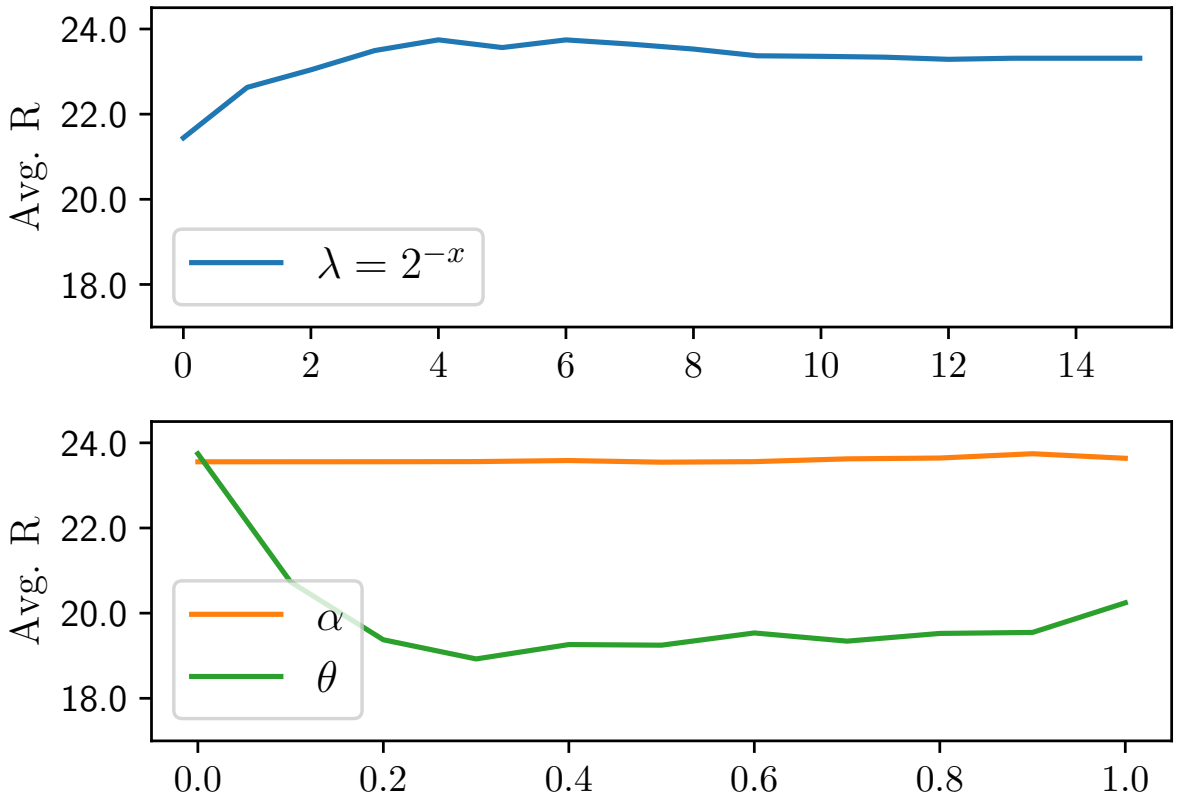

FIGURE 4.4. Average ROUGE-F1 (w/o word limit) results with different hyperparameter values on TAC-11.

balances the contextualized and TF-IDF sentence embeddings, and the edge weight threshold. The results show that our methods are relatively stable towards the hyperparameter values and could be easily adapted to unseen datasets.

4.5.3. Inference Approaches Analysis.

We also compare the efficiency and effectiveness of different inference methods. As shown in Figure 4.5, we compare sentence-level greedy search, set-level greedy search, set-level beam search (beam size = 4), and set-level exhaustive search with pre-filtering as inference methods for both unsupervised and adaptive settings. We pick the filter size of 20 here since the search space without filtering $C(N, K)$ is extremely large. According to the results, all set-level inference methods outperform the sentence-level methods. This suggests that extracting summaries at a set level (holistic) is optimal over the common sentence-level

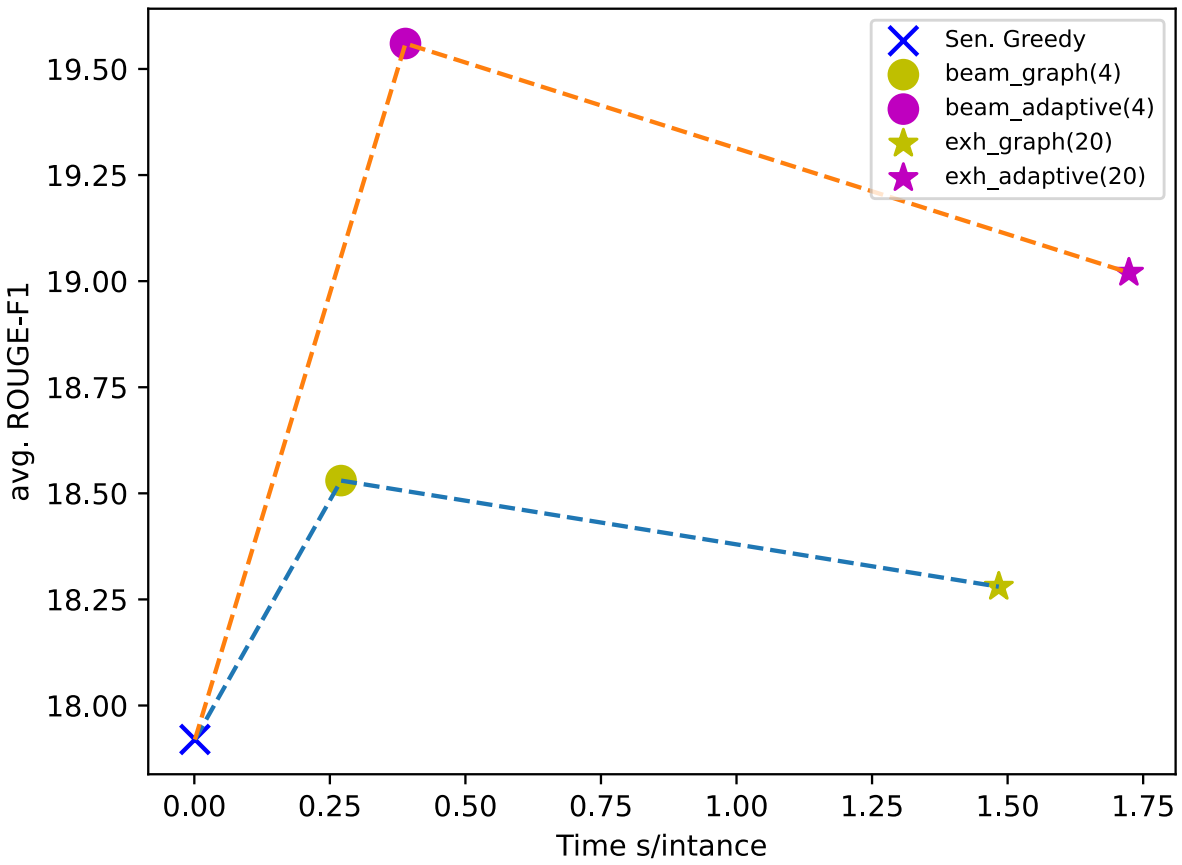

FIGURE 4.5. Efficiency vs. average ROUGE (w/o word limit) scores of different inference methods on TAC-11.

setting that extracts sentences individually. The finding is also consistent with the inherent performance gap between sentence-level and holistic extractors in [**136**].

Moreover, we realize that set-level beam search and set-level exhaustive search achieve comparable best performance. However, set-level beam search is much more efficient speed-wise than set-level exhaustive search. We also show the effect of different beam sizes in Table 4.5. The results indicate that a reasonably small beam size achieves the best ROUGE results, which are both effective and efficient. To conclude, set-level beam search with SRI shows the best overall performance.

Beam Size	**2**	**3**	**4**	**5**	**6**	**7**	**8**
ROUGE-1	33.43	33.65	33.62	**33.76**	34.72	33.64	33.67
ROUGE-2	7.71	**8.00**	7.87	7.93	7.84	7.86	7.85
ROUGE-L*	28.74	29.03	28.99	**29.10**	29.01	28.94	29.01

TABLE 4.5. ROUGE-F1 (w/o word limit) results of SRI-beam with different beam sizes on TAC with $\lambda = 0.125$.

4.6. Conclusion

This chapter introduces a holistic framework for unsupervised multi-document extractive summarization. Our framework incorporates holistic beam search inference methods and SRI, a holistically balanced measurement between importance and diversity. We conduct extensive experiments on both small and large-scale MDS datasets under both unsupervised and adaptive settings, and the proposed method outperforms strong baselines by a large margin. We also find that balancing summary set importance and diversity benefits both the quality and diversity of output summaries for MDS.

CHAPTER 5

Improving Summary Generation with Iterative Refinement

5.1. Introduction

Document summarization, aiming to condense text while preserving its key information, has become increasingly important with the proliferation of publicly available textual data. Recent advancements in summarization systems, leveraging neural networks and pre-trained language models, have shown significant progress [**13**, **54**, **66**, **78**, **131**]. However, these summarization systems typically follow an end-to-end approach, generating summaries in a single step. In contrast, human summarization often involves an iterative process of drafting and editing [**28**].

These end-to-end summarization systems face several challenges. Firstly, they frequently encounter the problem of hallucination, leading to the generation of ungrammatical or factually incorrect content [**51**]. Secondly, these systems are often optimized using imperfect reference summaries, and widely adopted evaluation metrics like ROUGE [**60**] may not accurately assess summary quality. Thirdly, most of these systems lack controllability, as they only produce a single generic summary conditionally on the input document. In practice, generating summaries that cater to specific aspects or queries would be more beneficial to meet the diverse requirements of users, rather than providing a single condensed version of the entire document.

The advent of advanced instruction-tuned large language models (LLMs), such as ChatGPT, has opened up exciting possibilities for summarization systems by demonstrating strong zero-shot performance in various downstream tasks. A recent study by Goyal et al. compared GPT-3 with traditional fine-tuning methods and found that despite lower ROUGE

scores, human annotators preferred the GPT-3 generated summaries. Another comprehensive analysis by Zhang et al. focused on large language models for news summarization and revealed that the quality of generated summaries is already on par with those created by humans. Furthermore, Liu et al. demonstrated the utilization of LLMs like GPT-4 as an effective natural language generation evaluator, showing a higher correlation with humans in the summarization task compared to previous reference-based methods.

