LB/TH/46/2025
TH6060

ENHANCING THE EXPLAINABILITY OF TRANSFORMER-BASED ABSTRACTIVE SUMMARIZATION MODELS

P. H. Panawenna

239167T

Master of Science in Data Science and Artificial Intelligence

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

May 2025

ENHANCING THE EXPLAINABILITY OF TRANSFORMER-BASED ABSTRACTIVE SUMMARIZATION MODELS

P. H. Panawenna

239167T

Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science in Data Science and Artificial Intelligence

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

May 2025

DECLARATION

I declare that this is my own work and this Dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 02-07-2025

The supervisor should certify the Dissertation with the following declaration.

The above candidate has carried out research for the Master of Science in Data Science and Artificial Intelligence Dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Sandareka Wickramanayake

Signature of the Supervisor: Date: 04/07/2025

DEDICATION

This report is dedicated to my parents for their unwavering support and unconditional love throughout the years.

ACKNOWLEDGEMENT

First and foremost, I would like to extend my heartfelt gratitude to my supervisor, Dr. Sandareka Wickramanayake, for her invaluable insights, unwavering support, and motivating guidance throughout this research. Her dedication inspired me, as she supported me during countless late nights. She was available whenever I had questions, and constantly encouraged me to push my limits. The opportunities she provided for publication and her mentorship in every aspect of the project have shaped me into a better researcher.

I would also like to sincerely thank Prof. Dulani Meedeniya for her valuable insights and continued support from the very beginning. Her guidance has been an important part of this research, along with the opportunities she provided for publications.

My heartfelt appreciation goes to Mr. Kasun Gayashan Hettihewa, who worked as a Research Assistant at the Department of Computer Science and Engineering. He contributed as a co-author in publications related to this research. His generous and consistent support has been instrumental in navigating the challenges of this project. His work in comparing the effectiveness of different feature attribution methods in explaining transformer-based abstraction summarization, presented as one of the sections in our publication [1], sheds light on the utility of the explanation framework introduced in this research.

To my parents, thank you for being my constant pillars of strength, standing by me through both the highs and lows of this journey.

To my friends and family, your patience, understanding, and words of encouragement were crucial in helping me balance the demands of a full-time job while pursuing my MSc Degree. I am deeply grateful for your support.

I would also like to acknowledge my workplace, ZeroBeta (Pvt) Ltd., for funding my MSc degree and for providing the necessary study leave, which enabled me to dedicate time and effort to this research.

Finally, I am truly thankful to the medical professionals who participated in the user study of this research. Despite their demanding schedules, they took the time to offer their honest opinions and expert insights. Their contributions added immense value to the evaluation and validation of this work.

To all who supported me along the way, thank you.

ABSTRACT

Abstractive Summarization (AS) is a Natural Language Processing (NLP) task that generates a concise and coherent summary of a given document by rephrasing or paraphrasing the content. It captures the essential information rather than directly extracting sentences or phrases from the source text, as opposed to Extractive Summarization (ES). AS is used in multiple mission-critical domains such as healthcare, law, and finance. Nevertheless, the existing state-of-the-art AS models are based on black-box deep learning models such as Transformers, and they cannot explain why specific facts were included in the summary while some facts were omitted. This research proposes a novel framework to explain which facts have been excluded from the summary by a given AS model and the rationale behind the selections. The new framework, Fact Omission Explanation (FOE) , utilizes a feature attribution method to analyze the fact-selection process of a given AS model and generate a linguistic explanation of which facts have been excluded and the respective reasons. The proposed framework was assessed using the PubMed dataset and Arxiv dataset, which consists of long documents in medical and scientific domains, and PEGASUS and T5 transformer models, which are state-of-the-art transformer-based AS models. A user study was conducted with the participation of medical professionals to assess the value addition of the framework in practice. The results demonstrate that the generated explanations help ensure the trustworthiness of AS models in mission-critical domains such as healthcare.

Keywords: Abstractive Summarization, Natural Language Processing, Transformers, Explainable AI

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
AMR	Abstract Meaning Representation
AS	Abstractive Summarization
BART	Bidirectional and Auto-Regressive Transformers
BERT	Bidirectional Encoder Representations from Transformers
CA	Cross Attention
CNN	Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Networks
ERASER	Evaluating Rationales And Simple English Reasoning
ES	Extractive Summarization
FOE	Fact Omission Explanation
GI	Gradient * Input
GPT	Generative Pre-Trained Transformers
Grad-CAM	Gradient-weighted Class Activation Mapping
GRU	Gated Recurrent Unit
GSG	Gap Sentences Generation
GSR	Gap Sentences Ratio
ILP	Integer Linear Programming
INITs	INformation ITems
LED	Longformer Encoder Decoder
ML	Machine Learning
MLM	Masked Language Model
NLP	Natural Language Processing
RNN	Recurrent Neural Networks
SA	Self Attention
SP	Summarization Programs
T5	Text-to-Text Transfer Transformers
XAI	Explainable Artificial Intelligence

CHAPTER 1

INTRODUCTION

Text summarization is a Natural Language Processing (NLP) task that generates concise summaries based on an input document. Text summarization can be performed in an extractive manner or an abstractive manner. While Extractive Summarization (ES) extracts pieces of the document itself as a summary, Abstractive Summarization (AS) goes a step beyond to mimic how a human would summarize a document. That is, it will filter out the key information but present it with new words and sentence structures. Figure 1.1 by Wang et al. [2] illustrates the difference between ES and AS.

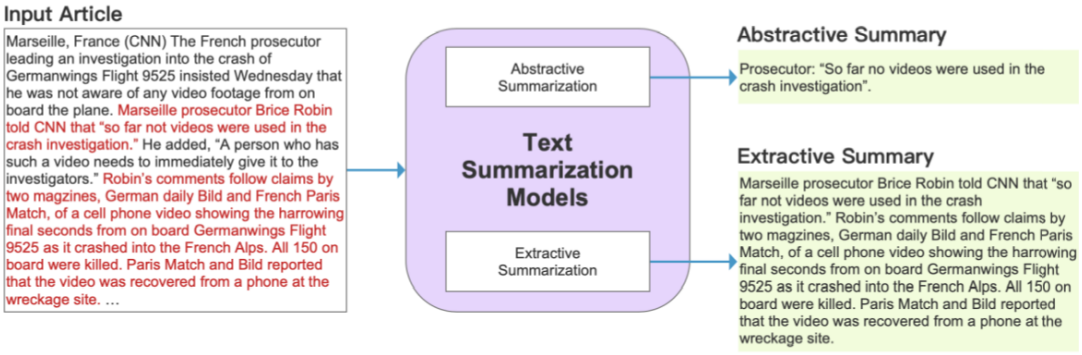

Fig. 1.1: Abstractive Summarization vs. Extractive Summarization [2]

In many Machine Learning (ML) applications, explainability is crucial, not only to instill confidence in end users regarding the output of the ML system, but also to assist designers in troubleshooting and improving performance. In text summarization, too, it is important for the end user to know what factors were considered in the summarization and to trust the summary on the stakes of their downstream tasks. This is especially important in high-stakes domains such as medicine, where doctors research treatment options; law, where attorneys analyze documents or past cases; and finance, where analysts interpret reports or news articles to inform strategic decisions. In general, any individual tasked with reviewing large volumes of information to make decisions benefits from trustworthy summaries.

Among summarization techniques, AS often offers greater value to the end user, as it closely mirrors how humans summarize, by paraphrasing and condensing key ideas, rather than copying text as is. However, this very strength introduces a major challenge: because abstractive summaries do not retain the original wording, it becomes difficult to trace which parts of the source document influenced the summary, what was excluded, and why. As a result, building explainable AS systems is both technically challenging and critically important.

At the current state-of-the-art, transformers [3] is becoming a preferred method for

AS. According to the existing literature, there are several techniques that have proven effective in enhancing the interpretability of transformer-based language models, as further discussed in Section 2.3 and Section 2.4.

While identifying the portions of the document that received the most attention is crucial, it is equally important to analyze the remainder of the document, specifically the information that was excluded during summarization. Understanding what was left out and why can significantly enhance explainability by offering insights into the model's decision-making process. This not only helps in identifying potential gaps or biases in the summary but also enables the end user to review omitted content, thereby imparting greater trust and transparency in the summarization process.

Thus, it can be established that easy access to information that is skipped when generating an abstractive summary and the reasons why would be useful for decision-makers at many levels. This will help them ensure decisions are made considering all relevant factors.

Out of the existing literature, as detailed in Chapter 2, several existing works have applied Explainable Artificial Intelligence (XAI) techniques to derive insights into AS models. However, most researchers focus on analyzing the information extracted to the output summary for explainability. There is a gap for a methodology to analyze the excluded facts and the reasons behind those exclusions.

Hence, this project focuses on XAI for building trustworthy transformer-based AS models, with a focus on explaining the omitted facts and the reasons behind the omissions.

1.1 Research Problem

XAI is currently gaining increased attention worldwide. With recent advances in Artificial Intelligence (AI) , AI and ML are becoming buzzwords in any commercial sector. Further, NLP is a field of much interest to the public as it enhances the interface between computers and machines.

A major breakthrough in NLP came with the introduction of Transformer architectures by Vaswani et al.[3], which revolutionized the field and laid the foundation for modern large language models. The widespread adoption of transformers has, in turn, sparked considerable interest in their explainability, driven by the need for users to better understand, interpret, and trust the outputs generated by these complex black box models.

As illustrated further in Chapter 2, many methods are being adopted to explain the output of transformers in NLP tasks. As described in Chapter 1, text summarization is an NLP task with many uses, and the explainability of transformers for text summarization is of vital importance. Out of the literature available for this, as detailed in Chapter 2, most researchers focus on analyzing the extracted information of the output

summary for explainability. However, depending on the downstream task, it is also crucial to understand what information is omitted during summarization and the rationale behind those omissions. This insight can assist human evaluators in reviewing the summarization model's decisions regarding excluded content. Additionally, it serves as a secondary layer of information for users, bridging the gap between the generated abstractive summary and the full source document by shedding light on key details that were not included without requiring the user to read the entire document.

Hence, the research problem explored in this research can be summarized as:

> "How can we generate easy-to-understand explanations for why transformer-based AS models omit certain key facts from input documents?"

1.2 Research Objectives

The objectives of this research can be defined as follows.

- To develop a framework that generates explanations for omitted facts in abstractive summaries produced by transformer-based AS models.
- To evaluate the effectiveness of the generated explanations from the proposed framework, particularly in mission-critical domains.

1.3 Research Questions

- How can feature attribution methods be used to identify the omission of key facts in summaries generated by transformer-based AS models?
- How to develop framework to generate user-understandable explanations for omitted content in AS outputs?
- How do domain experts perceive the relevance, usefulness, and trustworthiness of explanations generated by the proposed framework?

1.4 Scope and Limitations

This research focuses on developing an XAI framework to describe what key facts are omitted and why, during transformer-based AS.

To elaborate further on the choice of AS for the scope of this project, it is because it offers greater value addition, as it closely mirrors how humans naturally summarize text, by paraphrasing and synthesizing key information rather than simply extracting it. However, this very characteristic makes explainability more challenging. Since

the summary does not directly copy text from the source, it becomes difficult to trace which components were included or omitted, and why those choices were made. As such, developing a model capable of explaining omitted facts holds significant value for the AS task. Nevertheless, the approach proposed in this work remains extensible to ES as well.

Additionally, the scope would mainly focus on transformer-based AS, considering that the state-of-the-art for AS revolves around transformer-based models. Therefore, the feature attribution methods utilized in the proposed framework are applicable to transformers. Nevertheless, the proposed method can be made extensible to other AS models as well by applying the feature attribution methods relevant to those models. However, this will not be deeply analyzed within the scope of this research.

Given the utility of AS in summarizing long documents in mission-critical domains, most of the experiments of this study utilize a long document summarization dataset, specifically, the PubMed dataset [4] [1], which contains medical research articles and Arxiv dataset [5] [2], which contains scientific research articles. We have also experimented with shorter documents, however, the utility of the proposed framework was highlighted mostly during long document summarization. Details of this are discussed in Section 4.1 and Chapter 5.

The scope for document summarization in this project is defined as per categories defined by Wang et al. in [2] as shown in Table 1.1. For further details on the categories, refer to Section 2.1.

TABLE 1.1: SCOPE FOR SUMMARIZATION ACCORDING TO CATEGORIES BY WANG ET AL.

Category	**Scope**
Summarization Method	Abstractive
Source Document Quantity	Single Document
Source Document Length	Long
Summary Length	Short
Language	Single Language
Domain	General or Specific
Level of Abstraction	Generic

The scope for the explainability will be "Local Post-hoc" as per the categories defined by Danilevsky et al. [6]. That is, explanations will be generated for each instance separately in a separate process after the summarization is completed.

Due to resource limitations, the experiments on long documents had to be confined to input articles with less than 2000 words. However, these articles were still considerably long, which justified their usage in the experiments. Another experimental chal-

[1] https://pubmed.ncbi.nlm.nih.gov/
[2] https://arxiv.org/

lenge was the absence of a standardized evaluation metric for explanations. Therefore, for programmatic evaluation of explanations, ROUGE scores [7] and BERTScores [8] were adapted as detailed in the Chapter 4.

The remainder of this dissertation is organized as follows. Chapter 2 presents a review of the related literature. Chapter 3 describes the main contribution of this research, a framework to explain which facts have been excluded from the summary by a given AS model, and the rationale behind the selections. In Chapter 4, the experimental study is presented and the results are analyzed. This is followed by a discussion in Chapter 5 and a conclusion in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

The literature review first explores the existing work in AS, followed by transformer models in the context of NLP and AS. Then, the literature related to XAI and Explainable AS is analyzed. Figure 2.1 presents the hierarchy of the analysis along with sample work under each category.

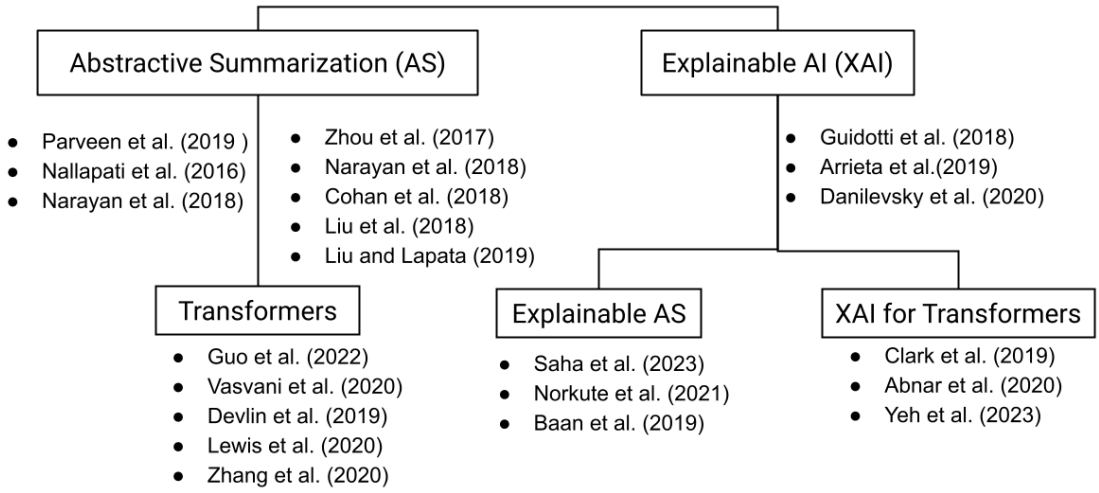

Fig. 2.1: Literature Review Categorization

2.1 Abstractive Summarization in the context of Automatic Summarization

AS is an automatic summarization technique used in natural language processing (NLP) to generate a concise summary of a piece of text by interpreting and synthesizing the information rather than simply extracting existing sentences. The alternative approach for automatic summarization is ES, where pieces of text from the original document are extracted and presented as is. In this section, let us explore the key works surrounding AS in the broader context of automatic summarization.

Automatic Summarization is gaining widespread attention due to its applications in numerous fields. In an age where content is available in abundance with the advancements of the internet and social media, summarization is a vital task that helps users consume content more efficiently. Wang et al. [2] describe several use-cases of document summarization in the modern world: one of them is news aggregation for readers to consume content quickly. Also, summarization is massively important in the legal industry where attorneys have to review a large number of documents in a short span of time [9, 10]. It can also be used in healthcare to summarize medical records to make

informed decisions [11, 12]. Finance is another field where document summarization is critical [13, 14].

Wang et al. [2] also contribute with a comprehensive categorization of a summarization tasks:

1. **Summarization Method:** Extractive vs Abstractive

   ES extracts the most important sentences or phrases from the original text and presents them as the summary. AS generates new phrases and sentences that rephrase and consolidate the important ideas from the source.

2. **Source Document Quantity:** Single-document vs Multi-document

   Single-document summarization focuses on summarizing a single source document. In contrast, multi-document summarization focuses on summarizing a collection of documents into a single output summary.

3. **Source Document Length:** Short vs Long

   In Short Document summarization, documents such as news articles or blog posts are used. In long document summarization, the sources are extensive and complex documents, such as books, reports, or collections of research papers.

4. **Summary Length:** Headline vs Short vs Long

   Summaries could be of varying lengths. A headline summarizes the document into a short phrase or sentence. Short summaries have an enhanced level of detail, containing a few sentences or a short paragraph. Long Summaries give the user a comprehensive overview consisting of several paragraphs or longer.

5. **Language:** Single-Language vs Multi-Language vs Cross-Lingual

   The most common form is Single-language translation, where the summarization model receives source documents in a single language and is expected to generate the summaries in that same language. Multi-language summarization refers to where the source document to the model could be from different languages, and the relevant output is expected to be in the same language as the source document. Cross-lingual translation refers to the source document being in a certain language, but the output is expected to be in a different language (it is a summarization+translation task).

6. **Domain:** General vs Specific domain

   Summaries could be general or specific domain. If from a specific domain (such as Law, Medicine, or Finance), special attention would need to be paid to terminologies, nuances, and special considerations specific to that domain.

7. **Level of Abstraction:** Generic vs Query-focused

   The summary could be generic in nature, or it could be specifically summarized to respond to a focused query.

To gain an insight into existing literature on text summarization, Widyassari et al. [15] comprehensive and systematic review of publications between 2008 and 2019. They categorize the research under eight research topics: extractive, abstractive, single document, multi-document, optimization, domain-specific, and real-time summarization. It is noteworthy that explainability had not come up in this research as a specific research topic in text summarization. Additionally, they state that ML is the most popular approach to tackle summarization in this recent research.

Before discussing AS in detail, let us first look at some key works related to ES. As described earlier, ES tries to extract the key sentences and phrases from the source document. A novel perspective for solving the text summarization problem was presented by Parveen et al. [16], which is based on a weighted graphical representation of documents. This is done through a topic modeling approach. As their method neither requires annotated training data nor optimization of parameters (except for the number of topics), the authors consider it an unsupervised approach. This is a specialty of this approach when compared with the existing work.

One of the key developments in ES using machine learning models was "SummaRuNNer" [17], where Nallapati et al. introduced a recurrent Neural Network Sequence Model. Here, the summarization is treated as a sequence classification problem. That is, each sentence in the source document is considered sequentially, and decided whether or not it needs to be included in the summary.

Another interesting approach for ES was introduced by Narayan et al. [18] in 2018. There, they introduced a method for ranking sentences to support ES coupled with reinforcement learning. Here, a sentence is ranked high for selection if it mostly occurs in high-scoring summaries. Reinforcement learning has been used to compare the generated summaries with the gold summary and decide on a reward and update the model. They conclude that the usage of reinforcement learning has been advantageous in outperforming the state-of-the-art for ES.

Dividing deeper into the unsupervised methods for ES, it can be seen that they rely on techniques such as scoring sentences based on their relevance, importance, or position within the source text [2]. One of the key graph-based ranking models for text processing, TextRank [19], consists of a model that represents the input as a graph. Here, the nodes of the graph could be either sentences or words, while the edges represent the similarity between nodes. Then they iteratively score nodes based on their connections, to see which components of the document are more important. LexRank [20] is another approach for unsupervised text processing. It follows the concept of eigenvector centrality in a graph representation of the original text.

While ES picks and chooses the most important components of a document for the summary, AS absorbs the key concepts in the document and generates new phrases and sentences representing them. This is closer to how a human would summarize a document [21]. Hence, AS has been gaining widespread attention in the past few decades.

An illustration of an AS provided by Zhou et al. [22] in their research on Selective Encoding for Abstractive Sentence Summarization is as follows:

- **Source Text:** "The Sri Lankan government on Wednesday announced the closure of government schools with immediate effect as a military campaign against Tamil separatists escalated in the north of the country."
- **Abstractive Summary:** "Sri Lanka closes schools as war escalates"

The above example illustrates how the abstractive summary contains words such as "Sri Lanka", "closes", "war", and "escalates", even though they are not present as they are in the source text.

A survey by Lin and Ng in 2019 [23] on the state-of-the-art AS highlights that text has shown a gradual shift from extractive methods to abstractive methods recently, especially due to the advances in ML and Deep Learning (DL) . They also describe three main steps of AS used in the traditional approaches (before the advancements of neural methods):

1. **Information Extraction:**

   This focuses on extracting important information from the input text. It could be an extraction of phrasal-level information [24] or a Query-based Extraction [25]. Genest and Lapalme [26] introduce a concept called INformation ITems (INITs) which is defined as the smallest element of coherent information in a sentence.

2. **Content Selection:**

   Here, a set of candidate phrases from the information extraction step is selected to be included in the final summary. Common methods used in this step are Heuristic Selection [26] or usage of Integer Linear Programming (ILP) ([27])

3. **Surface Realization:**

   This is the final step that consolidates the selected content using grammatical/syntactical rules to generate the abstractive summary. In the traditional approaches, natural language generators such as SimpleNLG [28] have been used for this purpose.

A classical approach to summarization is the work by Liu et al. [29], on AS using semantic representations. The basis of their work is a treebank for the Abstract Meaning Representation (AMR) . In their method, as the first step, the source text is parsed into a set of AMR graphs. Then the graphs are transformed into a summary graph, and finally, the abstractive summary is generated from the summary graph.

Sequence-to-sequence (Seq2Seq) models [30], initially introduced for machine translation, have been adapted for AS. These models use RNNs to map the source text to a summary in a token-by-token fashion.

While studying the above approaches is helpful to gain an insight into the subtasks required in an AS process, with the advancements of neural technologies, it is possible to train a model for AS directly, such that the neural model itself takes care of the above steps internally. Rush et al. [31] was the first to use neural machine translation techniques for AS.

Zhou et al. [22], in their research in 2017, introduced selective encoding for abstractive sentence summarization. This is an extension of the sequence-to-sequence framework. It has three main components:

1. **Sentence Encoder:** This is a recurrent neural network that reads the input sentences and creates a basic representation of them.

2. **Selective Gate Network:** This is a key component of this model that differentiates it from Seq-to-seq models. This creates a second-level sentence representation by means of controlling the information flow from encoder to decoder. This is specifically designed for summarization tasks.

3. **Decoder with Attention:** This is also a recurrent neural network, specifically a Gated Recurrent Unit (GRU) with attention to decoding the selected representation and producing the output summary.

Later, attention mechanisms were integrated into Seq2Seq models [32–34] to focus on relevant parts of the source text while generating the summary, improving the model's ability to capture long-range dependencies and produce coherent summaries. Since plain Seq2Seq methods tend to be factually incorrect and repetitive, See et al. have introduced a hybrid pointer-generator network [35] that copies words from the source text while producing novel words through the generator. Several other researchers have also approached AS as a two-step process of sentence selection and content generation [36]. A different perspective had been a hierarchical approach for document structural compression and coverage [37].

Another example of the usage of neural summarization is by Narayan et al. [38] where they introduce a task called "Extreme Summarization", which is a single document summarization that aims to create a short, one-sentence news summary of the

source article with a very high-level overview of what the article is about. Their model is based on Convolutional Neural Networks (CNN) . The convolutional encoder creates a multi-layer hierarchical representation over the input document. In this hierarchical representation, the words at closer distances interact at lower layers while distant words interact at higher layers to capture long-range dependencies. This uses the multi-layer convolutional structure to build a hierarchical representation over what has been predicted so far. This uses multi-hop attention for the model to remember which words it previously attended to.