The emergence of LLMs also presents new opportunities for summarization beyond the traditional one-shot generation setting. In this paper, we introduce SummIt, a framework that leverages large language models for iterative text summarization. Instead of generating summaries in a single step, our framework enables the model to iteratively refine the generated summary through self-evaluation and feedback, resembling the human process of drafting and revising summaries, as shown in Figure 5.1. According to our experiments, the rationale generation and summary refinement in SummIt can be effectively guided with in-context learning, eliminating the need for supervised training or reinforcement learning processes. Additionally, we explore the potential benefits of incorporating knowledge and topic extractors to enhance summary faithfulness and controllability. We instantiate SummIt with ChatGPT as the backbone, and the automatic evaluation results on three benchmark datasets demonstrate the effectiveness of SummIt in improving summary quality, faithfulness, and controllability within only a few iterations. Furthermore, we conduct a human evaluation to validate the iterative refinement quality and identify potential over-correction issues.

We summarize the contributions of this chapter as follows:

- We introduce SummIt, a novel framework for iterative text summarization, enabling the iterative refinement of generated summaries through self-evaluation and feedback mechanisms. Additionally, we propose the integration of knowledge and topic extractors to enhance the faithfulness and controllability of SummIt.

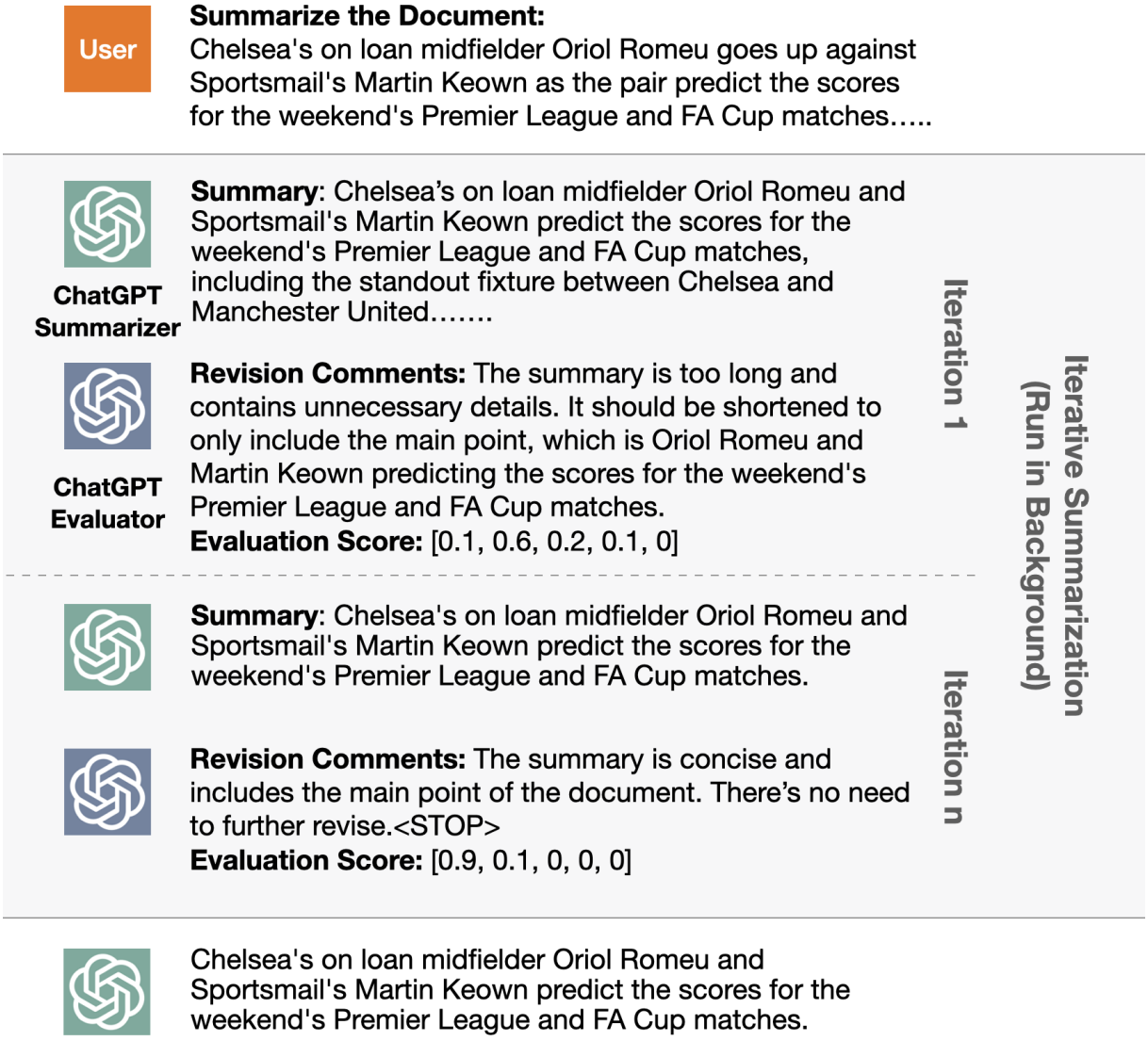

FIGURE 5.1. An illustration of the iterative summarization process. The summarizer continuously refines the summary according to self-feedback from the evaluator at each iteration.

- We conduct experiments on three benchmark summarization datasets, and automatic evaluation results demonstrate the effectiveness of our proposed framework in summary refinement.
- A human evaluation is conducted to examine the impact of self-evaluation-guided summary refinement. The results uncover a potential issue of over-correction, where

the large language model may prioritize its own evaluation criteria over closely aligning with human judgment.

5.2. Related Work

5.2.1. Text Summarization.

Recent years have seen significant advancements in text summarization systems with the development of deep neural networks and pre-trained language models. Automatic summarization methods can be broadly categorized into extractive and abstractive approaches. Extractive summarization involves directly extracting sentences from the source text to form summaries [**66**, **115**, **125**, **136**], while abstractive approaches conditionally generate summaries using a sequence-to-sequence (seq2seq) framework [**54**, **131**].

Existing approaches mentioned above generate summaries in a one-shot manner, and their outputs may not always align with user expectations and may contain hallucinated content [**51**]. To address this limitation, [**68**] proposes automatically correcting factual inconsistencies in generated summaries with generated human feedback. In contrast, our SummIt framework enables iterative summary refinement with self-evaluation and feedback, eliminating the need for costly human annotations. Additionally, we propose integrating knowledge and topic extractors to further enhance summary faithfulness and controllability.

5.2.2. Summarization with Large Language Models.

Recent years have seen a surge in training LLMs on huge amounts of text, such as GPT [**11**, **88**]. Several studies have explored the application of LLMs in the context of text summarization. For instance, [**34**] compared the performance of GPT-3-generated summaries with traditional fine-tuning methods, finding that although the former achieved slightly lower ROUGE scores, human evaluators expressed a preference for them. Similarly, [**133**] reported that LLM-generated summaries were on par with human-written summaries in the news domain. [**126**] benchmarked the performance of ChatGPT on extractive summarization and proposed to improve summary faithfulness with an extract-then-generate

pipeline. On the other hand, prior works have also leveraged LLMs for summarization evaluation [**29**,**64**,**71**], demonstrating that LLM-based metrics outperform all previous evaluation metrics like ROUGE [**60**] and BertScore [**132**] by a significant margin in terms of correlation with human evaluations.

5.2.3. Text Editing.

Our work is also closely related to the task of text editing. Traditional editing models are trained to solve specific tasks, such as information updating [**43**], Wikipedia edit [**91**], and grammar error correction [**5**]. Recent works also formulate text editing as an interactive task, such as command-based editing systems [**28**], and interactive editing systems [**97**]. [**122**] also proposed a benchmark for fine-grained instruction-based editing.

Recently, [**111**] introduced a self-corrective learning framework that incorporates a corrector into the language model to facilitate self-correction during sequence generation. [**1**] propose a reinforcement learning-based approach to generate natural language feedback for correcting generation errors. Concurrent work [**73**] presents a similar generation pipeline that enhances initial outputs through iterative feedback using a single LLM for short text generation tasks. In contrast, our SummIt framework differs from these approaches as it specifically focuses on the conditional generation task of summarization, with an emphasis on improving summary faithfulness and controllability. Additionally, we empirically observe that separating the summarizer and evaluator into different LLMs, each employing different in-context guidance leads to improved performance in our framework.

5.3. Methods

5.3.1. Iterative Summarization.

The overall architecture of our iterative text summarization system SummIt is shown in Figure 5.2. The system consists of two major components: a summarizer that generates and refines the summary, and an evaluator that generates feedback rationale.

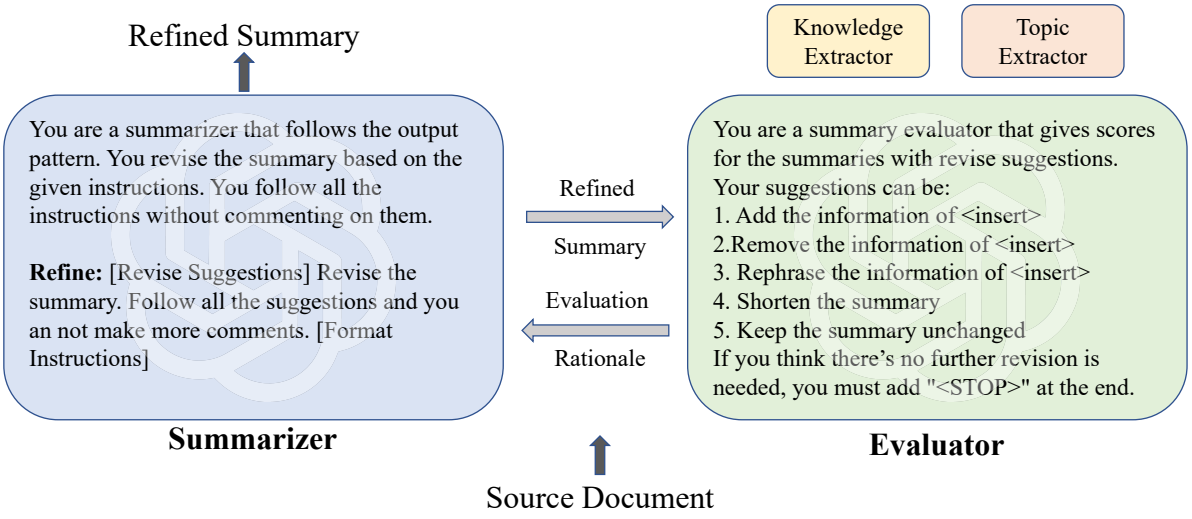

FIGURE 5.2. The overall framework of our proposed iterative text summarization system. The evaluator generates an evaluation rationale based on the current summary, and the summarizer refines the summary accordingly. The knowledge and topic extractors retrieve information from the source document to guide the process.

Summarizer: The summarizer is responsible for generating the initial summary and revising it based on the given explanations and source document. We instantiate the summarizer with an instruction-tuned language model S.