Addressing a specific problem in AS, Cohan et al. introduced a discourse-aware attention model for AS of long documents in 2018 [39]. Their focus is on abstractively summarizing scientific research papers, which are an example of long-form structured document types. Here, they leverage the structured nature of the long document for effective summarization. The encoder used in their model is a hierarchical Recurrent Neural Networks (RNN) that captures the document discourse structure. They first encode each discourse section and then encode the document. Complementing the encoder, the decoder also works on different discourse sections and allows the model to more accurately represent important information from the source.

Research by a team at Google Brain [40] shows that generating English Wikipedia articles can be approached as a multi-document summarization problem. Here, they first use ES techniques to identify key information from the source documents. This is then followed by a neural abstractive model to generate the article. The abstractive model has a decoder-only architecture that can scale and work with very long sequences, as a means of attending to multiple documents. With this model, they were able to create fluent and meaningful multi-sentence paragraphs and complete Wikipedia articles.

Liu and Lapata [41] in their research in 2019 introduce a method for pretraining encoders for text summarization. The base model they use is Bidirectional Encoder Representations from Transformers (BERT) [42] (a transformer model), which they extend into a novel document-level encoder that can express the semantics of a document and obtain representations for its sentences. They focus on both ES and AS with this work. The extractive model is constructed upon the encoder by stacking several inter-sentence Transformer layers. For AS, their contribution is a new fine-tuning schedule that uses different optimizers for the encoder and the decoder. This is to reduce the difference between the two, as the encoder is pretrained while the decoder is not. This fine-tuning method has increased the quality of generated abstractive summaries.

The work discussed above are neural methods that obtained successful results in summarization, before the introduction of transformers [3]. The introduction of transformers, though initially intended for machine translation, promised great advancements to AS. Transformers-based AS is discussed in detail in Section 2.2

TABLE 2.1: SUMMARY OF LITERATURE ON ABSTRACTIVE SUMMARIZATION AND AUTOMATIC SUMMARIZATION

Paper	**Summary**
Parveen et al. [16]	Unsupervised Approach for Text Summarization based on a weighted graphical representation of documents.
Nallapati et al. [17]	ES using an RNN Model, treating summarization as a sequence classification problem.
Narayan et al. [18]	Ranking Sentences to support ES coupled with Reinforcement Learning.
Zhou et al. [22]	Selective Encoding for Abstractive Sentence Summarization. (extension of the sequence-to-sequence framework)
Sutskever et al. [30]	Sequence-to-sequence (Seq2Seq) models, initially introduced for machine translation. Later adapted for AS.
Liu et al. [29]	AS using Semantic Representations (Treebank for AMR)
Rush et al. [31]	First to use neural machine translation techniques for AS.
Wang et al. [32]	Summary-aware attention in social media short text AS
Liang et al. [33]	A selective reinforced sequence-to-sequence attention model for abstractive social media text summarization
Nallapati et al. [34]	First model for AS of single, longer-form documents and uses a new hierarchical encoder that models the discourse structure of a document, and an attentive discourse-aware decoder to generate the summary
See et al. [35]	Hybrid pointer-generator network that copies words from the source text while producing novel words through the generator one-sentence news summary with a very high-level overview.
Chen et al. [36]	Fast summarization that first selects salient sentences and then rewrites them for AS.
Li et al. [37]	Extends the basic neural encoding-decoding framework with an information selection layer
Li et al. [43]	Uses structural regularization to improve AS
Narayan et al. [38]	Extreme Summarization" using a CNN model to create one-sentence news summary with a very high-level overview.
Cohan et al. [39]	Discourse-Aware Attention Model for AS of Long Documents.
Liu et al. [40]	Generating English Wikipedia articles as a multi-document summarization problem, using both Extractive and Abstractive techniques.
Liu et al. [41]	Extend BERT into a novel document-level encoder that can express the semantics of a document and obtain representations for its sentences.

A summary of the above literature is presented in Table 2.1. It is evident that during the early stages, the work was more focused on ES, but then a gradual shift towards AS was observed, influenced by advancements in ML. Literature suggests that AS

is more effective for downstream tasks rather than ES, as it represents how a human would summarize, by picking the relevant pieces of information and restructuring the sentences into shorter, to-the-point versions. Hence, this project will focus on explanations for AS.

2.2 Transformer-based Models for Text Processing and Summarization

This section describes several Transformer-based Models and their applications in text processing and AS.

The transformer was introduced with [3], which revolutionized the NLP realm. The authors highlight that the seq-to-seq model of the transformer is solely based on multi-headed self-attention mechanisms, without using recurrence or convolutions at all. They also show that, despite the state-of-the-art performance in machine translation tasks, the model is parallelizable and requires significantly less time to train.

Based on the Transformer, the Bidirectional Encoder Representations from Transformers or BERT model [42] was introduced. BERT has been designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning from left-to-right and right-to-left in all layers. The unidirectionality constraint of the previous work is avoided in BERT by using an Masked Language Model (MLM) pre-training objective, where it randomly masks some of the tokens from the input, and the objective is to predict those words based on context. Additionally, they also use a "next sentence prediction" task that jointly pretrains text-pair representations. Followed by this pre-training, a fine-tuned can be done using labeled data from the downstream tasks. The authors show that the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks without modifying the architecture.

Following the introduction of BERT, different variations of the models were introduced to better meet requirements in different domains. FinBERT [44] and SciBERT [45] are two such models. FinBERT is specifically designed to achieve state-of-the-art Sentiment Analysis in the Finance domain [44]. SciBERT is a model specifically designed to deal with scientific publications and show improved performance on downstream NLP tasks.

HIBERT, which stands for HIerachical Bidirectional Encoder Representations from Transformers, [46] on the other hand, are a variation of BERT specifically designed for the task of ES. The HIBERT model is trained in three stages: two pre-training stages and one finetuning stage.

1. **Stage 1:** This is an open-domain pre-training, where the input data is not limited to the scientific domain.

2. **Stage 2:** This is a domain-specific pre-training on where the training data comes from, only the scientific domain.

3. **Stage 3:** This is the finetuning stage, which is done using domain-specific data only.

They have experimented with either using only Stage 1 or using Stage 2 only as well. However, they conclude the two-stage method yields the best results.

Similar to BERT, Bidirectional and Auto-Regressive Transformers (BART) [47] is another transformer model for natural language tasks. It is a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained in two steps: first, the input text is corrupted with an arbitrary noising function, and then a model learns to reconstruct the original text. The authors state that BART architecture generalizes many recent pre-training schemes, such as BERT (due to the bidirectional encoder used in BART, similar to BERT), and Generative Pre-Trained Transformers (GPT) (due to the left-to-right decoder used in BART, similar to GPT). BART has been able to show massive improvement in AS. BART creates summaries in fluent English with only a few phrases copied as is from the input. In addition to the output being factually accurate, it presents evidence from across the input document showcasing background knowledge.

Text-to-Text Transfer Transformers (T5) [48] leverages transfer learning with a unified text-to-text transformer. This methodology treats every text processing problem as a "text-to-text" problem, allowing a generalized mechanism to apply the same model, objective, training procedure, and decoding process to multiple tasks. They show T5 is successful in AS, providing coherent and largely factually correct summaries. Based on T5, a new transformer architecture, LongT5 [49] was introduced with a focus on operations on long documents, which allows for scaling both input length and model scale at the same time and also uses a new local/global attention mechanism called TGlobal. LongT5 shows state-of-the-art results in long document summarization.

Longformer [50] is a transformer-based model scalable for processing long documents for a wide range of NLP tasks without chunking the long input. Its attention pattern combines local and global information. The authors also introduce Longformer Encoder Decoder (LED) , an encoder-decoder variant of Longformer for modeling sequence-to-sequence tasks, and show that it performs well in long document summarization.

ProphetNet [51] is a new sequence-to-sequence pre-training model with a self-supervised objective named future n-gram prediction, where the next n tokens are predicted simultaneously based on previous context tokens at each time step. ProphetNet, too, is effective in AS.

OpenAI's GPT models [52] are also widely used for document summarization. The authors show that scaling up language models greatly improves task-agnostic, few-shot performance.

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) [53] is a state-of-the-art model for AS based on Transformer models developed by Google in 2020. This was specifically developed to address the research gap at the time for pre-training objectives tailored for AS tasks. Here, vital sentences are masked from an input document and generated together as one output sequence from the remaining sentences, in an approach named "self-supervised objective Gap Sentences Generation (GSG) . Their experiments show that choosing important sentences to be masked (i.e. top m sentences based on ROUGE [7] score between the sentence and the rest of the document), outperforms masking the leading sentences (the first m sentences) or randomly selected ones (m random sentences). This approach differs from BERT or BART as it masks multiple whole sentences rather than smaller continuous text spans. They define a Gap Sentences Ratio (GSR) as the number of selected gap sentences to the total number of sentences in the document. The authors note that this Ratio needs to be chosen carefully, to strike a balance between hiding enough sentences to better train the model and hiding too much information such that context is lost. 0.3 is the ratio used by the authors in their work.

BigBird Transformers [54] uses a sparse attention mechanism coupled with global attention and random attention to be able to handle long input documents efficiently. By combining BigBird with the Pegasus tokenizer method [53], the BigBirdPegasus model available on HuggingFace [55] specializes in long document summarization,

Researchers have also explored diverse training techniques to improve AS, such as SimCLS [56], a method for contrastive learning of AS that performs metric-oriented training via a generate-then-evaluate two-stage framework. Several other works have explored frameworks for generating multiple candidate summaries for AS and determining their qualities [57, 58].

A summary of the work related to transformers and their text processing and summarization applications is given in Table 2.2. After the introduction of transformers revolutionized the NLP realm, many researchers have presented novel pre-training methods curated to achieve specialized performance on different tasks. The BERT, BART, T5, and PEGASUS models are highlights of great performance on AS, with PEGASUS being the state-of-the-art for the task and having a training objective specifically designed for AS. Informed by the above analysis, this research utilizes PEGASUS and T5 models for the AS-related experiments.

TABLE 2.2: SUMMARY OF LITERATURE ON TRANSFORMER-BASED MODELS FOR TEXT PROCESSING AND SUMMARIZATION

Paper	**Summary**
Vasvani et al. [3]	Introduction of the Transformer a seq-to-seq model solely based on multi-headed self-attention mechanisms, without using recurrence or convolutions
Devlin et al. [42]	BERT model pre-trained for deep bidirectional representations from the unlabeled text using a Masked Language Model pre-training objective. Can fine-tune for a large range of NLP tasks
Lewis et al. [47]	BART, a denoising autoencoder for pretraining sequence-to-sequence models. Shows massive improvement in Abstractive Summarization.
Raffel et al. [48]	T5 treats every text processing problem as a "text-to-text" problem, allowing a generalized mechanism to apply the same model, objective, training procedure, and decoding process to multiple tasks.
Guo et al. [49]	LongT5 scales both input length and model scale simultaneously and uses a new local/global attention mechanism (TGlobal) to handle long documents.
Beltagy et al. [50]	LED is an encoder-decoder variant of Longformer for modeling sequence-to-sequence tasks suitable for long document summarization.
Qi et al. [51]	ProphetNet has self-supervised objective named future n-gram prediction, Effective in AS.
Brown et al. [52]	GPT models are widely used for document summarization. The authors show that scaling up language models greatly improves task-agnostic, few-shot performance.
Zhang et al. [53]	State-of-the-art Transformer-based model for AS. Pre-training objective tailored for AS tasks using Gap Sentences Generation.
Zaheer et al. [54]	BigBird Transformers uses a sparse attention mechanism coupled with global attention and random attention, to be able to handle long input documents efficiently. Can be combined with PEGASUS for AS tasks.
Liu et al. [56]	SimCLS is a method for contrastive learning of AS that performs metric-oriented training via a generate-then-evaluate two-stage framework.
Liu et al. [57]	BRIO framework generates multiple candidate summaries for AS and assigns probability mass according to their quality.
Ravaut et al.[58]	SummaReranker is a framework where a second-stage model performs re-ranking on a set of summary candidates.

2.3 Explainable AI

As outlined in previous sections, the growing efficiency of Transformer-based models and other ML techniques over traditional rule-based systems has brought the issue of explainability to the forefront.

This section first discusses XAI in general and then highlights some techniques that can be adapted to enhance the explainability of transformers.