Formally, given the input source document $\mathbf{x}$, the initial summary $\mathbf{y}^0$ generation process can be represented as:

$$p_S(\mathbf{y}^0 \mid \mathbf{x}) = \prod_{t=1}^{m} p_S\left(y_t^0 \mid \mathbf{y}_{<t}^0, \mathbf{x}\right), \tag{5.1}$$

, where $\mathbf{y}_{<t}^0$ denotes the generated tokens, y_t^0 refers to the t-th summary token, and m denotes the summary length.

After obtaining the i-step self-evaluation feedback $\mathbf{e}^i$ from the evaluator E, the summarizer will refine the summary accordingly and then generates refined summary $\mathbf{y}^{(i+1)}$ as: $p_S(\mathbf{y}^{(i+1)} \mid \mathbf{x}, \mathbf{e}^i)$.

Evaluator: The evaluator is another instance of language model E that generates summary quality evaluation and corresponding explanations $\mathbf{e}^i$ for the i-th iteration as: $p_E(\mathbf{e}^i \mid \mathbf{x}, \mathbf{y}^i)$.

Stopping Criteria: The evaluator provides a quality assessment of the generated summary and then outputs the rationale for the evaluation as feedback. The summarizer receives the model evaluation and feedback from the evaluator, subsequently refining the summary based on this input.

This iterative process can be repeated until either

- the evaluator determines that no further refinement is required or
- certain rule-based stopping criteria are met, such as reaching a maximum iteration number.

5.3.2. In-context Learning.

Since the summarizer and evaluator in SummIt are not fine-tuned with supervised data or trained reinforcement learning rewards, it would be beneficial to guide the explanation and summary generation process with the desired format or template. Recent studies have shown that large language models have strong few-shot performance on various downstream tasks, known as in-context learning (ICL) [**11**]. This means that by providing a few examples or prompts in the desired format, the model can adapt its behavior to follow the specified pattern. Therefore, we can leverage in-context learning to guide the generation process in SummIt by providing prompts or examples that indicate the desired structure or content of the summary and feedback rationale.

The standard ICL prompts a language model, M, with a set of exemplar source-summary pairs, $\mathbf{C} = \{(x_1, y_1)...(x_m, y_m)\}$, and generates summary $\mathbf{y}$ by concatenating the exemplar source-summary pairs and input document as prompt: $p_M(\mathbf{y} \mid \mathbf{x}, \mathbf{C})$.

We also utilize in-context learning to guide our iterative summarization system. Specifically, we provide "document-reference summary" pairs as the context for the summarizer S and "document-reference summary-human written explanation" triplets as the context for the evaluator E. Through this approach, we aim to leverage the model's ability to adapt its behavior based on the provided examples or prompts, thereby enhancing the effectiveness of our system. Empirical results demonstrate that in-context learning contributes to the improved performance of our framework.

5.3.3. Summary Faithfulness and Controllability.

In real-world applications, ensuring the faithfulness of generated summaries is crucial, in addition to their overall quality [**51**]. Previous studies have shown that incorporating knowledge extraction from source documents can improve the faithfulness of generated summaries [**42**, **139**]. Building on these findings, we suggest integrating a knowledge extractor into our iterative summarization system.

Knowledge Extractor: In particular, we utilize OpenIE [1], which extracts knowledge $\mathbf{k}$ in the form of triplets from the source document. During each iteration, the summarizer (S) is guided to refine the summary in accordance with the extracted knowledge, represented as: $p_S(\mathbf{y}^{(i+1)} \mid \mathbf{x}, \mathbf{e}^i, \mathbf{k})$. Moreover, the evaluator (E) can be directed to factor in faithfulness when delivering feedback, denoted as $p_E(\mathbf{e}^i \mid \mathbf{x}, \mathbf{y}^i, \mathbf{k})$, as LLMs have shown to be efficient faithfulness evaluators [**71**].

Moreover, in real-world applications, there is often a need to generate summaries tailored to specific aspects or queries, rather than producing a single generic summary of the entire document. Our iterative summarization framework offers enhanced controllability for aspect-based summarization tasks, allowing users to specify particular aspects or queries to focus on during the summarization process.

Topic Extractor: Incorporating aspect-oriented queries into our iterative summarization framework involves prompting both the summarizer S and evaluator E to initially extract

[1] https://stanfordnlp.github.io/CoreNLP/openie.html

relevant snippets, each containing less than 5 words, from the source document $\mathbf{x}$. After the extraction, these components proceed to either generate or assess the summary, considering the extracted snippets. The iterative nature of our framework further facilitates controllable summary generation, enabling the transformation of generic summaries into topic-focused summaries based on the user's preferences.

Model	System Prompt
Summarizer	You are a summarizer that follows the output pattern. You revise the summary based on the given instructions. You follow all the instructions without commenting on them. Make sure the summary is concise and accurate.
Evaluator	You are a summary evaluator that follows the output pattern. You give scores for the summaries as well as revise suggestions. Your score should correspond to your suggestions. Your suggestions can be: 1. Add the information of [] 2. Remove the information of [] 3. Rephrase the information of [] 4. Shorten the summary. 5. Do nothing. Only ask for the information that appeared in the document. If you find the summary is too long, ask for a shorter summary. Keep the summary concise. If you think there's no further revision is needed, you must add $< STOP >$ at the end of your output at the end of the comment. Give precise and clear suggestions.

TABLE 5.1. System prompts of the summarizer and the evaluator for all settings.

5.3.4. Prompt Format.

We utilize both system prompts and user prompts following the OpenAI API in our system implementations. The full prompts used in the experiments can be found in Table 5.1 and Table 5.2. Notably, we empirically find that pre-defining the possible edit operations for the evaluator improves the system performance significantly since it avoids free-form edits to the summary by the large language model. Thus, we adopt the five types of text editing operations commonly used in text editing systems [**28**, **91**]. We specifically require the

Setting	Model	User Prompt
Quality	**Summarizer**	**Summarize**: [*In-context Examples*] Please summarize the following document. [*Document Content*] [*Format Instructions*] **Refine**: [*Revise Suggestions*] Revise the summary. Follow all the suggestions and you can not make more comments. [*Format Instructions*]
	Evaluator	**Evaluate**: [*In-context Examples*]Please evaluate the summary for the document.[*Document Content*] [*Summary Content*].The output should be a probability distribution of assigning the score between 1-5 as well as its justification. Please give revise comments if you think this summary is not good enough.[*Format Instructions*]
Control	**Summarizer**	**Summarize**: [*In-context Examples*] Please summarize the following document based on the given topic sentence. [*Document Content*] [*Topic Sentence*] [*Format Instructions*] **Refine**: [*Revise Suggestions*] Revise the summary. Follow all the suggestions and you can not make more comments. [*Format Instructions*]
	Evaluator	**Evaluate**: [*In-context Examples*] Please evaluate the summary for the document to check if the summary follows the given topic sentence.[*Document Content*] [*Summary Content*] [*Topic Sentence*].The output should be a probability distribution of assigning the score between 1-5 as well as its justification. Please give revise comments if you think this summary is not good enough.[*Format Instructions*]
Faithfulness	**Summarizer**	**Summarize**: [*In-context Examples*] Please summarize the following document based on the given relationships. [*Document Content*] [*OpenIE Relationships*] [*Format Instructions*] **Refine**: [*Revise Suggestions*] Revise the summary. Follow all the suggestions and you can not make more comments. [*Format Instructions*]
	Evaluator	**Evaluate**: [*In-context Examples*] Please evaluate the summary for the document to check if the summary follows the given relationships.[*Document Content*] [*Summary Content*] [*OpenIE Relationships*].The output should be a probability distribution of assigning the score between 1-5 as well as its justification. Please give revise comments if you think this summary is not good enough.[*Format Instructions*]

TABLE 5.2. User prompts of summarizer and evaluator for different settings.

evaluator to generate feedback based on the source document and summary at this iteration with the following five types of possible refinement operations:

- **Add**: Add the information of $< insert >$
- **Remove**: Remove the information of $< insert >$ from the summary
- **Rephrase**: Rephrase the information of $< insert >$ in the summary
- **Simplify**: Shorten the summary
- **Keep**: Keep the summary unchanged

5.4. Experiments

In this section, we validate our SummIt framework on three benchmark summarization datasets. We employ both automatic metrics and human assessment to evaluate the quality (Section 5.4.2), faithfulness (Section 5.4.3), and controllability (Section 5.4.4) of the generated summaries.

Dataset	#Test	Doc #words	Sum #words	#Sum
XSum	11,334	430.2	23.3	1
CNN/DM	11,489	766.1	58.2	1
NEWTS	600	738.5	70.1	2

TABLE 5.3. Detailed statistics of the experimental datasets. Doc # words and Sum # words refer to the average word number in the source document and summary. # Sum refers to the number of output summaries per document.

5.4.1. Experiment Settings.

Datasets: We conduct experiments on the following three publicly available benchmark datasets, as presented in Table 5.3, ensuring they are consistent with previous fine-tuning approaches:

- *CNN/DailyMail* [**37**] is the most widely adopted summarization dataset that contains news articles and corresponding human-written news highlights as summaries. We use the non-anonymized version in all experiments.

- *XSum* [**79**] is a one-sentence news summarization dataset with all summaries professionally written by the original authors of the BBC news.
- *NEWTS* [**7**] is an aspect-focused summarization dataset derived from the CNN/DM dataset. It contains two summaries focusing on different topics for the same news.

Evaluation metrics: For assessing summary quality, we utilize ROUGE scores [**60**] and G-Eval [**64**] as the automatic metrics. The ROUGE scores include ROUGE-1, ROUGE-2, and ROUGE-L, measuring the overlap of unigrams, bigrams, and the longest common sequence between the generated summary and the reference summary, respectively. On the other hand, G-Eval is an LLM-based metric that provides scores on a scale ranging from 1 to 5. It employs an LLM with chain-of-thoughts (CoT) and a form-filling paradigm to evaluate the quality of natural language generation (NLG) outputs. G-Eval has shown the highest correlation with human judgments compared to other summarization quality metrics.

To assess summary faithfulness, we employ FactCC [**51**] and DAE (Defining arc entailment) [**33**] as our evaluation metrics. FactCC is a weakly supervised BERT-based metric designed to verify factual consistency by applying rule-based transformations to sentences from the source document. It demonstrates a high correlation with human judgments in evaluating summary faithfulness. On the other hand, DAE decomposes entailment at the level of dependency arcs, analyzing the semantic relationships within the generated output relative to the input. Unlike aggregate decisions, DAE evaluates the semantic relationship manifested by individual dependency arcs in the generated output supported by the input.

To evaluate the controllability of query-focused summarization, we utilize BM25 [**93**] and DPR [**47**] to measure the similarity between the query and the summary, incorporating both sparse and dense evaluations. BM25 is a probabilistic retrieval function commonly used in information retrieval tasks. It ranks documents based on the frequency of query terms, providing a measure of relevance between the query and the summary. DPR, on the other hand, leverages dense vector representations for scalable retrieval. It embeds both questions

and passages into fixed-length vector spaces, allowing for nuanced similarity calculations between the query and the summary.

In line with previous research findings that have emphasized the inclination of human annotators towards summaries generated by LLM models, even in the presence of comparatively lower ROUGE scores [**34**], we further validate the effectiveness of SummIt through a dedicated human study. Specifically, we use:

- Five-point Likert scale ratings [**59**] to assess various aspects of summary quality, including coherence, fluency, relevance, consistency, conciseness, and overall evaluation.
- Human preference test: annotators are shown summaries of the same source document from all five summarization systems and then asked to select their most preferred summary or summaries.

We evaluate the performance using 1000 random samples from CNN/DM and XSum test sets, with seed 101, and the full NEWTS test set. Our prompts were refined with a 50-example development set.

For the baseline models, we utilize the official checkpoints of BART, T5, and PEGASUS from Huggingface. As for the backbone LLM for both generating and evaluating summaries in SummIt, we employ the *gpt-3.5-turbo* model. To maintain reproducibility, we set the temperature parameter to 0.

To ensure consistency and reliability in our evaluations, we follow a rigorous procedure for dataset sampling and model tuning. Specifically, we randomly sample 1000 samples from the test set of both the CNN/DM and XSum datasets using a random seed of 101. For the NEWTS dataset, we utilize the entire test set. Additionally, we fine-tune the optimal prompt and hyperparameters of the LLM on a development set comprising 50 examples. To address any potential variability, each experimental run is conducted three times, and the average results are reported to mitigate the instability inherent in small datasets.

5.4.2. Generic Summary Quality Evaluation.

Model	CNN/DM				XSum			
	R1	**R2**	**RL**	**G-Eval**	**R1**	**R2**	**RL**	**G-Eval**
Zero-shot setting								
PEGASUS_{ZS}	32.90	13.28	29.38	3.23	19.27	3.00	12.72	3.52
BART_{ZS}	32.83	13.30	29.64	3.42	19.26	3.30	14.67	3.49
T5_{ZS}	**39.68**	**17.24**	26.28	3.47	19.66	2.91	15.31	3.55
ChatGPT	39.44	16.14	**29.83**	3.46	21.61	**5.98**	17.60	3.47
SummIt (ours)	36.50	13.49	26.76	**4.33**	**21.92**	5.93	**17.62**	**4.24**
Few-shot setting								
ChatGPT	**40.00**	**16.39**	**30.02**	3.57	**23.96**	**7.36**	**19.36**	3.57
SummIt (ours)	37.29	13.60	26.87	**4.35**	22.04	6.20	17.46	**4.32**

TABLE 5.4. Automatic evaluation results on the CNN/DM and XSum datasets under both zero-shot and few-shot settings. A random sample of 1,000 data points was taken from each dataset for evaluation. G-Eval represents the score evaluated by the ChatGPT evaluator in our framework.

Automatic Evaluation: The automatic evaluation results for generic summarization quality are shown in Table 5.4. We used previous pre-trained language models, including PEGASUS [**131**], BART [**54**], and T5 [**89**], as baseline models. We compared our framework SummIt with these baseline models under a zero-shot setting for a fair comparison.

It is observed that SummIt has inferior ROUGE scores compared to fine-tuning approaches on CNN/DM, while exhibiting significantly higher LLM-based evaluation metric G-Eval. On the other hand, it outperforms all baseline methods on the XSum dataset. Compared to the output of ChatGPT, the summaries of ChatGPT after our iterative refinement show a consistent improvement in the G-Eval score. These results are consistent with the previous conclusions in [**133**], where large language model summary outputs receive lower ROUGE scores due to the low quality of reference summaries.

In addition to the zero-shot setting, we investigate the effects of in-context learning for SummIt, as shown in the lower block of Table 5.4. The results consistently demonstrate that incorporating in-context learning significantly enhances the model's performance on ROUGE and G-Eval scores. This observation underscores the substantial few-shot capabilities of

SummIt, showcasing its ability to adapt effectively and generate high-quality summaries in contexts with very few examples.

Model	Coherence	Fluency	Relevance	Consistency	Conciseness	Overall	Human Pref
				CNN/DM			
BART	3.92	4.16	4.00	3.12	3.64	3.24	0.04
T5	3.72	4.24	**4.32**	3.52	3.84	3.68	0.10
PEGASUS	3.20	3.53	3.33	2.87	1.85	1.63	0.00
ChatGPT	4.20	4.36	4.28	4.01	**3.92**	4.01	0.34
SummIt	**4.24**	**4.50**	4.29	**4.12**	3.84	**4.09**	**0.52**
				XSum			
BART	3.97	4.30	4.13	3.30	**3.93**	3.84	0.30
T5	3.84	4.32	4.02	3.63	3.84	3.25	0.08
PEGASUS	3.13	4.10	3.52	2.87	2.03	2.41	0.00
ChatGPT	4.03	**4.40**	**4.30**	3.93	3.87	3.92	0.24
SummIt	**4.04**	4.35	4.28	**4.05**	3.72	**3.96**	**0.38**

TABLE 5.5. Human study results on generic summary quality. The first five columns include Likert scale ratings and the last column is the human preference results.

Human Evaluation: To further verify the summary quality, we conducted a human study to evaluate the overall quality of the summaries as shown in Table 5.5. According to the five-point Likert scale ratings, the summaries of ChatGPT and SummIt consistently outperformed pre-trained language model results. The iterative refinement of SummIt also provided consistent improvements, which align with the G-Eval results obtained from the automatic evaluation. We also conducted a human preference study, where summaries from all models were presented to human annotators. They were tasked to select the best summary, without any prior knowledge of the origin of each summary. Consistent with the findings in [**34**], the results revealed a clear preference among human annotators for summaries generated by large language models (LLMs) for both CNN (86%) and BBC (62%) style summaries. We also noticed that the summaries of ChatGPT after our iterative refinement (SummIt) showed a significant improvement in human preference, with 18% and 14% percent improvements on CNN/DM and XSum datasets, respectively. These results demonstrate the effectiveness of refining generic summaries of our framework.

5.4.3. Summary Faithfulness Evaluation.

To evaluate the efficacy of the SummIt framework in enhancing summary faithfulness with the knowledge extractor, we conducted additional experiments, as presented in Table 5.6. The findings demonstrate that our framework's iterative approach to refining summaries yields significant improvements in summary faithfulness, as indicated by both FactCC and DAE results. Furthermore, the integration of a knowledge extractor such as OpenIE further enhances the level of faithfulness. The LLM-based evaluation score G-Eval also indicates a higher level of satisfaction with the refined summaries when guided by the extracted knowledge triplets. In conclusion, our study reveals that iterative refinements with the incorporation of the knowledge extractor effectively enhance summary faithfulness without compromising the quality of the summaries.

	R1	**R2**	**RL**	**G-Eval**	**FactCC**	**DAE**
ChatGPT	21.61	**5.98**	17.60	3.47	28.00	10.34
SummIt	21.92	5.93	17.62	4.24	36.00	33.02
ChatGPT-IE	**22.01**	5.11	**17.06**	3.85	**51.68**	**93.68**
SummIt-IE	19.72	3.85	15.36	**4.95**	47.24	90.36

TABLE 5.6. Experimental results of incorporating knowledge extractor on summary quality and faithfulness on XSum dataset. -IE refers to the model integrated with OpenIE.

5.4.4. Query-focued Summarization Controlability Evaluation.

	R1	**R2**	**RL**	**G-Eval**	**BM25**	**DPR**
ChatGPT	30.01	8.94	27.03	1.06	33.09	77.22
ChatGPT-Topic	**33.24**	**10.20**	**29.88**	1.16	36.20	78.77
SummIt-Topic	30.45	8.48	27.19	**4.74**	**39.11**	**82.41**

TABLE 5.7. Experimental results on NEWTS dataset to test the controllability of our framework. -Topic indicates a model that is prompted to extract topic-related snippets before generating a summary.

We utilized the query-based summarization dataset NEWTS as our testbed to demonstrate the controllability ability of SummIt. The results obtained, as depicted in Table 5.7,

highlight the framework's capability to align the focus of a generic summary with the specific topic of interest or query provided by the user. We also observed improved G-Eval evaluation scores by directing the summary generation process toward the intended topic.

Furthermore, we evaluated the controllability of the summarization systems by quantifying the similarity between the query and the generated summary. Both BM25 and DPR were employed as similarity metrics, and we consistently observed enhancements after the iterative refinement process. This observation serves as evidence that SummIt effectively refines the summary to align with the topics specified in the query.

5.5. Analysis

5.5.1. Ablation Studies.

Table 5.8 presents the results of the ablation study by removing refinement operations. The ablation study is conducted on the CNN/DM dataset under the zero-shot settings. According to the results, each option contributes to the success of our method, and the add operation affects the ROUGE score most, while the simplify operation affects the GPT-evaluation scores the most. Without the add operation, the information in the iterative process will only decrease, resulting in less n-gram overlap. On the other hand, without the simplify and remove operations, the redundant information results in low G-Eval scores.

	R1	**R2**	**RL**	**G-Eval**
SummIt	36.50	13.49	26.76	4.33
-w/o Add	33.01	11.55	24.71	3.98
-w/o Remove	36.46	13.44	26.55	3.64
-w/o Rephrase	34.71	12.12	26.31	3.82
-w/o Simplify	33.49	12.33	25.76	3.55
-w/o Keep	33.87	13.03	25.70	3.94