A survey of the state of XAI for NLP was presented by Danilevsky et al. [6] in 2020, considering publications in the NLP conference for the most recent 7 years. There, they categorize explanations into two aspects (following the categorizations earlier by Guidotti et al. [59]).

1. The first aspect looks at whether the explanation is done on the individual prediction (local) itself or the model's prediction process as a whole (global).

2. The second aspect differentiates between the explanation emerging directly from the prediction process (self-explaining) versus requiring post-processing (post-hoc).

They further show that these aspects can be combined into the following categorization.

1. Local Post-Hoc - Explain a single prediction by performing additional operations, after the model has emitted a prediction

2. Local Self-Explaining - Explain a single prediction using the model itself. This is calculated from information made available from the model as part of making the prediction.

3. Global Post-Hoc - Perform additional operations to explain the entire model's predictive reasoning

4. Global Self-Explaining - Use the predictive model itself to explain the entire model's predictive reasoning (this is almost like a white-box model)

This categorization is key in understanding the explainability workflow of any ML model, not just NLP models.

Feature attribution is one of the main XAI approaches that explain a decision of a model by indicating the influence of each input feature (e.g., word) on the model decision. Some of the widely used feature attribution methods for transformers and DL models include Attention Visualizations [60, 61], Gradient-weighted Class Activation Mapping (Grad-CAM) [62] and Gradient * Input (GI) [63].

Attention visualization is an XAI technique for interpreting attention-based models such as transformers. Attention mechanisms in Deep Neural Networks (DNN) assign weights to input elements (e.g., words in a sentence) based on their relative importance in influencing the model's decision. Visualizing these weights facilitates the understanding of what the model attends to when making its decision [64–66]. Grad-CAM was introduced for CNN-based computer vision tasks. Grad-CAM generates heatmaps that indicate which parts of the input image contributed most to a CNN's prediction by utilizing gradients and activation output of the selected layer. GI utilizes the gradients of the model's output with respect to the inputs to calculate the influence of each input feature on the model prediction.

Self-attention mechanism is one of the key components of a transformer [3]. It is this Self-attention mechanism that allows transformers to learn and use a rich set of relationships between input elements. Therefore, visualizing attention is one of the key methods of explaining transformer output. One of the recent contributions to attention visualization is by Yeh et al. with AttentionViz [61]. This method relies on visualizing a joint embedding of the query and key vectors used by transformer models to compute attention. AttentionViz can visualize attention in both language and vision transformers. A key novelty in this work is that it can analyze global patterns across multiple input sequences, unlike previous work ([67], [60], [68]) which worked on just a single sequence (one sentence input at a time). In addition to the global view, AttentionViz can also provide details in a single attention head or input sequence.

With [69], Abnar and Zuidema propose an approach for quantifying attention flow in transformers. They state that, since the self-attention of the Transformer combines information from attended embeddings into the representation of the focal embedding in the next layer, the information originating from different tokens gets mixed constantly, making the attention weights unreliable as explanations probes. Their proposed method quantifies this flow of information through self-attention.

They propose two approaches to calculate attention scores to input tokens at each layer, by taking the embedding attention of that layer and that of the previous layers.

1. **Attention Rollout:**

   This assumes that the identities of input tokens are linearly combined through the layers based on the attention weights. The attention weights are adjusted here by rolling out the weights to capture the propagation of information from input tokens to intermediate hidden embeddings.

2. **Attention Flow:**

   This considers the attention graph as a flow network. Using a maximum flow algorithm, it computes maximum flow values, from hidden embeddings (sources) to input tokens (sinks).

They conclude that compared to raw attention, the token attentions from the above two methods have higher correlations with the importance scores obtained from input gradients. Additionally, higher correlations are present against blank-out, an input ablation-based attribution method. However, as highlighted by Chefer et al. in [70], the Attention Rollout Method assumes that attentions are combined linearly and considers paths along the pairwise attention graph, which may cause an emphasis on irrelevant tokens. This is because even average attention scores can be attenuated in this case. The method also does not distinguish positive and negative contributions to the decision. Then those positive and negative attributions could get mixed and obtain high relevancy scores, even though they ideally should have been canceled out. Chefer et al. also state that the attention flow method is somewhat slower than expected.

Overcoming the aforementioned challenges, Chefer et al.[70] have presented a study on Transformer Interpretability Beyond Attention Visualization. Their approach assigns local relevance based on the Deep Taylor Decomposition principle and then propagates these relevancy scores through the layers. This propagation involves attention layers and skip connections. This maintains total relevancy across layers. This relevancy propagation rule applies to both positive and negative attributions. The benchmark for NLP they have considered in this paper is Evaluating Rationales And Simple English Reasoning (ERASER) [71], whose task is to identify the excerpt that humans marked as leading to a decision.

Another interesting take on Transformer explainability with self-attention by Chefer et al. is [72], where they propose a Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers. They state that, unlike Transformers that only use self-attention, Transformers with co-attention need to consider multiple attention maps in parallel to highlight the relevant information. Hence, they propose the first method to explain prediction by any Transformer-based architecture, including bi-modal Transformers and Transformers with co-attention. By comparing their results with the benchmark for the three most commonly used Transformer architectures; pure self-attention, self-attention combined with co-attention, and encoder-decoder attention, they show better results compared to existing methods for single modality explainability.

A summary of the above work is given in Table 2.3. The aforementioned work highlights the importance of providing explanations for outputs provided by black-box models and provides an insight into how explainability has been interpreted in different ways and different methods of providing explanations. Following the categorization provided by Danilevsky et al. [6], this project will focus on enhancing the transformer-based AS models with a "Local Post-hoc" explanation model. The feature attribution methods applicable to the transformer models analyzed in this section will inform the XAI framework proposed in this research.

TABLE 2.3: SUMMARY OF LITERATURE ON XAI FOR NLP AND TRANSFORMERS

Paper	**Summary**
Danilevsky et al. [6]	Survey of State of XAI for NLP. Presents four categories for XAI based on whether explanations are provided as part of the inference process itself or afterwards, and whether it provides explanations for each inference separately or globally.
Vig et al. [60]	Introduces BertViz, an open-source tool that visualizes attention at multiple scales, each of which provides a unique perspective on the attention mechanism
Yeh et al. [61]	"AttentionViz" for visualizing a joint embedding of the query and key vectors used by transformer models to compute attention. Can visualize attention in both language and vision transformers. Can analyze global patterns across multiple input sequences.
Selvaraju et al. [62]	Grad-CAM is a feature attribution technique for CNNs indicating which parts of the input image contributed most to the prediction.
Shrikumar et al. [63]	GI uses the gradients of the model's output with respect to the inputs to calculate the influence of each input feature towards the model prediction.
Abnar et al. [69]	Quantifying Attention Flow in Transformers. Propose two approaches to calculate attention scores to input tokens at each layer: Attention Rollout and Attention Flow.
Chefer et al. [70]	For transformer interpretability, the methodology assigns local relevance and then propagates these relevancy scores through the layers.
Chefer et al. [72]	Generic attention-model explainability for interpreting bi-modal and Encoder-Decoder Transformers.

2.4 Explanations for Abstractive Summarization

In this section, we explore the existing key works related to XAI for AS.

Several works have explored the use of attention to understand the workings of AS models. Norkute et al. [9] have derived "attention highlights" on the input text, highlighting the key areas from which the Abstractive Summary is derived. In their study, they tested two different approaches for adding an explainability feature to a legal text summarization solution based on a DL model. The participants of their study have expressed increased trust in the DL model after the attention highlights feature was implemented. The model used in this study is a Pointer-generator network, prior to the introduction of Transformers.

Baan et al. [73] in 2019 presented a study titled "Do Transformer Attention Heads

Provide Transparency in Abstractive Summarization?". Here, they used Self Attention (SA) and Cross Attention (CA) and tried to identify the different interpretable parts of the document that each attention head focuses on (POS tags, named entities, and relative position). Hence, their study focuses on enhancing the transparency of how each head behaves rather than explaining to the user which key information was extracted from the source text and why for each output summary.

Saha et al. [74] proposed Summarization Programs (SP) , using Neural Modular Trees to enhance interpretability in AS. Each tree maps summary sentences (root nodes) to source sentences (leaf nodes) via modular operations like fusion, compression, and paraphrasing. They introduced SP-SEARCH to identify SPs for human summaries and used them to train seq2seq models that generate and execute SPs to produce summaries. However, the approach sacrifices some factual consistency and coherence compared to state-of-the-art models like BART and PEGASUS.

A summary of the work discussed above is presented in Table 2.4. We have discussed work directly related to explanations for AS. While the above studies attempted to enhance the explainability of AS models, none of them have focused on explaining what facts have been omitted and why in natural language to support end users in their decision-making processes. This research gap will be explored in this study.

TABLE 2.4: SUMMARY OF LITERATURE ON EXPLANATIONS FOR TRANSFORMER-BASED ABSTRACTIVE SUMMARIZATION

Paper	**Summary**
Norkute et al. [9]	Deriving "attention highlights" on input text, highlighting the key areas from which the Abstractive Summary is derived. Focused on Legal Document Summarization
Baan et al. [73]	Study on whether Transformer Attention Heads provide Transparency in Abstractive Summarization. Identifying the different interpretable parts of the document that each attention head focuses on (POS tags, named entities and relative position).
Saha et al. [74]	SPs uses Neural Modular Trees to enhance interpretability in AS.

2.5 Summary of Literature

In summary, the literature suggests that text summarization is a vital NLP task for the modern world that has been enhanced using transformers. The literature also highlights the importance of the explainability of the text summarization tasks, as well as the transformer models.

As highlighted in Section 2.4, while there are several studies aimed at enhancing the explainability of AS models, none of them have focused on explaining the omission

of certain key facts and the rationale behind those omissions. Additionally, existing literature has not focused on providing an explanation in natural language, tailored to support end users in their decision-making processes.

Hence, this project aims to develop a framework for explaining the outputs of transformer-based AS models with a particular focus on identifying and justifying the key fact omissions. The explanations will be presented in natural language to enhance user-friendliness. Further, the proposed method will follow a "Local Post-Hoc" approach for explanation.

CHAPTER 3

METHODOLOGY

As described in Section 1.1, this research aims to improve the explainability of transformer-based AS models by analysing which facts were omitted during summarization and why.

This section introduces the Fact Omission Explanation (FOE) framework [1], which builds upon the findings presented in our previous work in [75] and [1]. The proposed framework first identifies the sentences to which the AS model has given the least priority in generating the summary using a feature attribution method. Next, the input sentences that had low relevance to the summary but contained the key phrases of the input document are derived. KeyBERT [76] is utilized to identify the key phrases. Finally, the input document, the generated summary, and the sentences identified above are fed into an LLM to generate the explanations illustrating which facts have been omitted from the summary and the reasons for those omissions.

Section 3.1 provides the technical details of the proposed framework. The framework is designed to generate explanations that are factually consistent, complete, relevant, and coherent. This framework is an extension of our previous work in [75].

3.1 Overview of Fact Omission Explanation framework

The FOE framework consists of three primary stages. First, a feature attribution method is employed to identify sentences in the input document that the AS model assigns minimal relevance during summary generation. Next, sentences with low relevance to the generated summary but containing key phrases from the input document are extracted, employing KeyBERT [76] for key phrase identification. Finally, the input document, the generated summary, and the identified sentences are provided as input to an LLM to generate explanations, highlighting omitted facts and providing potential reasons for their exclusion. An overview of the FOE framework is shown in Figure 3.1.

The algorithm of the proposed FOE framework is shown in Algorithm 1. It accepts a Transformer-based AS model M, an input document D, an LLM, and a Key-phrase Extraction Model (KPM) as inputs. First, D is fed to M to generate the abstractive summary, S. Next, Algorithm 2 is called to obtain the Low Relevance Sentences LRS in the input document that received low relevance from M in generating the summary. Afterward, Algorithm 3 is called to extract the Key Phrases in D (KPD) and filter out the set of low-attention sentences containing any of the key phrases in the input document, denoted as LRS-KPD. Next, given D, S and LRS-KPD, prompt θ is created. Finally, the LLM generates an explanation E guided by the prompt.