TABLE 5.8. Ablation Study on Iterative Refinement Operations

5.5.2. Over-correction Issue.

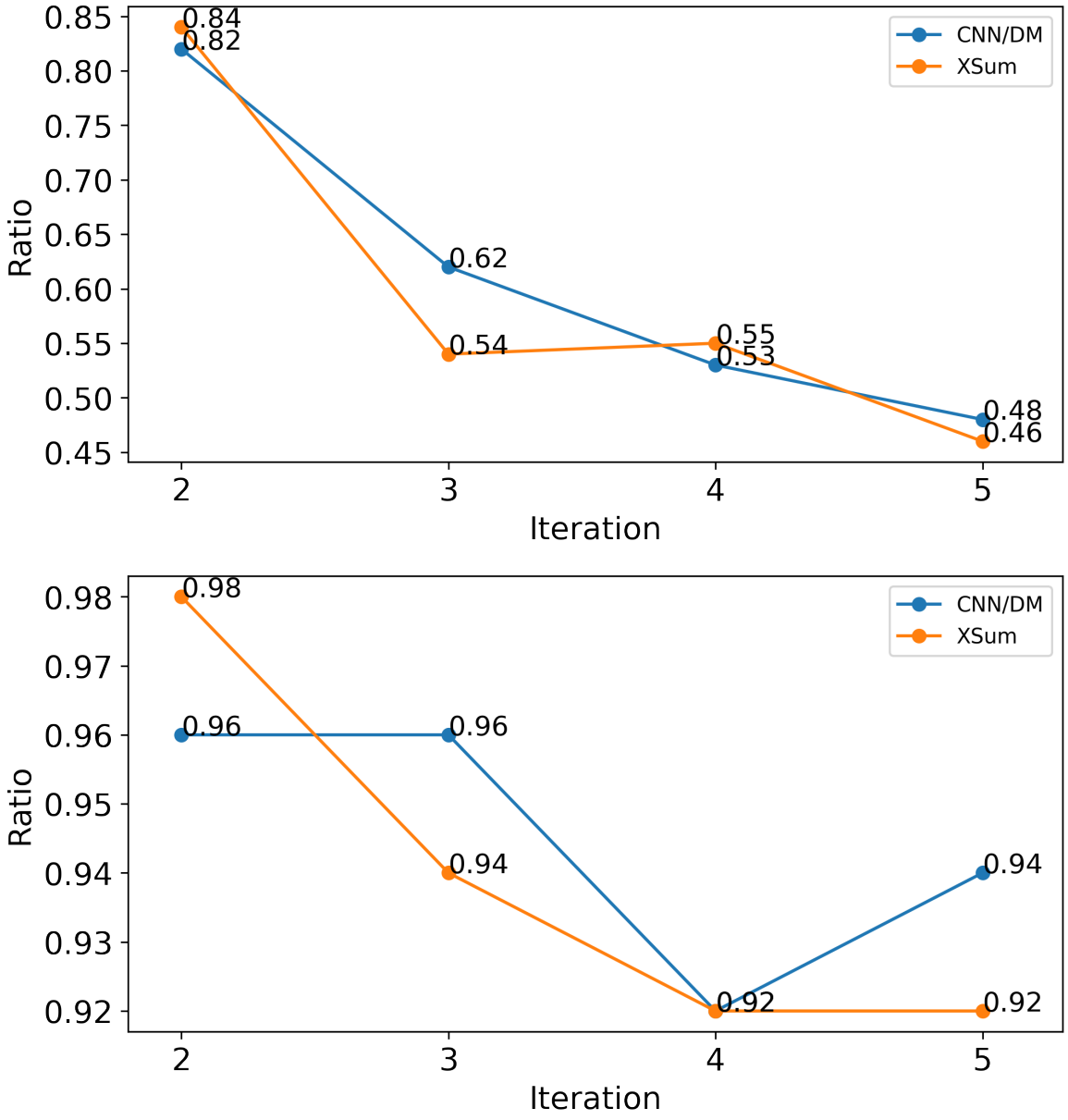

FIGURE 5.3. Human evaluation to justify the refinement behavior of SummIt. The top plot refers to the human justification of the ratio that the summary is improved at each iteration and the bottom plot indicates the ratio that the summarizer follows the evaluator's evaluation rationale.

A recent work [**64**] highlights the potential issue of LLM-based evaluators having a bias towards the LLM outputs, which raises the doubt:

(1) Does the refinement actually improve the summary?

(2) Does the refinement actually follow the rationale feedback from the evaluator?

To address these two concerns and provide further validation for the step-wise summary refinement in SummIt, we conducted the corresponding human evaluations. Specifically, we asked expert human annotators to label:

(1) Whether these edits resulted in improvements to the summary based on human judgment.

(2) Whether the edits made by the summarizer align with the feedback provided in the last step by the evaluator.

The results of the human evaluation, presented in Figure 5.3, indicate that approximately 90% of the edits performed by the summarizer adhered to the provided feedback as intended on both datasets. However, only around 50-60% of these edits after 2 or more iterations were deemed beneficial according to human judgment, whereas the evaluator in SummIt still asks for refinements to be performed. We also notice a clear trend that the percentage of beneficial refinements decreases as the iteration number increases. The finding shows an **Over-correction** problem: the LLM may demand itself to continuously refine the summary based on its own evaluation criteria, rather than adhering to the true evaluation criteria of good summaries by humans.

This finding highlights the need for better stopping criteria in the development of iterative summarization systems, and we argue that incorporating human-in-the-loop may be a potential solution. We leave this for future work.

5.5.3. Case Study.

We present an example of iterative summary refinement in Table 5.9. The evaluator provides a detailed rationale for the summary in the first iteration, suggesting revisions to the blue and orange sentences by removing information and rephrasing sentences. The summarizer can then refine the summary accordingly.

Summary-Iter1: Hospitals in Wales may have to choose between emergency care and non-urgent surgery during peak winter months, according to Dr. Andrew Goodall. He suggested that hospitals may decide not to schedule surgery in order to focus on "front door pressures." Two hospitals in Swansea and Bridgend have already cancelled some surgical procedures until after Christmas.
Evaluation-Iter1: The summary effectively conveys the main point of the article, but it could be shortened for conciseness. Consider removing the specific hospitals mentioned and rephrasing the sentence about hospitals having to choose between emergency care and non-urgent surgery to make it more concise.
Summary-Iter2: Hospitals in Wales may have to prioritize emergency care over non-urgent surgery during peak winter months, according to Dr. Andrew Goodall. Some surgical procedures have already been cancelled until after Christmas.
more iterations...

TABLE 5.9. An example of iterative summary refinement from the XSum dataset. The revision between the two iterations and their corresponding comments are presented in the same color. The blue color refers to the rephrase revision and the orange color refers to the remove operation.

5.6. Conclusion

In this chapter, we introduce a novel framework for text summarization by iteratively refining summaries with model feedback. Our framework is entirely based on large language models and does not require supervised training or reinforcement learning alignment. Furthermore, we demonstrate that the system enhances faithfulness and controllability by integrating knowledge and topic extractors. Through extensive experiments and analyses on three benchmark datasets, we show that our iterative summarization system outperforms one-shot generation setting systems with LLM, underscoring the effectiveness of our approach.

Our human evaluation reveals that the summary refinement by our framework can effectively follow the self-evaluation feedback, albeit being highly biased toward its own evaluation criteria rather than human judgment. We believe this potential issue could be mitigated with human-in-the-loop feedback. We anticipate that the insights gained from this work will inform future research endeavors aimed at constructing more robust LLM-based summarization systems.

CHAPTER 6

Fusing Extractive and Abstractive Summarization with Large Language Model

6.1. Introduction

Document summarization aims to compress text material while retaining its most salient information. With the increasing amount of publicly available text data, automatic summarization approaches have become increasingly important. These approaches can be broadly classified into two categories: abstractive and extractive summarization. While abstractive methods [**78**] have the advantage of producing flexible and less redundant summaries, they often struggle with generating ungrammatical or even nonfactual contents [**51**]. In contrast, extractive summarization directly selects sentences from the source document to form the summary, resulting in summaries that are grammatically correct and faithful to the original text.

The growing interest in applying advanced large language models (LLMs), such as ChatGPT [1], for text summarization tasks has sparked significant attention. A recent study by [**34**] compared GPT-3 with traditional fine-tuning methods and found that, despite lower Rouge scores, human annotators preferred the GPT-3 generated text. Another study by [**133**] conducted a comprehensive analysis of large language models for news summarization and found that the generated summaries were comparable to those produced by humans. However, existing research [**71**,**117**] has only focused on abstractive summary approaches, and the performance of ChatGPT for extractive summarization remains an open question. Moreover,

[1] https://chat.openai.com/chat

the hallucination problem has dramatically hindered the practical use of abstractive summarization systems, highlighting the need to explore extractive summarization with LLMs for faithful summaries.

In this chapter, we comprehensively evaluate ChatGPT's performance on extractive summarization and investigate the effectiveness of in-context learning and chain-of-thought explanation approaches. Our experimental analysis demonstrates that ChatGPT exhibits inferior extractive summarization performance in terms of ROUGE scores compared to existing supervised systems, while achieving higher performance based on LLM-based evaluation metrics. Additionally, we observe that using an extract-then-generate pipeline with ChatGPT yields large performance improvements over abstractive baselines in terms of summary faithfulness.

The main contributions of this chapter are:

- This study represents the first attempt to extend the application of ChatGPT to extractive summarization and evaluate its performance.
- We investigate the effectiveness of in-context learning and chain-of-thought reasoning approaches for extractive summarization using ChatGPT.
- We further extend the extraction step to abstractive summarization and find that the extract-then-generate framework could improve the generated summary faithfulness by a large margin compared to abstractive-only baselines without compromising summary quality.

Together, these contributions provide novel insights into the capabilities of ChatGPT for faithful text summarization tasks and underscore the potential of LLMs as a powerful tool for NLP research.

6.2. Related Work

Most extractive summarization works formulate the task as a sequence classification problem and use sequential neural models with diverse encoders such as recurrent neural networks [**13**, **78**] and pre-trained language models [**66**, **125**]. Another group of works formulate extractive summarization as a node classification problem and apply graph neural networks to model inter-sentence dependencies [**107**, **115**, **123**, **124**].

Several studies also explored the use of large language models [**11**] for summarization. [**34**] found that while the former obtained slightly lower Rouge scores, human evaluators preferred them. Likewise, [**133**] reported that large language model-generated summaries were on par with human-written summaries in the news domain. In addition, [**117**] explored the limits of ChatGPT on query-based summarization other than generic summarization. [**71**] explored the use of ChatGPT as a factual inconsistency evaluator for abstractive text summarization. [**127**] proposed a self-evaluation and revisement framework with ChatGPT. While most of the existing research has focused on abstractive summarization, this work aims to investigate the applicability of ChatGPT to extractive summarization and examine whether extractive methods could enhance abstractive summarization faithfulness.

6.3. Method

6.3.1. Task Formulation.

Extractive summarization systems form a summary by identifying and concatenating the most salient sentences from a given document. These approaches have gained widespread traction in various real-world applications owing to their ability to produce accurate and trustworthy summaries devoid of grammatical inconsistencies.

Formally, given a document d consisting of n sentences, the goal of an extractive summarization system is to produce a summary s comprising of $m (m \ll n)$ sentences, by directly extracting relevant sentences from the source document. Most existing work formulates it

as a sequence labeling problem, where the sentences are selected by model M based on the probability of whether it should be included in the summary s:

$$\hat{s} = \arg\max_{s} p_M(s \mid d). \tag{6.1}$$

In the training of supervised summarization models, it is common to employ a greedy algorithm, as described in [**77**], to generate extractive ground-truth labels (ORACLE) by selecting multiple sentences that maximize the ROUGE score compared to the gold summary.

6.3.2. In-context Learning.

Recent studies have shown that large language models have strong few-shot performance on various downstream tasks, known as in-context learning (ICL) [**11**]. The standard ICL prompts a large language model, M, with a set of k exemplar document-summary pairs and predicts a summary $\hat{s}$ for the document by:

$$\hat{s} = \arg\max_{s} p_M(s \mid d, \{(d^1, s^1)...(d^k, s^k)\}). \tag{6.2}$$

Besides simple input-output pairs, previous works also show that including explanations and chain-of-thought (COT) reasoning in prompts [**110**] also benefits language models, represented as:

$$\hat{s} = \arg\max_{s} p_M(s \mid d, C). \tag{6.3}$$

Here $C = \{(d^1, e^1, s^1)...(d^k, e^k, s^k)\}$ is the set of input-explanation-output triplets in prompts. Besides zero-shot setting, this study also investigates the impact of in-context learning on extractive summarization, with and without explanations.

6.3.3. Extract-abstract Summarization.

It is not new to use extractive summaries to guide abstractive summary generations [**23**, **108**]. Here we also propose to use LLM in a two-stage manner: extract salient sentences to form extractive summaries (s^E) first, and then ask the LLM to generate summaries guided by the extractive summaries, represented as:

$$p(s \mid d) = \prod_{t=1}^{T} p\left(s_t \mid s_{<t}, d, s^E\right), \tag{6.4}$$

where $s_{<t}$ denotes the previous generated tokens before step t. We explore the extract-then-generate pipeline in this study, aiming to alleviate the hallucination problems in LLM summary generation.

Dataset	**Domain**	**Doc #words**	**Sum #words**	**#Ext**
Reddit	Media	482.2	28.0	2
XSum	News	430.2	23.3	2
CNN/DM	News	766.1	58.2	3
PubMed	Paper	444	209.5	6

TABLE 6.1. Detailed statistics of the datasets. Doc # words and Sum # words refer to the average word number in the source document and summary. # Ext refers to the number of sentences to extract.

6.4. Experiments and Analysis

6.4.1. Experiment Settings.

Datasets: We selected four publicly available benchmark datasets as listed in Table 6.1, ensuring they are consistent with previous fine-tuning approaches.

- CNN/DailyMail [**37**] is one of the most widely adopted summarization datasets that contains news articles and corresponding highlights as summaries. We use the non-anonymized version and follow the common training/validation/testing splits (287,084/13,367/11,489).

- XSum [**79**] is a one-sentence news summarization dataset with professionally written summaries. We follow the common training/validation/testing splits (204,045/ 11,332/11,334).
- PubMed [**16**] is a scientific paper summarization dataset and we use the introduction section as the article and the abstract section as the summary following [**136**] with common training/validation/testing splits (83,233/4,946/5,025).
- Reddit [**49**] is a highly abstractive dataset collected from social media platforms with training/validation/testing splits (41,675/645/645).

Evaluation: We conducted an evaluation of ChatGPT's summarization performance utilizing ROUGE [**60**], following previous studies. We also employed a GPT-based evaluation metric, G-EVAL [**64**]. To investigate the faithfulness of the summaries, we employed common metrics FactCC [**51**] and QuestEval [**98**]. FactCC is a weakly supervised BERT-based model metric that verifies factual consistency through rule-based transformations applied to source document sentences. It shows a high correlation in assessing summary faithfulness with human judgments. Questeval [**98**] is a question answering based metric to measure summary faithfulness by how many generated questions could be answered by the summary.

Experiment Setup: We employed the GPT-3.5-turbo model[2] for the generation and assessment of summaries, maintaining a temperature setting of 0 to ensure reproducibility.

Regarding the datasets, a random sampling method was adopted, where 1000 samples were chosen for each dataset for experimental purposes. Furthermore, a smaller subset of 50 samples was utilized for the discovery of optimal prompts and hyperparameters. The random seed was established at 101 to promote consistency.

The experiments involving each dataset, which includes 1000 examples, will run for 1.5 hours to perform both inference and evaluation. The detailed prompts used in the experiments under different settings are summarized in Table 6.2.

6.4.2. Experiments Results.

[2]https://platform.openai.com/docs/guides/gpt/chat-completions-api

Setting	Prompt
Extractive	**System**: You are an extractive summarizer that follows the output pattern. **User**: Please extract sentences as the summary. The summary should contain **m** sentences. Document: [*Test Document*] [*Format Instruction*].
Abstractive	**System**: You are an abstractive summarize that follows the output pattern. **User**: Please write a summary for the document. Document: [*Test Document*] [*Format Instruction*]
In-context	**System**: You are an extractive summarizer that follows the output pattern. **User**: The following examples are successful extractive summarization instances: [*n Document-Summary Pairs*]. Please summarize the following document. Document: [*Test Document*]. The summary should contain m sentences. [*Format Instruction*].
Explanation	**System**: You are an extractive summarizer that follows the output pattern. **User**: The following examples are successful extractive summarization instances: [*n Document-Summary-Reason Triads*]. Please summarize the following document and give the reason. Document: [*Test Document*]. The summary should contain m sentences. [*Format Instruction*].
Extract-abstract	**System**: You are an abstractive summarizer that follows the output pattern. **User**: Please revise the extracted summary based on the document. The revised summary should include the information in the extracted summary. Document: [*Test Docuemnt*] Extractive Summary: [*Extractive Summary*] [*Format Instruction*].
Evaluator	**System**: You are a summary evaluator that follows the output pattern. You give scores for the summaries based on the comprehensive consideration following criteria: (1) Coherence: "the collective quality of all sentences"; (2) Consistency: "the factual alignment between the summary and the reference"; (3) Fluency: " the quality of individual sentences"; (4) Efficiency: "If the summary is concise" **User**: Please evaluate the summary based on the reference summary.Reference:[*Reference Summary*] Summary:[*Predicted Summary*][*Format Instruction*].

TABLE 6.2. Prompts used for both extractive and abstractive summarization. m is the number of extracted sentences defined in Table 6.1. Document-summary pairs and document-summary-reason triads are the input contexts. n is the number of context instances.

Models	CNN/DM				XSum			
	R1	**R2**	**RL**	**G-EVAL**	**R1**	**R2**	**RL**	**G-EVAL**
SOTA-Ext	**44.41**	**20.86**	**40.55**	3.28	**24.86**	4.66	**18.41**	2.60
ChatGPT-Ext	39.25	17.09	25.64	3.24	19.85	2.96	13.29	2.67
+ context	42.38	17.27	28.41	**3.30**	17.49	3.86	12.94	2.69
+ reason	42.26	17.02	27.42	3.10	20.37	**4.78**	14.21	**2.89**
SOTA-Abs	**47.78**	**23.55**	**44.63**	3.25	**49.07**	**25.13**	**40.40**	2.79
ChatGPT-Abs	38.48	14.46	28.39	**3.46**	26.30	7.53	20.21	**3.47**
	Reddit				**PubMed**			
Models	**R1**	**R2**	**RL**	**G-EVAL**	**R1**	**R2**	**RL**	**G-EVAL**
SOTA-Ext	**25.09**	**6.17**	**20.13**	1.82	**41.21**	**14.91**	**36.75**	2.03
ChatGPT-Ext	21.40	4.69	14.62	**1.87**	36.15	11.94	25.30	2.12
+ context	22.32	4.86	14.63	1.83	36.78	11.86	25.19	2.14
+ reason	21.87	4.52	14.65	1.83	37.52	12.78	26.36	**2.18**
SOTA-Abs	**32.03**	**11.13**	**25.51**	1.87	**45.09**	**16.72**	**41.32**	**2.78**
ChatGPT-Abs	24.64	5.86	18.54	**2.43**	36.05	12.11	28.46	2.70

TABLE 6.3. Summarization results on four benchmark datasets. '+context' and '+reason' refer to ChatGPT with three in-context examples and human reasoning. The best results in both extractive and abstractive settings are in bold.

The overall results are shown in Table 6.3. The upper block includes extractive results and SOTA scores from MatchSum [**136**]. The lower block includes abstractive results and SOTA scores from BRIO [**69**] for CNN/DM and XSum, SummaReranker [**90**] for Reddit, and GSum [**23**] for PubMed.

It is observed that ChatGPT generally achieves lower ROUGE scores in comparison to previous fine-tuning methods for all datasets under both extractive and abstractive settings but achieves higher scores in terms of the LLM-based evaluation metric G-EVAL. The findings are consistent with the previous conclusions in [**34**, **133**].

We also observe that ChatGPT-Ext outperforms ChatGPT-Abs on extractive datasets CNN/Dailymail and PubMed, while performing worse in the other two abstractive datasets. We argue that the results are due to the bias within the reference summaries of the dataset and the limit of ROUGE scores.

Nonetheless, we notice that despite being primarily designed for generation tasks, ChatGPT achieves impressive results in extractive summarization, which requires comprehension of the documents. The decoder-only structure of ChatGPT doesn't degrade its comprehension capability compared to encoder models like BERT. We also find that the ROUGE score gap between ChatGPT and SOTA fine-tuned baselines is smaller in the extractive setting than in the abstractive setting.

In-context: The detailed in-context learning results can be found in Table 6.4. The results also indicate that in-context learning and reasoning are generally beneficial for the extractive summarization task across four datasets in different domains. We only observe performance degradation for in-context learning on the XSum dataset. We argue that the degradation comes from the short ORACLE of XSum, which brings more confusion with a few ORACLE examples. However, with chain-of-thought reasoning explanations, ChatGPT can better understand the pattern and thus shows improvements with in-context reasoning.

# Context	CNN/DM			XSum		
	R1	**R2**	**RL**	**R1**	**R2**	**RL**
0	39.25 ± 0.23	15.36 ± 1.10	25.90 ± 0.97	19.85 ± 2.59	2.96 ± 2.59	13.29 ± 1.30
1	40.62 ± 0.70	17.00 ± 1.06	26.44 ± 0.84	15.33 ± 0.50	2.48 ± 0.19	11.48 ± 0.13
1w/R	38.83 ± 0.91	14.94 ± 2.53	25.36 ± 1.82	17.86 ± 1.73	3.29 ± 0.85	12.55 ± 1.29
2	40.91 ± 0.69	15.68 ± 0.61	26.13 ± 0.83	18.61 ± 0.39	4.42 ± 0.97	14.06 ± 2.01
2w/R	41.70 ± 0.70	15.95 ± 0.92	26.98 ± 1.33	17.95 ± 3.03	4.11 ± 1.01	13.46 ± 1.76
3	$\mathbf{42.38 \pm 0.13}$	17.27 ± 0.23	$\mathbf{28.41 \pm 0.31}$	17.49 ± 1.87	3.86 ± 1.55	12.94 ± 2.16
3w/R	42.26 ± 1.38	17.02 ± 1.60	27.42 ± 1.62	$\mathbf{20.37 \pm 1.61}$	$\mathbf{4.78 \pm 0.44}$	$\mathbf{14.21 \pm 1.07}$
4	42.26 ± 0.50	$\mathbf{17.41 \pm 0.83}$	27.96 ± 0.83	16.68 ± 1.56	3.72 ± 0.20	12.12 ± 1.19
4w/R	41.23 ± 0.93	17.08 ± 0.38	28.25 ± 0.93	18.17 ± 0.28	4.05 ± 0.38	12.74 ± 0.94
5	40.71 ± 1.92	16.96 ± 0.91	27.42 ± 1.26	17.43 ± 1.08	3.53 ± 0.96	12.33 ± 0.51
5w/R	40.18 ± 0.83	15.15 ± 1.44	25.98 ± 1.91	19.55 ± 0.64	4.29 ± 0.46	13.13 ± 0.68

TABLE 6.4. In-context learning experimental results on CNN/DM and XSum datasets. For each dataset, we randomly sampled 50 data from the test set. In each section, w/R means we provide human written reasons for each context document. For the test document, we also ask the system to generate the reason why it choose selected sentences.