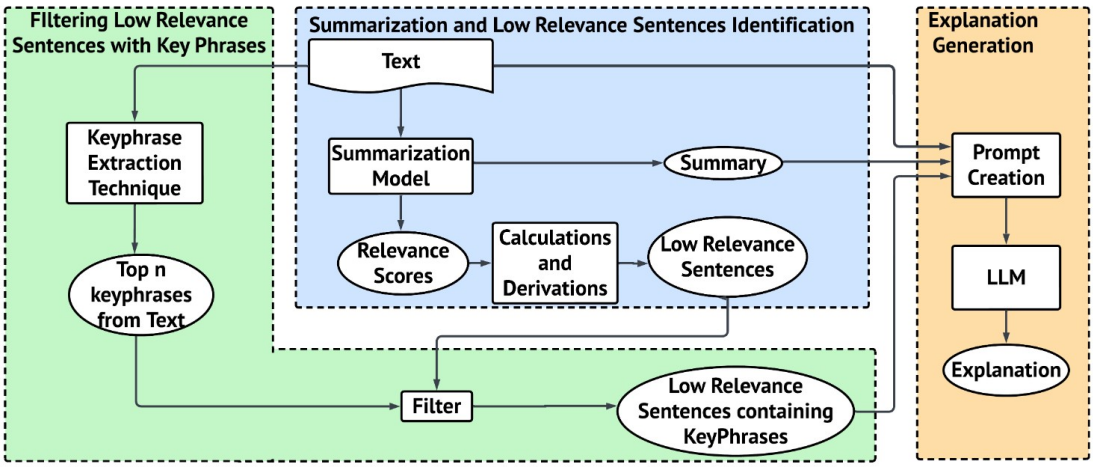

Fig. 3.1: The overview of the proposed FOE Framework

Algorithm 1 Fact Omission Explanation (FOE)

Input: Transformer-based AS Model (M), Input Document (D), Key-phrase Extraction Model (KPM), Large Language Model (LLM)

Output: Explanation for omitted facts (E)

1: $E \leftarrow \{\}$ Initialization
2: $S \leftarrow M(D)$ {*Generating summary of* D *using* M}
3: $LRS \leftarrow deriveLowRelevanceSentences(M, D, S)$ {*Algorithm 2*}
4: $LRS\text{-}KPD \leftarrow deriveLRS\text{-}KPD(LRS, D, KPM)$ {*Algorithm 3*)}
5: $\theta \leftarrow createPrompt(D, S, LRS\text{-}KPD)$
6: $E \leftarrow LLM(\theta)$ {*Generate explanation for omitted facts*}
7: **return** E

Algorithm 2 Derive Low Relevance Sentences

Input: Transformer-based AS Model (M), Input Document (D), Summary (S), Feature Attribution Method (F)

Output: Set of Low Relevance Sentences (LRS) {*The input sentences with overall low relevance to the output*}

$LRS \leftarrow \{\}$ Initialization
$S_D \leftarrow$ the set of sentences in D
$T_D \leftarrow$ the set of tokens in D
$R_D \leftarrow$ the set of relevance scores corresponding to the tokens in T_D
$n_D \leftarrow$ number tokens in D

for $t_D \in T_D$ **do**
    $\alpha_{t_D} \leftarrow F(t_D, S, M)$ {*Calculating the relevance of* t_D *towards* S *using* F}
    $R_D \leftarrow R_D + \alpha_{t_D}$
end for
$\eta \leftarrow Sum(R_D)/n_D$ {*Calculating the threshold of relevance scores as the mean*}

{*Selecting sentences with overall low relevance scores*}
for $s_D \in S_D$ **do**
    $\beta \leftarrow 0$ {*Total Relevance Score of* s_D *w.r.t.* S}
    $n_{S_D} \leftarrow getNumberOfTokens(S_D)$
    for $t_D \in s_D$ **do**
        $\alpha_{t_D} \leftarrow getRelevance(t_D, R_D)$
        $\beta \leftarrow \beta + \alpha_{t_D}$
    end for
    if $\beta/n_{S_D} < \eta$ **then**
        $LRS \leftarrow LRS + s_D$
    end if
end for
return LRS

The algorithm to derive LRS is shown in Algorithm 2. It accepts M, D, and S as the inputs. Suppose the set of tokens in D is denoted as T_D, and the set of sentences in D is denoted as S_D. Then, using a feature attribution method F, the relevance of each token $t_D \, \epsilon \, T_D$ to the output summary S is calculated, denoted by α_{t_D}. Based on those relevance scores, the threshold relevance η is calculated as the mean of the relevance scores of all the tokens in T_D. Next, a sentence-level relevance score for each sentence $s_D \, \epsilon \, S_D$ is calculated. Finally, the set of sentences with overall low relevance, less than η, is selected as the LRS.

Finally, the algorithm to derive a set of low-attention sentences containing any of the key phrases in the input document, denoted as $LRS\text{-}KPD$, is described in Algorithm 3. The algorithm accepts Key-phrase Extraction Model (KPM), D, and LRS as the inputs. Then, it passes D through KPM to obtain the key-phrases in D, denoted by KPD. Next, for each sentence in LRS, the algorithm checks whether at least one key-phrase is present in the sentence and if so, that sentence is included in the set of $LRS\text{-}KPD$.

Algorithm 3 Derive $LRS\text{-}KPD$

Input: Key-phrase Extraction Model (KPM), Input Document (D), Low Relevance Sentences (LRS)
Output: Set of Low Relevance Sentences containing Key Phrases ($LRS\text{-}KPD$)

$LRS\text{-}KPD \leftarrow \{\}$ Initialization
$S_{LRS} \leftarrow$ the set of sentences in LRS
$KPD \leftarrow KPM(D)$ {*Extracting Key phrases in D using KPM*}

{*Selecting low relevance sentences containing key-phrases in the document*}
for $s_{LRS} \in S_{LRS}$ **do**
  $hasKeyphrases \leftarrow$ **false**
  for $kpd \in KPD$ **do**
    if kpd in s_{LRS} **then**
      $hasKeyphrases \leftarrow$ **true**
    end if
  end for
  if $hasKeyphrases$ **then**
    $LRS\text{-}KPD \leftarrow LRS\text{-}KPD + s_{LRS}$
  end if
end for
return $LRS\text{-}KPD$

CHAPTER 4

EXPERIMENTAL STUDY

This chapter outlines the experimental setup used to evaluate the framework's effectiveness, focusing on the factual consistency, relevance, completeness, and coherence of the generated explanations. Section 4.1 presents the results and conclusions derived from these experiments, followed by the user study detailed in Section 4.2. The experimental results indicate that the proposed approach can generate explanations that reflect the AS model's reasoning behind the omission of certain key factors.

Given the utility of AS in summarizing lengthy documents in mission-critical domains, the PubMed dataset [4] (containing medical research articles) and the Arxiv dataset [5] (containing scientific research papers) were used in these experiments. However, due to context window limitations, it was impossible to use the original PEGASUS summarization model for this dataset. Therefore, the BigBirdPegasus model available on HuggingFace [55] was used, which is an adaptation of BigBird Transformers introduced in [54] coupled with the Pegasus tokenizer method [53]. To make the experimentation process fast and efficient, the BigBirdPegasus model pretrained on PubMed datasets was used, which was publicly available on HuggingFace [77]. Since both datasets were of a similar style, this model could be used with both datasets. For comparison, experiments were repeated using a LongT5 [49] model pre-trained with PubMed available on HuggingFace [78]. Experiments were conducted on the XSum [38] dataset as well, as a comparative study on short input documents.

The Pytorch library and HuggingFace Transformer models were used in this implementation. The experiments were conducted in a Google Colab environment with system RAM of 83.5 GB, GPU RAM of 40 GB, and Disk space of 235.7 GB. For experiments with Attention, a larger System RAM of 334.6 GB was used. Such a specification with high RAM was needed to conduct experiments on long documents. Due to resource constraints, 50 randomly selected instances of the PubMed test dataset with less than 2000 words were used for the experiments. Similar selection criteria were applied to the Arxiv test dataset. For the comparative study, 50 random instances of the XSum dataset were used.

Experiments were conducted with several prompt structures for the LLM as an ablation study, with the prompt containing:

1. $D + S + LRS\text{-}KPD$ (Algorithm 1)
2. $D + S + KPDiff$ (where $KPDiff$ are the keyphrases in D but not in S)
3. $D + S + LRS\text{-}KPD{+}KPDiff$
4. $D + S + LRS$

```
Prompt = [

        { system_prompt =
You are a Language Model explaining the decisions of an
Abstractive Summarization (AS) model.
The following information is given to you in a user prompt.
1. Text - This is the input passage
2. Summary - This is the summary generated by the AS model.
3. Low Relevance Sentences with Key Phrases (KP) - The AS model
has considered these input sentences as the least important
sentences. However, they contain KPs of the Text.

In your response mention what facts have been excluded in the
Summary compared to the Text and explain why those facts have
been excluded considering the main ideas of the Text and the
Summary.
Keep the output precise and concise. Do not exceed 100 words.
        }

        {user_prompt =
Text, Summary, Low Relevance Sentences containing KeyPhrases
        }
]
```

Fig. 4.1: LLM Prompt for FOE Methodology in Algorithm 1

The prompt for the method detailed in Algorithm 1 is depicted in Figure 4.1.

Experiments were conducted comparing the GI method and CA analysis as the feature attribution method F to derive LRS in Algorithm 1. The initial analysis conducted in our previous work [1] revealed that both CA and the GI method were effective in identifying the tokens of the input document that influenced the summary. However, the GI method requires much less computational power than attention analysis.

The KeyBert model [76], which performs keyword extraction based on BERT [42], was used as the keyphrase extraction in Algorithm 3. It extracts sub-phrases from the document and compares the cosine similarity between the sub-phrase embeddings and the full document embedding. The most similar sub-phrases are selected as key phrases. In the experiments, the candidate keyphrase length limit was set to three, as that limit allowed for diversity while also retaining coherence. The top 30 key phrases from D were extracted. Then the top 10 key phrases from S were extracted to compare and obtain $KPDiff$.

The LLM used in the experiments with PubMed and XSum was "ChatGPT-4o-latest", currently one of the most advanced models released for researchers by OpenAI [52]. The same model was used in experiments with Arxiv dataset as well. However

Arxiv experiments were additionally repeated with "Claude-3-Haiku" by Anthropic [79] for comparison.

A key consideration in the experiments was explanation length. Excessively short explanations may lack coherence, while overly long ones contradict the purpose of summarization. A 100-token limit was determined to be optimal, ensuring clarity while effectively covering omitted facts.

The user prompt components for method (1) of the prompt combinations described above (Algorithm 1) and the explanation derived from the LLM for a test example (based on PubMed article [80]) are given in Figure 4.2, Figure 4.3, Figure 4.4 and Figure 4.5.

4.1 Evaluation and Results

To evaluate the proposed method, the BertScore method [8] and the ROUGE method [7] were utilized. ROUGE is a widely used evaluation metric for assessing the quality of automatically generated summaries by comparing them to reference summaries. It quantifies the overlap of n-grams between the generated and reference summaries, with higher ROUGE scores indicating greater similarity. In this study, ROUGE-1 scores were utilized to measure unigram overlap between the reference and candidate texts. BertScore, too, is an automatic evaluation metric for text generation. It computes token-level similarity between candidate and reference sentences using contextual embeddings. This is a more nuanced assessment of semantic similarity, capturing meaning beyond basic token overlap.

In the proposed adaptation to evaluate the quality of generated explanations, the explanation is considered as the candidate text while considering the set of sentences in D containing $KPDiff$ (defined in Chapter 4), as the reference text. Let us call this set of sentences S_{KPDiff}. Hence, a higher ROUGE score reflects that the explanation has a higher degree of overlap with the parts of the input document containing key information that are not there in the summary. Similarly, a higher BertScore score indicates that the explanation has a high semantic similarity with the input document's components containing key information omitted in the summary.

To evaluate whether the proposed method truly improves the explanations of omitted facts, the results were compared with a benchmark explanation. The benchmark explanation was generated using the same LLM but with only D and S as inputs, without additional insights into the model's inner workings.

Table 4.1 and Table 4.2 present average ROUGE and BERTScores for 50 PubMed test cases with ChatGPT-4o. Results for the Arxiv dataset with ChatGPT-4o are summarized in Table 4.3 and Table 4.4, while Table 4.5 and Table 4.6 present the same using Claude-3-Haiku. Each table reports four experimental settings combining feature attribution methods (GI and CA) with AS models (BigBirdPegasus and LongT5).

“Enteric Duplications (EDs) are uncommon anomalies that can occur at any point of the gastrointestinal tract. The small intestine is the most common location; retroperitoneum is an extremely rare site. In general, diagnosed in the neonatal period or during infancy, they are increasingly diagnosed prenatally; early prenatal detection is possible. There have been seven reported cases of retroperitoneal ed cyst in the english literature. A female newborn, vaginally born at 39 weeks of gestation from a 32-year - old mother, gravid 3, para 3. Prenatal ultrasound at 22 weeks of gestation objectified an abdominal cystic mass located in the left upper abdominal quadrant, associated with fetal pyelectasis. Birth weight was 4000 g, length was 51 cm, and head circumference was 35 cm. Postnatal ultrasound found a retroperitoneal para - aortic liquid - filled mass measuring 60 mm 33 mm 22 mm. Magnetic resonance imaging (mri) confirmed the presence of a retroperitoneal cyst occupying the upper left retroperitoneal space; with mass effect displacing the left kidney down [figure 1].

Preoperative finding was a retroperitoneal cyst above the left adrenal, displacing the left kidney down, measuring 70 mm 30 mm; with no communication with any portion of the alimentary tract. Histopathologic examination revealed an ed cyst lined by small intestinal epithelium, with no ectopic gastric mucosa.

The postoperative period was uneventful; the patient was discharged on the 5 postoperative day. Magnetic resonance imaging showing retroperitoneal cyst occupying the upper left retroperitoneal space; with mass effect displacing the left kidney. Intraabdominal duplications account for two - thirds of localization, among them, jejunoileal duplications account for 65%.

EDs occur early in intrauterine life; their pathogenesis remains unclear; several hypotheses have been proposed. Reported 38 cases of ed cysts; only one was retroperitoneal communicating with a nonfunctioning right kidney. EDs are a part of fetal intraabdominal cysts which evoke several differential diagnoses including ovarian cysts, renal cysts, choledochal cysts, hepatic cysts, and mesenteric cysts. Marchitelli et al. found high concordance between prenatal and postnatal findings in fetal intra - abdominal cystic lesions (90.4%). Double - wall sign and peristaltism are suggestive of ed and help to differentiate it of differential diagnosis.

Postnatal ultrasound helps diagnosis showing an anechoic fluid - filled mass, rarely an echogenic mass is found due to complication (hemorrhage, necrosis). More serious complications may occur: ulceration, perforation, severe hemorrhage, and malignant changes associated with the presence of ectopic gastric mucosa.”

Fig. 4.2: Sample Text from PubMed

Summary:

Enteric Duplications (EDs) are uncommon anomalies that can occur at any point of the gastrointestinal tract . The small intestine is the most common location ; retroperitoneum is an extremely rare site . In general , diagnosed in the neonatal period or during infancy , they are increasingly diagnosed prenatally ; early prenatal detection is possible. There have been seven reported cases of retroperitoneal ED cyst in the english literature.

Fig. 4.3: Sample Summary for text in Figure 4.2

Low Relevance Sentences containing KeyPhrases:

The small intestine is the most common location ; retroperitoneum is an extremely rare site.

Prenatal ultrasound at 22 weeks of gestation objectified an abdominal cystic mass located in the left upper abdominal quadrant associated with fetal pyelectasis.

Postnatal ultrasound found a retro peritoneal para - aortic liquid - filled mass measuring 60 mm 33 mm 22 mm.

Magnetic resonance imaging (MRI) confirmed the presence of a retro peritoneal cyst occupying the upper left retro peritoneal space ; with mass effect displacing the left kidney down [figure 1].

Per operative finding was a retro peritoneal cyst above the left adrenal rarely an echo genic mass is found due to complication (hemorrhage).

Fig. 4.4: Low relevance sentences containing Key-phrases for text in Figure 4.2

Explanation:
The summary excludes details about specific diagnostic findings, such as the prenatal ultrasound showing a cystic mass in the upper left abdominal quadrant, its postnatal size and location, MRI results, and preoperative findings. These facts were excluded as they focus on a detailed case presentation and measurements, which, while useful for clinical insights, are not central to the broader context of ED anomalies. The summary focuses on the rarity, primary locations, and general diagnostic trends, maintaining a high-level overview of EDs rather than delving into specific case data.

Fig. 4.5: Sample explanation for text in Figure 4.2

TABLE 4.1: ROUGE-1 Scores comparing variants of the proposed method with the benchmark using the PubMed Dataset and ChatGPT-4o-latest

Feature Attribution Method	ROUGE-1 Score	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
BigBirdPEGASUS						
GI	Precision	**0.5392**	0.5182	0.5236	0.5111	0.5206
	Recall	**0.1187**	0.1118	0.1149	0.1112	0.1149
	F1	**0.1919**	0.1813	0.1854	0.1801	0.1852
Attention	Precision	**0.5661**	0.5182	0.5274	0.5126	0.5206
	Recall	**0.1199**	0.1118	0.1175	0.1147	0.1149
	F1	**0.1951**	0.1813	0.1887	0.1840	0.1852
LongT5						
GI	Precision	0.5276	0.5121	**0.5352**	0.4945	0.5228
	Recall	0.1013	0.1013	**0.1066**	0.0917	0.1051
	F1	0.1669	0.1661	**0.1746**	0.1526	0.1725
Attention	Precision	**0.5553**	0.5345	0.535	0.4906	0.5228
	Recall	0.1039	0.1055	**0.108**	0.0906	0.1051
	F1	0.1726	0.1733	**0.1773**	0.1511	0.1725

TABLE 4.2: BertScores comparing variants of the proposed method with the benchmark using the PubMed Dataset and ChatGPT-4o-latest

Feature Attribution Method	BertScore	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
BigBirdPEGASUS						
GI	Precision	**0.774**	0.7723	0.7738	0.7725	0.7722
	Recall	**0.8336**	0.8291	0.8296	0.8292	0.83
	F1	**0.8027**	0.7996	0.8007	0.7998	0.8
Attention	Precision	**0.7741**	0.7723	0.7738	0.7717	0.7722
	Recall	**0.8353**	0.8291	0.8303	0.8284	0.83
	F1	**0.8035**	0.7996	0.801	0.799	0.8
LongT5						
GI	Precision	0.7697	0.7705	**0.7713**	0.767	0.7705
	Recall	0.8316	0.8308	**0.8322**	0.8255	0.8274
	F1	0.7994	0.7995	**0.8006**	0.7952	0.7979
Attention	Precision	**0.7722**	0.7703	0.7717	0.7657	0.7705
	Recall	**0.835**	0.8302	0.8322	0.8261	0.8274
	F1	**0.8023**	0.7991	0.8008	0.7947	0.7979

TABLE 4.3: ROUGE-1 Scores comparing variants of the proposed method with the benchmark using the Arxiv Dataset and ChatGPT-4o-latest

Feature Attribution Method	ROUGE-1 Score	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
BigBirdPEGASUS						
GI	Precision	**0.5658**	0.5349	0.5653	0.5193	0.5522
	Recall	0.1257	0.1215	0.1285	0.1123	**0.1287**
	F1	0.2017	0.1938	**0.2066**	0.1809	0.2045
Attention	Precision	**0.5677**	0.5286	0.5577	0.5345	0.5496
	Recall	0.1195	0.1192	**0.1283**	0.1123	0.1253
	F1	0.1943	0.1911	**0.2052**	0.1824	0.2004
LongT5						
GI	Precision	0.5507	0.5411	**0.5584**	0.5394	0.5438
	Recall	0.1168	0.1204	**0.1262**	0.1132	0.1226
	F1	0.1898	0.1932	**0.2016**	0.1836	0.1967
Attention	Precision	**0.5633**	0.5495	0.5502	0.5364	0.5504
	Recall	0.1189	0.1206	0.1223	0.1128	**0.1287**
	F1	0.1921	0.194	0.1969	0.1827	**0.2043**

TABLE 4.4: BertScores comparing variants of the proposed method with the benchmark using the Arxiv Dataset and ChatGPT-4o-latest

Feature Attribution Method	BertScore	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
BigBirdPEGASUS						
GI	Precision	0.7709	0.7706	**0.7718**	0.7676	0.7714
	Recall	**0.8355**	0.8317	0.8348	0.8316	0.8332
	F1	0.8019	0.7999	**0.802**	0.7983	0.8011
Attention	Precision	0.7681	0.7692	**0.7709**	0.7676	0.7702
	Recall	**0.8356**	0.8309	0.8346	0.832	0.8309
	F1	0.8004	0.7988	**0.8014**	0.7985	0.7994
LongT5						
GI	Precision	0.7685	0.7693	0.7701	0.7685	**0.7702**
	Recall	0.8343	0.8323	**0.8353**	0.8322	0.8297
	F1	0.8	0.7995	**0.8014**	0.7991	0.7988
Attention	Precision	0.7684	0.7693	0.7698	0.768	**0.7708**
	Recall	**0.836**	0.8326	0.8347	0.832	0.831
	F1	0.8008	0.7997	**0.8009**	0.7987	0.7997

TABLE 4.5: ROUGE-1 Scores comparing variants of the proposed method with the benchmark using the Arxiv Dataset and Claude-3-Haiku

Feature Attribution Method	ROUGE-1 Score	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
BigBirdPEGASUS						
GI	Precision	0.6581	0.6644	0.6275	**0.6704**	0.663
	Recall	0.2085	0.222	0.1865	0.2111	**0.2386**
	F1	0.311	0.3265	0.2824	0.3156	**0.3469**
Attention	Precision	0.6357	0.6389	0.6192	0.6512	**0.6573**
	Recall	0.1737	0.211	0.1699	0.2012	**0.2434**
	F1	0.269	0.3122	0.2627	0.3006	**0.3488**
LongT5						
GI	Precision	0.6451	0.6432	0.6158	**0.6486**	0.6384
	Recall	0.2112	0.2051	0.1814	0.2157	**0.235**
	F1	0.3128	0.3047	0.2755	0.3178	**0.3365**
Attention	Precision	0.6468	**0.6491**	0.6184	0.64	0.6456
	Recall	0.1819	0.2104	0.1743	0.1995	**0.2312**
	F1	0.2799	0.3114	0.2667	0.2981	**0.3347**

TABLE 4.6: BertScores comparing variants of the proposed method with the benchmark using the Arxiv Dataset and Claude-3-Haiku

Feature Attribution Method	BertScore	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
BigBirdPEGASUS						
GI	Precision	**0.7942**	0.7832	0.7787	0.7833	0.7926
	Recall	0.8542	0.8524	0.8497	**0.8562**	0.845
	F1	0.8119	0.8163	0.8126	**0.8181**	0.8122
Attention	Precision	**0.7821**	0.7819	0.7744	0.78	0.7798
	Recall	**0.8541**	0.8513	0.847	0.8506	0.8479
	F1	0.8132	**0.8151**	0.809	0.8138	0.8078
LongT5						
GI	Precision	**0.796**	0.779	0.7755	0.7801	0.789
	Recall	**0.8603**	0.8494	0.847	0.8514	0.8318
	F1	0.8098	0.8126	0.8096	**0.8141**	0.8109
Attention	Precision	0.7794	0.7796	0.7744	0.7778	**0.7861**
	Recall	**0.8615**	0.8493	0.8475	0.8479	0.8437
	F1	**0.816**	0.8129	0.8092	0.8113	0.8094

Let us first analyze the results of BigBirdPegasus for the PubMed dataset, with explanations being generated using ChatGPT-4o. Table 4.1 and Table 4.2 show that the highest ROUGE-1 precision, recall, and F1, and the highest BertScore precision, recall, and F1 were obtained by method (1) described in the FOE algorithm. Method (4) has consistently performed worse than the benchmark in almost all metrics (except Bertscore precision using the GI method). Method (1) performing best can be attributed to the prompt providing only the necessary information to generate an explanation. Method (2) does not contain any insights into the summarization model workings, hence performing worse than Method (1). This reflects the importance of using an insight into the summarization process (using a feature attribution method) to obtain a better explanation from the LLM. Method (3) seems to have redundant information compared to (1), KPDiff, which slightly worsens the performance. Method (4) overwhelms the LLM and results in the worst performance by providing the LLM with excessive unnecessary information, that is, all the low-relevance sentences in D unfiltered. Given the substantial length of the input documents, it was essential to design a concise yet informative prompt. Method (1) achieves a good tradeoff between prompt length and informativeness, ensuring that the LLM can effectively process the input while maintaining explanation quality.