6.4.3. Extract Then Generate.

We conduct further experiments to examine the effectiveness of the extract-then-generate framework as presented in Table 6.5.

Dataset	Setting	RL	G-EVAL	FactCC	QuestEval
Reddit	Abs	18.54	2.43	9.46	40.79
	Ext-Abs	18.26	2.60	**60.40**	**49.45**
	Oracle-Abs	**19.37**	**2.64**	59.75	48.93
XSum	Abs	20.21	2.67	5.42	46.14
	Ext-Abs	18.55	2.28	**55.73**	**53.25**
	Oracle-Abs	**21.10**	**2.72**	55.03	53.21
PubMed	Abs	28.46	2.70	8.37	42.83
	Ext-Abs	26.50	2.81	26.38	44.32
	Oracle-Abs	**26.51**	**2.83**	**27.35**	**44.50**
CNN/DM	Abs	28.39	3.24	6.35	45.32
	Ext-Abs	29.16	3.50	51.65	51.72
	Oracle-Abs	**33.32**	**3.51**	**53.67**	**52.46**

TABLE 6.5. Summarization results of the extract-then-generate pipeline. Abs, Ext-Abs, and Oracle-Abs refer to the generate-only baseline, the extract-then-generate pipeline, and generation based on ORACLE, respectively.

The results show large improvements in summary factual consistency across all four datasets with the extract-then-generate framework. Notably, the FactCC scores are extremely low for generate-only baselines (less than 10 percent), highlighting the hallucination problems of ChatGPT-based summarization, where ChatGPT tends to make up new content in the summary.

Nevertheless, the extract-then-generate framework effectively alleviates the hallucination problem of abstractive summaries by guiding the summary generation process with extracted salient sentences from the documents. We also find that guiding ChatGPT summary generation with its own extracted summaries leads to similar summary faithfulness improvements compared to guiding generation with ORACLE.

In terms of summary quality, the results demonstrate that the performance of ChatGPT improves largely in terms of ROUGE scores when grounded with the ORACLE summaries. However, the ROUGE score performance of the extract-then-generate framework relies heavily on the extractive performance when grounded with its own extractive summaries. In summary, the extract-then-generate framework could effectively improve the summary faithfulness with similar or even better summary quality.

6.4.4. Positional Bias.

Lead bias is a common phenomenon in extractive summarization, especially in the news domain, where the early parts of an article often contain the most salient information. As shown in Figure 6.1, we find that the position distribution of the ChatGPT-extracted summary sentences is skewed towards a higher position bias than the ORACLE sentences. In addition, in-context learning brings more positional bias to the summaries. The results indicate that LLMs may rely on superficial features like sentence positions for extractive summarization.

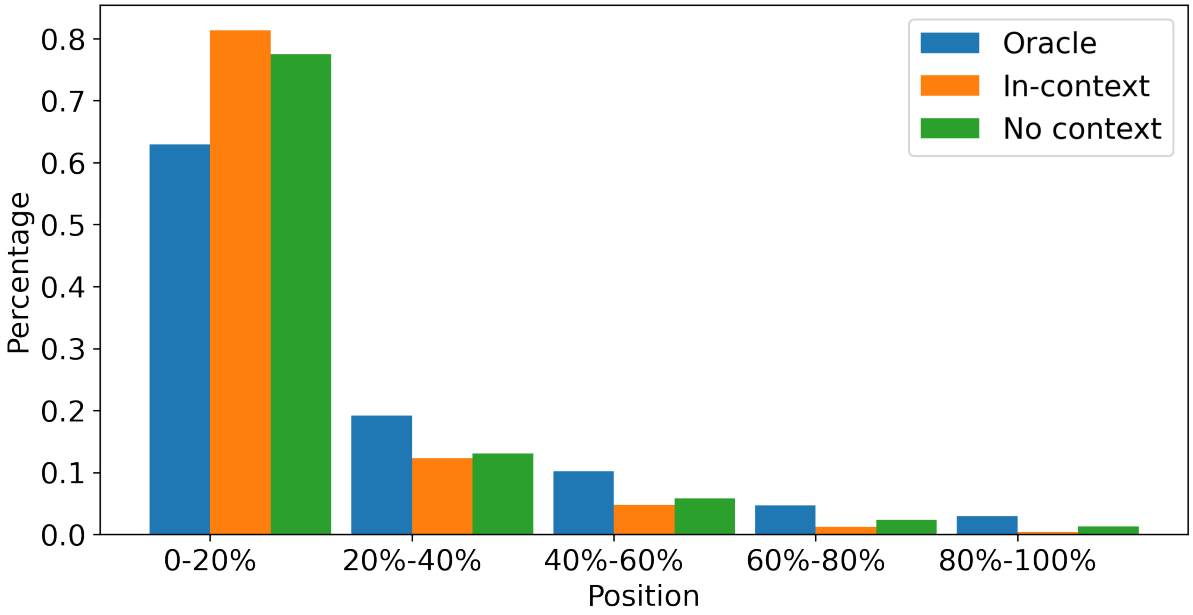

FIGURE 6.1. Position distribution of extracted sentences on 1000 random samples from the CNN/DM test set.

6.4.5. Case Study.

Here we show a case study example from the CNN/Daily dataset in Table 6.6 and the ChatGPT-generated summaries with different prompt settings in Table 6.7.

Document	Daredevil Nik Wallenda says he'll walk untethered on top of a 400-foot observation wheel in Orlando, Florida, this month. Wallenda said Monday at a New York City news conference that the Orlando Eye will be moving when he attempts his feat on April 29. The Orlando Eye, part of a new entertainment complex, will offer views of central Florida from inside 30 enclosed, air-conditioned glass capsules when it opens to the public on May 4. Eyes on the prize: high-wire performer Nik Wallenda announces his latest stunt at the 400-foot Orlando Eye, during a news conference, in New York on Monday. Tough challenge: the 36-year-old daredevil will walk atop the Orlando Eye as it turns on April 29. The Orlando Eye team issued a statement saying it's excited to have Wallenda attempt the 'amazing stunt.' No distance for the performance has been set yet, but Wallenda, 36, said he was not likely to walk the entire 20 minutes or so that it takes the wheel to go a full circle. Wallenda previously walked atop a Ferris wheel in Santa Cruz, California, but he said the size of the much larger Orlando wheel and the fact that he will not use a pole sets this stunt apart. The seventh-generation member of the 'Flying Wallenda' family of acrobats has walked across the Grand Canyon and Niagara Falls. In November, he walked twice between two Chicago skyscrapers without a net or harness, doing the second walk blindfolded. Wallenda is the great-grandson of Karl Wallenda, who fell to his death in Puerto Rico at 73.
Reference	The 36-year-old will stage his next stunt on April 29. In November, Wallenda walked back and forth between two Chicago skyscrapers in a live television event. His great-grandfather Karl Wallenda died in a tightrope walk in Puerto Rico in 1978. Wallenda has also tightrope walked across Niagara Falls and the Grand Canyon.
ORACLE	Tough challenge: the 36-year-old daredevil will walk atop the Orlando Eye as it turns on April 29. The seventh-generation member of the 'Flying Wallenda' family of acrobats has walked across the Grand Canyon and Niagara Falls. In November, he walked twice between two Chicago skyscrapers without a net or harness, doing the second walk blindfolded.

TABLE 6.6. The source document, reference summary, and extractive ground truth of case study example.

Abstractive	Daredevil Nik Wallenda plans to walk untethered atop the 400-foot Orlando Eye observation wheel on April 29th. The 36-year-old high-wire performer, a seventh-generation member of the "Flying Wallenda" acrobat family, has previously completed daring walks across the Grand Canyon, Niagara Falls, and between Chicago skyscrapers. Although the exact distance for this stunt hasn't been determined, Wallenda stated he's unlikely to walk the entire 20 minutes it takes for the wheel to complete a full circle.
Extractive	Daredevil Nik Wallenda says he'll walk untethered on top of a 400-foot observation wheel in Orlando, Florida, this month. Wallenda said Monday at a New York City news conference that the Orlando Eye will be moving when he attempts his feat on April 29. The 36-year-old daredevil will walk atop the Orlando Eye as it turns on April 29.
In-Context	Daredevil Nik Wallenda says he'll walk untethered on top of a 400-foot observation wheel in Orlando, Florida, this month. Wallenda said Monday at a New York City news conference that the Orlando Eye will be moving when he attempts his feat on April 29. Wallenda is the great-grandson of Karl Wallenda, who fell to his death in Puerto Rico at 73.
Extract-abstract	Nik Wallenda, a daredevil and seventh-generation member of the "Flying Wallenda" family of acrobats, will attempt to walk untethered on top of a 400-foot observation wheel in Orlando, Florida, on April 29. The Orlando Eye, part of a new entertainment complex, will offer views of central Florida from inside 30 enclosed, air-conditioned glass capsules when it opens to the public on May 4. Wallenda previously walked atop a Ferris wheel in Santa Cruz, California, but he said the size of the much larger Orlando wheel and the fact that he will not use a pole sets this stunt apart. Wallenda is the great-grandson of Karl Wallenda, who fell to his death in Puerto Rico at 73.

TABLE 6.7. The output summaries under different settings of the case study example in Table 6.6.

6.5. Conclusion

In summary, this chapter presents a comprehensive evaluation of ChatGPT's performance on extractive summarization across four benchmark datasets. The results highlight ChatGPT's strong potential for the task and the feasibility of generating factual summaries using the extract-generate framework. Overall, this study suggests that ChatGPT is a powerful tool for text summarization, and we hope the insights gained from this work can guide future research in this area.

CHAPTER 7

Conclusion and Future Directions

7.1. Conclusion

This dissertation has focused on natural language processing methods and systems that facilitate efficient access and digestion of data for humans through automatic text summarization. The main emphasis lies on the three critical steps of constructing intelligent and reliable summarization systems: document modeling and understanding, salient information extraction, and faithful summary generation. The overarching goal is to develop an advanced AI assistant system with profound semantic understanding and generation capabilities.

To this end, I have presented my work that addresses problems falling under three broad categories:

- The first focus is on enhancing the language understanding capabilities of these systems through the structural modeling of text documents, enabling machines to better comprehend the semantic meaning and inherent logic of documents.
- The second focus is on identifying and extracting salient information from documents for extractive summarization. This is achieved by modeling the salience of sentences and extracting information from documents from a holistic perspective.
- The last focus is on improving the quality and faithfulness of generated summaries for abstractive summarization. This is achieved by controlling and augmenting the generation decoding process, and iteratively revising the summary outputs.

In my work, I have also demonstrated how to build intelligent summarization systems using backbones from deep learning models, pre-trained language models, and recently, large language models. The text summarization research track has experienced significant breakthroughs in the past few years, and I am honored to have contributed to this development.

7.2. Future Work

Looking into the future, I am particularly enthusiastic about a few directions of research in text summarization, and more broadly, in natural language processing.

7.2.1. More Advanced Summarization Systems.

Great progress has been achieved in the development of automatic text summarization systems. In the future, I am very interested in expanding the existing summarization system circle, providing broader application scenarios and impacts.

Firstly, building multi-modal summarization systems that could take different forms of data input such as figures, code, video, and tables is of great interest. Summarization is an essential task not only for text but generally for all formats of data. Enabling summarization systems to understand different formats of input will spark innovation and address real-world needs.

Secondly, I am interested in incorporating world knowledge and common sense into the cycle, building open-domain summarization systems, and creating retrieval-augmented summarization systems. Existing ATS frameworks mostly rely on parametric knowledge from pre-trained LMs to generate the summary. However, this knowledge could be out of date and contain factual errors. If we could retrieve world knowledge and common sense from an external knowledge base, the quality of the generated summary would be further improved. This also ensures that the summaries are accurate, up-to-date, and factually correct, which is essential for many applications.

Thirdly, I am interested in developing customized and personalized ATS systems that could produce more tailored and relevant summaries. Different users have different preferences in terms of summary formats, length, and density. How to build summarization systems that could take more users' input and adjust the style of summary generation according to the user's preference is another promising direction.

7.2.2. Harnessing Large Language Models.

Large language models such as ChatGPT and GPT-4 have excelled in numerous NLP tasks, demonstrating their remarkable capacity to distill knowledge from web-scale pre-training corpora, thereby reshaping the entire NLP research landscape. I am also interested in continuing my research on LLMs.

I aim to investigate the integration of real-time, streaming new information into LLMs. Currently, after their initial pre-training, LLMs remain static and unable to adapt to new information in our ever-changing world. Incorporating and updating knowledge could provide a more cost-effective and efficient approach, considering that the training of LLMs requires tens of millions of dollars. I am also interested in exploring methods to integrate LLMs with different data modalities and unlock new dimensions of AI capabilities.

In terms of applying LLMs for text summarization, my research interest lies in providing explainable and interpretable outputs, and addressing fairness and bias issues in the output summaries.

Bibliography

[1] Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, 2023.

[2] Chenxin An, Ming Zhong, Zhiyong Wu, Qin Zhu, Xuan-Jing Huang, and Xipeng Qiu. Colo: A contrastive learning based re-ranking framework for one-stage summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5783–5793, 2022.

[3] Diego Antognini and Boi Faltings. Gamewikisum: a novel large multi-document summarization dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6645–6650, 2020.

[4] Ben Athiwaratkun, Cicero dos Santos, Jason Krone, and Bing Xiang. Augmented natural language for generative sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 375–385, 2020.

[5] Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. Parallel iterative edit models for local sequence transduction. 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)*, pages 4260–4270, Hong Kong, China, November 2019. Association for Computational Linguistics.

[6] Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang-goo Lee. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 10–20, 2019.

[7] Seyed Ali Bahrainian, Sheridan Feucht, and Carsten Eickhoff. Newts: A corpus for news topic-focused summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 493–503, 2022.

[8] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[10] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[12] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

[13] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, 2016.

[14] Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1027–1038, 2019.

[15] Sangwoo Cho, Chen Li, Dong Yu, Hassan Foroosh, and Fei Liu. Multi-document summarization with determinantal point processes and contextualized representations. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 98–103, 2019.

[16] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, 2018.

[17] Peng Cui, Le Hu, and Yuanchao Liu. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, 2020.

[18] Hoa Trang Dang, Karolina Owczarzak, et al. Overview of the tac 2008 update summarization task. In *TAC*, 2008.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[20] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[21] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. Be more with less: Hypergraph attention networks for inductive text classification. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 4927–4936. Association for Computational Linguistics (ACL), 2020.

[22] Yue Dong, Andrei Mircea, and Jackie Chi Kit Cheung. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, 2021.

[23] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. Gsum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, 2021.

[24] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. All nlp tasks are generation tasks: A general pretraining framework. *arXiv preprint arXiv:2103.10360*, 2021.

[25] Elozino Egonmwan and Yllias Chali. Transformer-based model for single documents neural summarization. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 70–79, 2019.

[26] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[27] Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 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*, pages 1074–1084, 2019.

[28] Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. Text editing by command. In *North American Chapter of the Association for Computational Linguistics*, 2020.

[29] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire, 2023.

[30] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348, 2010.

[31] Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040, 2020.

[32] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2022.

[33] Tanya Goyal and Greg Durrett. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[34] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

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

[36] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of human language technologies: The 2009 annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, 2009.

[37] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[38] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[39] Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[41] Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1608–1616, 2014.

[42] Luyang Huang, Lingfei Wu, and Lu Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, 2020.

[43] Hayate Iso, Chao Qiao, and Hang Li. Fact-based Text Editing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 171–182, Online, July 2020. Association for Computational Linguistics.

[44] Hanqi Jin, Tianming Wang, and Xiaojun Wan. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6244–6254, 2020.

[45] Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. Multiplex graph neural network for extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, 2021.

[46] Jiaxin Ju, Ming Liu, Huan Yee Koh, Yuan Jin, Lan Du, and Shirui Pan. Leveraging information bottleneck for scientific document summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4091–4098, 2021.

[47] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[48] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[49] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531, 2019.

[50] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2020.

[51] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online, November 2020. Association for Computational Linguistics.

[52] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

[53] Logan Lebanoff, Kaiqiang Song, and Fei Liu. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, 2018.

[54] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pages 7871–7880, 2020.

[55] Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. Keywords-guided abstractive sentence summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8196–8203, 2020.

[56] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022.

[57] Zizhong Li, Haopeng Zhang, and Jiawei Zhang. Unveiling the magic: Investigating attention distillation in retrieval-augmented generation. In *Proceedings of the Association for Computational Linguistics: NAACL 2024*, 2024.

[58] Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1685–1697, 2021.

[59] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[60] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157, 2003.

[61] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 510–520, 2011.

[62] Jingzhou Liu, Dominic JD Hughes, and Yiming Yang. Unsupervised extractive text summarization with distance-augmented sentence graphs. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2313–2317, 2021.

[63] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

[64] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[65] Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 5070. Association for Computational Linguistics, 2019.

[66] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, 2019.

[67] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[68] Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed Hassan Awadallah. On improving summarization factual consistency from natural language feedback. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[69] Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. Brio: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, 2022.

[70] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[71] Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. Chatgpt as a factual inconsistency evaluator for abstractive text summarization. *arXiv preprint arXiv:2303.15621*, 2023.

[72] Shulei Ma, Zhi-Hong Deng, and Yunlun Yang. An unsupervised multi-document summarization framework based on neural document model. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1514–1523, 2016.

[73] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[74] Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*, pages 557–564. Springer, 2007.

[75] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[76] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[77] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[78] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[79] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, 2018.

[80] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, 2018.

[81] Ani Nenkova, Sameer Maskey, and Yang Liu. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2011.

[82] Paul Over and James Yen. An introduction to duc-2004. In *National Institute of Standards and Technology.*, 2004.

[83] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

[84] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[85] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, 2017.

[86] Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 9308–9319, 2020.

[87] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[88] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[89] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[90] Mathieu Ravaut, Shafiq Joty, and Nancy Chen. Summareranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4504–4524, 2022.

[91] Machel Reid and Graham Neubig. Learning to model editing processes. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3822–3832, 2022.

[92] Nils Reimers and Iryna Gurevych. 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)*, pages 3982–3992, 2019.

[93] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[94] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[95] Qian Ruan, Malte Ostendorff, and Georg Rehm. Histruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, 2022.

[96] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

[97] Timo Schick, A Yu Jane, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. Peer: A collaborative language model. In *The Eleventh International Conference on Learning Representations*, 2022.

[98] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. Questeval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, 2021.

[99] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[100] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[101] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*, 2022.

[102] Frederick Suppe. The structure of a scientific paper. *Philosophy of Science*, 65(3):381–405, 1998.

[103] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[104] Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618, 2007.

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[106] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[107] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing Huang. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, 2020.

[108] Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. Salience allocation as guidance for abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6094–6106, 2022.

[109] Lu Wang and Claire Cardie. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, 2013.

[110] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

[111] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*, 2022.

[112] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. Primera: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, 2022.

[113] Wen Xiao and Giuseppe Carenini. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, 2019.

[114] Wen Xiao and Giuseppe Carenini. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528, 2020.

[115] Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, 2020.

[116] Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. A unified generative framework for various ner subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, 2021.

[117] Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*, 2023.

[118] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, 2017.

[119] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.

[120] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*, 2022.

[121] Haopeng Zhang, Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Hongwei Wang, Jiawei Zhang, and Dong Yu. Unsupervised multi-document summarization with holistic inference. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 123–133, 2023.

[122] Haopeng Zhang, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. Xatu: A fine-grained instruction-based benchmark for explainable text updates. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024.

[123] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. Hegel: Hypergraph transformer for long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10167–10176, 2022.

[124] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. Contrastive hierarchical discourse graph for scientific document summarization. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 37–47, 2023.

[125] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. Diffusum: Generation enhanced extractive summarization with diffusion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13089–13100, 2023.

[126] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. Extractive summarization via chatgpt for faithful summary generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3270–3278, 2023.

[127] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. Summit: Iterative text summarization via chatgpt. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10644–10657, 2023.

[128] Haopeng Zhang, Semih Yavuz, Wojciech Kryściński, Kazuma Hashimoto, and Yingbo Zhou. Improving the faithfulness of abstractive summarization via entity coverage control. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 528–535, 2022.

[129] Haopeng Zhang and Jiawei Zhang. Text graph transformer for document classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8322–8327, 2020.

[130] Jiawei Zhang, Haopeng Zhang, Li Sun, and Congying Xia. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

[131] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11328–11339, 2020.

[132] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

[133] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12, 2024.

[134] Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, 2019.

[135] Hao Zheng and Mirella Lapata. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, 2019.

[136] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, 2020.

[137] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, 2019.

[138] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, 2018.

[139] Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, 2021.