Looking at the results for LongT5 for the PubMed dataset in Table 4.1 and Table 4.2, it can be observed that Method (3) has performed better in certain metrics when compared with Method (1). This can be attributed to the difference in the internal working of BigBirdPegasus and LongT5. BigBirdPegasus uses sparse attention, which focuses more on local patterns in the input. This causes its summaries to omit entire sentences that are not relevant to the main focus, even if they contain important keyphrases. In contrast, LongT5 utilizes global-local attention and tends to produce more abstract summaries. While retaining the sentence-level structure, it might omit specific keyphrases that are semantically important but not important to the main theme. This causes Method (1) to be effective for BigBirdPegasus and Method (3) to be effective for LongT5, considering that in the latter, an explicit list of omitted keyphrases is provided to the LLM to understand what was excluded and explain why it might have been omitted. Hence, the prompt structure for the LLM may need to be slightly tweaked based on the transformer model used for AS.

As shown in Table 4.3 and Table 4.4, for the Arxiv dataset, when generating explanations using ChatGPT-4o, Method (3) emerged as the preferred approach across both AS models, with Method (1) following closely, differing by only small margins. When repeating the same experiment using Claude-3-Haiku as the LLM, it was observed that the BertScores (Table 4.6) have favoured method (1) in many cases, while favouring method (4) marginally in some other cases. However, most of the ROUGE scores, as shown in Table 4.5, seem to have favoured the benchmark over FOE. This may be because the LLM Claude-3-Haiku may have had a tendency to use exact words

in the input text in its explanation when additional details are not provided. However, from the BertScores, which is a more nuanced mechanism to measure text similarity, it is evident that the FOE algorithm had been effective in generating explanations that capture omitted key facts even for the Arxiv dataset.

To explore whether the framework is successful in the context of short document summarization as well, several experiments were conducted using the XSum dataset [38], using the LongT5 model. Results are summarized in Table 4.7 and Table 4.8. While most of the metrics across the two feature attribution methods favour Method (1), margins are notably smaller than those observed in the PubMed experiments. Several metrics have favoured Method (2), and notably, three of the ROUGUE-1 metrics have favoured the benchmark over FOE. A possible explanation for these observations is that, as the XSum input documents are short, the LLM is capable of generating adequate explanations based solely on the input text and summary. The feature attribution-based insights provided by FOE have sometimes marginally contributed to improving the explanations. Even though this is the case for short document summarization, the utility of FOE for long document summarization is evident from the results reported for PubMed and Arxiv datasets.

Additionally, for both the transformer models, it is notable that the results obtained using Attention as the feature attribution method were slightly higher than the results achieved with GI. However, the GI method has significantly greater computational efficiency compared to Attention.

TABLE 4.7: ROUGE-1 Scores comparing variants of the proposed method with the benchmark using XSum Dataset and ChatGPT-4o-latest

Feature Attribution Method	ROUGE-1 Score	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
GI	Precision	0.5309	0.5349	0.5073	0.5201	**0.5459**
	Recall	0.1606	**0.171**	0.1554	0.1543	0.1649
	F1	0.225	0.2288	0.213	0.2138	**0.2289**
Attention	Precision	0.5384	0.5204	0.5176	0.5119	**0.5393**
	Recall	**0.1877**	0.1875	0.1799	0.1691	0.1829
	F1	**0.2508**	0.2432	0.2371	0.2314	0.245

Also, it can be observed at a glance that the scores are somewhat low in general, especially the ROUGE-1 scores. The reason for this is that both the ROUGE-1 and BERTScore evaluation metrics are inherently designed to evaluate summarization itself by comparing a candidate text with the reference text. However, when adapting those metrics to evaluate explanations in the proposed methodology, the explanation was compared with the set of sentences in the input document that contain key phrases. Hence, it is expected that the scores would be low in general, especially ROUGE-1

TABLE 4.8: BertScores comparing variants of the proposed method with the benchmark using XSum Dataset and ChatGPT-4o-latest

Feature Attribution Method	BertScore	(1) D + S + LRSKPD	(2) D + S + KPDiff	(3) D + S + LRSKPD + KPDiff	(4) D + S + LRS	Benchmark (D + S)
GI	Precision	**0.8193**	0.8169	0.8162	0.8164	0.817
	Recall	**0.8504**	0.8461	0.8481	0.8478	0.8479
	F1	**0.8345**	0.8311	0.8317	0.8317	0.8321
Attention	Precision	0.8155	**0.8169**	0.8137	0.8135	0.8163
	Recall	**0.8467**	0.8463	0.8437	0.8453	0.8459
	F1	0.8307	**0.8312**	0.8283	0.829	0.8308

scores, which compare the unigram overlap between the candidate and the reference. ROUGE-1 specifically looks for exact word matched while completely ignoring semantic similarities.

Results show that the relevance (precision) and completeness (recall) of the explanations of omitted facts in long documents can be improved using the proposed method. Using ROUGE-1 scores and BertScores, they have been evaluated with two mechanisms: unigram overlap and cosine similarity of contextual embeddings. This shows that providing the LLM with insights into the text selection process of the summarization model aids the LLM in generating an explanation that closely represents the AS model's reasoning while successfully capturing which key details were omitted from the summary and why. The proposed framework is especially successful in the context of long document summarization.

4.2 User Study

A user study was conducted with the participation of medical professionals as an expert evaluation of the summarization explanations generated by the FOE methodology. The study was conducted in the form of a questionnaire, which contained (a) five articles from the PubMed Dataset, (b) their respective summaries, and (c) two explanations on skipped facts, one generated with FOE and the other being the benchmark explanation. The participants rated the explanations on a scale of 1 to 5 in the following aspects:

1. Factually Consistent - How correct are the facts given?
2. Complete - Has the explanation captured key information that was skipped in the summary?
3. Relevant - How medically relevant is the information given in the explanation?
4. Coherent - How good is the flow of the explanation?

5. Overall Usefulness - If you had first seen the summary and then read this explanation, how useful is it for you to decide whether you should go back and read the whole article?

The results are summarized in Table 4.9.

TABLE 4.9: Results of the user study comparing the proposed method with the benchmark

Method	Factually Consistent	Complete	Relevant	Coherent	Useful	Average
Benchmark	3.1143	**3.0286**	3.1714	**3.2286**	3.2571	3.1600
FOE	**3.2571**	2.9714	3.1714	3.0857	**3.3714**	**3.1714**

As indicated by the average score in Table 4.9, explanations generated by FOE surpass the benchmark, indicating the significance of the proposed methodology. The overall usefulness of the FOE explanations as perceived by the medical professionals is higher than that of the benchmark explanations, indicating they would prefer FOE-based explanations in a practical setting. The factual consistency of the FOE explanations is also rated higher than that of the benchmark. Both explanations have been equally rated in terms of relevance. The benchmark has achieved a marginally higher score for completeness, even though automatic evaluation metrics in Section 4.1 indicate higher recall for FOE-based explanations. Participants have found that the benchmark explanations are more coherent than FOE explanations. This may be because when LLM generates the explanations in FOE, the set of LRS-KPD that could have been extracted from different parts of the document may affect the flow of the explanation. Therefore, there is room for improvement in coherence in FOE explanations.

CHAPTER 5

DISCUSSION

5.1 Study Contributions

In this research, an algorithm, FOE, is proposed that leverages feature attribution methods to gain insights into the inner workings of Transformer-based AS models and provides natural language explanations on what key facts were omitted and why. Given that omissions of key facts are particularly significant in the context of long document summarization, the experiments were focused on that area. The BigBirdPegasus model [55] and LongT5 model [49] were used as transformer-based AS models in the experiments, and the GPT-4 [52] model and Claude-3-Haiku [79] models were utilized as LLMs to generate the explanations. The experiments were conducted on the PubMed dataset [4] and Arxiv dataset [5], which comprises long research articles in medical and scientific domains. For comparison, several experiments were conducted for short document summarization as well, using the XSum [38] dataset.

The results for long document summarization show that the explanations thus generated are effective in terms of factual consistency, relevance, and completeness, and overall usefulness is practice as reflected in Table 4.1, Table 4.2, Table 4.3, Table 4.4, Table 4.6, and Table 4.9. However, there is room for improvement in terms of the coherence of the explanations, which will be investigated as part of future work. The results for short document summarization in Table 4.7 and Table 4.8 show that FOE marginally improves the explanations, while explanations generated even without the support of the FOE framework are also adequate in general. Hence, the utility of the proposed approach is higher for long document summarization, rather than short document summarization.

The research objectives highlighted in Section 1.2 have been achieved via this research as described below:

- To develop a framework that generates explanations for omitted facts in abstractive summaries produced by transformer-based AS models - In Section 3 we show that the proposed explainable AS algorithm based on a feature attribution method, FOE, is capable of providing factually consistent, relevant, and complete, and user-friendly explanations on fact omission during transformer-based AS.

- To evaluate the effectiveness of the generated explanations from the proposed framework, particularly in mission-critical domains - A user study with the participation of medical professionals was conducted as per Section 4.2, whose results show the practical utility and usability of the FOE framework. Automatic

evaluation of the explanations detailed in Section 4.1 also reflects the superiority of the FOE framework over the benchmark in explaining omitted facts.

5.2 Comparison with the existing studies

As described in the related studies in Section 2.4, none of the existing work has explicitly focused on explaining fact selection/omission of a transformer-based AS model in natural language, which is critical for applications requiring high transparency, such as legal or medical decision-making. Norkute et al. [9] derived "attention highlights" on the input text, rather than providing explanations in natural language. However, this approach requires the user to manually review the highlights and infer which key facts were omitted and the reasons for their exclusion. Baan et al. [73] attempted to uncover the behavior of individual attention heads in Transformer-based models by mapping them to interpretable features, enhancing transparency at the model level. However, this lacked direct actionable insights for downstream users. The SP framework introduced by Saha et al. [74] went a step further to model the generative process of summaries, as a mode of enhancing explainability. However, despite its transparency, the summaries produced in this method were less effective than state-of-the-art Transformer-based models such as PEGASUS. Additionally, this method also does not output explanations in natural language.

The proposed method addresses these limitations by offering feature attribution-based insights into which key facts were omitted from the summary and the reasons for their exclusion, all within a concise natural language explanation. Furthermore, this approach does not compromise the performance of the Transformer-based AS model in terms of quality, as this method merely “observes” the inner workings of the model, extracts insights from it, and uses those insights to guide an LLM to generate a useful explanation that is easily consumable by an end-user. The results imply that the explanations generated by FOE would be beneficial in practice in mission-critical domains such as medicine, where long document summarization is vital, in order for the users to trust the summary generated by the model while having an insight into the reasons behind any key fact omissions.

5.3 Open Challenges and Future Directions

A key open challenge in explainable AS is the absence of a standardized evaluation metric for explanations. Although ROUGE scores and BERTScore were employed to assess the effectiveness of the proposed method, these metrics are not specifically designed to evaluate the quality of natural language explanations. The ROUGUE Score, especially, may not reflect an intuitive evaluation of the generated summaries, as it looks for exact word matches while ignoring semantic similarities. While BertScore

overcomes this problem while looking for cosine similarities between embeddings, it is also still an evaluation metric designed to assess summarization itself. Therefore, developing a dedicated metric for evaluating explanations is an important avenue for future research.

In addition to the need for a dedicated evaluation metric, there is also a need to design datasets that include explanations for text summarization [81]. One approach to achieving this would be extending existing summarization datasets with explanations related to fact selection/omission and the reasons behind them.

In this study, for the long document summarization experiments, only input documents with less than 2000 words were used due to computational constraints. Future work should explore the framework's performance on longer documents, including full-length books. Moreover, evaluating the framework in multi-document summarization scenarios will also be beneficial in assessing its generalizability and robustness in varied summarization scenarios.

CHAPTER 6

CONCLUSION

6.1 Summary

This study proposed a novel algorithm, the Fact Omission Explanation (FOE) framework, designed to explain why transformer-based Abstractive Summarization (AS) models omit certain key facts during the summarization process. The framework leverages insights from feature attribution methods and incorporates a Large Language Model (LLM) to generate human-interpretable explanations that reflect the decision-making process of AS models.

The issue of key information omission is particularly prevalent in the summarization of long documents. To evaluate the effectiveness of the proposed framework, experiments were conducted using the PubMed dataset and Arxiv dataset, which contain research articles in the domains of medicine and science, and the BigBird-Pegasus model and LongT5 model, transformer architectures optimized for processing long input sequences. For comparison, experiments were conducted for short document summarization as well, using the XSum dataset. ChatGPT-4o and Claude-3-Haiku were the LLMs utilized for explanation generation.

This study achieves two main research objectives. First, it introduces the FOE framework, which generates user-friendly explanations that reveal the rationale behind the omission of key facts in generated summaries. Second, it evaluates the FOE framework, especially in a mission-critical domain, based on a user study with the participation of medical professionals and also using automatic evaluation metrics. Experimental results demonstrate that the FOE framework produces explanations that align with the internal reasoning of the model, while maintaining relevance and usability for end-users. Experiments also highlight the utility of the FOE framework in long document summarization, in effectively identifying and justifying key fact omissions.

6.2 Limitations

Due to computational resource limitations, experiments were limited to documents under 2000 words. Future investigations should consider evaluating the framework on longer texts, even books, to assess its scalability.

Moreover, this study's long document summarization experiments primarily focused on the medical and scientific domains. Evaluating the framework's applicability in other mission-critical domains, such as law and finance, would be valuable in understanding its broader utility.

6.3 Future Directions

Several avenues for future research remain open. Extending the framework to multi-document summarization tasks would provide valuable insights into its ability to generalize across complex and varied summarization scenarios.

Another important direction involves the development of dedicated evaluation metrics and benchmark datasets tailored specifically to the assessment of natural language explanations in abstractive summarization. These resources will enable a systematic and reliable evaluation of explanation quality, driving the progress in explainable summarization research.

REFERENCES

[1] P. H. Panawenna, K. G. Hettihewa, S. Wickramanayake, and D. Meedeniya, "Explainable artificial intelligence for building trustworthy transformer-based abstractive summarization models," in *Explainable Artificial Intelligence for Trustworthy Decisions in Smart Applications*. Springer Nature, 2025, ch. 10, pending publication.

[2] G. Wang and W. Wu, "Surveying the landscape of text summarization with deep learning: A comprehensive review," *Discrete Mathematics, Algorithms and Applications*, 2023.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[4] "ccdv/pubmed-summarization - datasets at hugging face," https://huggingface.co/datasets/ccdv/pubmed-summarization, accessed: 2025-4-15.

[5] "ccdv/arxiv-summarization - datasets at hugging face," https://huggingface.co/datasets/ccdv/arxiv-summarization, accessed: 2025-6-15.

[6] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A survey of the state of explainable AI for natural language processing," 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*, K.-F. Wong, K. Knight, and H. Wu, Eds. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 447–459. [Online]. Available: https://aclanthology.org/2020.aacl-main.46

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

[8] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[9] M. Norkute, N. Herger, L. Michalak, A. Mulder, and S. Gao, "Towards explainable ai: Assessing the usefulness and impact of added explainability features in

legal document summarization," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[10] A. Shukla, P. Bhattacharya, S. Poddar, R. Mukherjee, K. Ghosh, P. Goyal, and S. Ghosh, "Legal case document summarization: Extractive and abstractive methods and their evaluation," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 1048–1064.

[11] M. Wang, M. Wang, F. Yu, Y. Yang, J. Walker, and J. Mostafa, "A systematic review of automatic text summarization for biomedical literature and EHRs," *J. Am. Med. Inform. Assoc.*, vol. 28, no. 10, pp. 2287–2297, Sep. 2021.

[12] R. Jain, A. Jangra, S. Saha, and A. Jatowt, "A survey on medical document summarization," 2022.

[13] N. Zmandar, A. Singh, M. El-Haj, and P. Rayson, "Joint abstractive and extractive method for long financial document summarization," in *Proceedings of the 3rd Financial Narrative Processing Workshop*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Lancaster, United Kingdom: Association for Computational Linguistics, 15-16 Sep. 2021, pp. 99–105. [Online]. Available: https://aclanthology.org/2021.fnp-1.19/

[14] N. Foroutan, A. Romanou, S. Massonnet, R. Lebret, and K. Aberer, "Multilingual text summarization on financial documents," in *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, M. El-Haj, P. Rayson, and N. Zmandar, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 53–58. [Online]. Available: https://aclanthology.org/2022.fnp-1.7/

[15] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, and D. R. I. M. Setiadi, "Review of automatic text summarization techniques & methods," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, pp. 1029–1046, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:219504970

[16] D. Parveen, H.-M. Ramsl, and M. Strube, "Topical coherence for graph-based extractive summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1949–1954. [Online]. Available: https://aclanthology.org/D15-1226

[17] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," 2016.

[18] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," 2018.

[19] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, D. Lin and D. Wu, Eds. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 404–411. [Online]. Available: https://aclanthology.org/W04-3252

[20] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, p. 457–479, Dec. 2004. [Online]. Available: http://dx.doi.org/10.1613/jair.1523

[21] T. Cohn and M. Lapata, "Sentence compression beyond word deletion." 01 2008, pp. 137–144.

[22] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1095–1104. [Online]. Available: https://aclanthology.org/P17-1101

[23] H. Lin and V. Ng, "Abstractive summarization: A survey of the state of the art," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 9815–9822, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5056

[24] L. Bing, P. Li, Y. Liao, W. Lam, W. Guo, and R. J. Passonneau, "Abstractive multi-document summarization via phrase selection and merging," 2015.

[25] Y. Mehdad, G. Carenini, and R. T. Ng, "Abstractive summarization of spoken and written conversations based on phrasal queries," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Toutanova and H. Wu, Eds. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1220–1230. [Online]. Available: https://aclanthology.org/P14-1115

[26] P.-E. Genest and G. Lapalme, "Fully abstractive approach to guided summarization," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, H. Li, C.-Y. Lin,

M. Osborne, G. G. Lee, and J. C. Park, Eds. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 354–358. [Online]. Available: https://aclanthology.org/P12-2069

[27] G. Murray, G. Carenini, and R. Ng, "Generating and validating abstracts of meeting conversations: a user study," in *Proceedings of the 6th International Natural Language Generation Conference*, J. Kelleher, B. M. Namee, and I. v. d. Sluis, Eds. Association for Computational Linguistics, Jul. 2010. [Online]. Available: https://aclanthology.org/W10-4211

[28] A. Gatt and E. Reiter, "SimpleNLG: A realisation engine for practical applications," in *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, E. Krahmer and M. Theune, Eds. Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 90–93. [Online]. Available: https://aclanthology.org/W09-0613

[29] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward abstractive summarization using semantic representations," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, R. Mihalcea, J. Chai, and A. Sarkar, Eds. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 1077–1086. [Online]. Available: https://aclanthology.org/N15-1114

[30] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[31] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," 2015.

[32] Q. Wang and J. Ren, "Summary-aware attention for social media short text abstractive summarization," *Neurocomputing*, vol. 425, pp. 290–299, Feb. 2021.

[33] Z. Liang, J. Du, and C. Li, "Abstractive social media text summarization using selective reinforced Seq2Seq attention model," *Neurocomputing*, vol. 410, pp. 432–440, Oct. 2020.

[34] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, S. Riezler and Y. Goldberg, Eds. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. [Online]. Available: https://aclanthology.org/K16-1028

[35] A. See, P. J. Liu, and C. 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)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017.

[36] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018.

[37] W. Li, X. Xiao, Y. Lyu, and Y. Wang, "Improving neural abstractive document summarization with explicit information selection modeling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018.

[38] S. Narayan, S. B. Cohen, and M. 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*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 1797–1807. [Online]. Available: https://aclanthology.org/D18-1206

[39] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," 2018.

[40] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," 2018.

[41] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," 2019.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[43] W. Li, X. Xiao, Y. Lyu, and Y. Wang, "Improving neural abstractive document summarization with structural regularization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018.

[44] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," 2019.

[45] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," 2019.

[46] X. Zhang, F. Wei, and M. 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*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5059–5069. [Online]. Available: https://aclanthology.org/P19-1499

[47] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

[48] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020.

[49] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang, "LongT5: Efficient text-to-text transformer for long sequences," in *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 724–736. [Online]. Available: https://aclanthology.org/2022.findings-naacl.55/

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

[51] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, "Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training," 2020. [Online]. Available: https://arxiv.org/abs/2001.04063

[52] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[53] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," 2020.

[54] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big bird: transformers for longer sequences," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[55] "BigBirdPegasus," https://huggingface.co/docs/transformers/en/model_doc/bigbird_pegasus, accessed: 2025-4-15.

[56] Y. Liu and P. Liu, "SimCLS: A simple framework for contrastive learning of abstractive summarization," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2021.

[57] Y. Liu, P. Liu, D. Radev, and G. Neubig, "BRIO: Bringing order to abstractive summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022.

[58] M. Ravaut, S. Joty, and N. 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)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022.

[59] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," 2018.

[60] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, M. R. Costa-jussà and E. Alfonseca, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 37–42. [Online]. Available: https://aclanthology.org/P19-3007

[61] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas, and M. Wattenberg, "Attentionviz: A global view of transformer attention," 2023.

[62] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017.

[63] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important FeaTures through propagating activation differences," 2016.

[64] Z. Han, M. Shang, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. L. P. Chen, "SeqViews2SeqLabels: Learning 3D global features via aggregating se-

quential views by RNN with attention," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 658–672, Sep. 2018.

[65] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 697–10 706.

[66] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1245–1256, Jun. 2017.

[67] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2Tensor for neural machine translation," in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, C. Cherry and G. Neubig, Eds. Boston, MA: Association for Machine Translation in the Americas, Mar. 2018, pp. 193–199. [Online]. Available: https://aclanthology.org/W18-1819

[68] S. Liu, T. Li, Z. Li, V. Srikumar, V. Pascucci, and P.-T. Bremer, "Visual interrogation of attention-based models for natural language inference and machine comprehension," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, E. Blanco and W. Lu, Eds. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 36–41. [Online]. Available: https://aclanthology.org/D18-2007

[69] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," 2020.

[70] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," 2021.

[71] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace, "Eraser: A benchmark to evaluate rationalized nlp models," 2020.

[72] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," 2021.

[73] J. Baan, M. ter Hoeve, M. van der Wees, A. Schuth, and M. de Rijke, "Do transformer attention heads provide transparency in abstractive summarization?" *arXiv preprint arXiv:1907.00570*, 2019.

[74] S. Saha, S. Zhang, P. Hase, and M. Bansal, "Summarization programs: Interpretable abstractive summarization with neural modular trees," in *ICLR*, 2023.

[75] P. H. Panawenna and S. Wickramanayake, "Understanding omitted facts in transformer-based abstractive summarization," in *2024 Moratuwa Engineering Research Conference (MERCon)*. IEEE, Aug. 2024, pp. 624–629.

[76] M. P. Grootendorst, "KeyBERT," https://maartengr.github.io/KeyBERT/index.html, accessed: 2025-4-15.

[77] "Google/bigbird-pegasus-large-pubmed - hugging face," https://huggingface.co/google/bigbird-pegasus-large-pubmed, accessed: 2025-4-15.

[78] "thanhkt/long-t5-tglobal-base-16384-book-summary-finetuned-pubmed - hugging face," https://huggingface.co/thanhkt/long-t5-tglobal-base-16384-book-summary-finetuned-PubMed, accessed: 2025-4-15.

[79] "The claude 3 model family: Opus, sonnet, haiku." [Online]. Available: https://api.semanticscholar.org/CorpusID:268232499

[80] I. D. Ayadi, A. Bezzine, E. B. Hamida, and Z. Marrakchi, "Fetal cyst reveling retroperitoneal enteric duplication," *J Indian Assoc Pediatr Surg*, vol. 22, no. 1, pp. 60–61, Jan. 2017.

[81] M. Dhaini, E. Erdogan, S. Bakshi, and G. Kasneci, "Explainability meets text summarization: A survey," in *International Conference on Natural Language Generation*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272602332