Biomedical Lay Summarization Using Pre-Trained Adapters

by

V. S. Sandeep Reddy Dwarampudi
Bachelor of Technology, Manipal Institute of Technology, Manipal, India, 2021

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2024

Chicago, Illinois

Defense Committee:
Prof. Shweta Yadav, Chair and Advisor
Prof. Sourav Medya
Prof. Cornelia Caragea

Copyright by

V. S. Sandeep Reddy Dwarampudi

2024

To Almighty, source of all creation and knowledge, the guiding light of truth and wisdom.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to those who have supported me throughout my journey in completing this thesis.

First and foremost, I am deeply grateful to my advisor, Prof. Shweta Yadav. Her guidance, expertise, and encouragement were instrumental in shaping this thesis. Her insightful feedback and willingness to dedicate her time throughout the research process were invaluable.

I would also like to extend my heartfelt thanks to my parents. Their unwavering love, support, and belief in me provided the foundation that allowed me to pursue my academic goals.

DVS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

KEGG	Kyoto Encyclopedia of Genes and Genomes
LLMs	Large Language Models
MLM	Masked Language Modeling
MSI	Multi-Scale Interactome
NER	Named Entity Recognition
NLP	Natural Language Processing
PLABA	Plain Language Adaptation of Biomedical Abstracts
PLMs	Pre-trained Language Models
PLOS	Public Library of Science
TF-IDF	Term Frequency-Inverse Document Frequency
UMLS	Unified Medical Language System

SUMMARY

There is growing interest among the general public in accessing biomedical literature to find treatments and causes of common health problems, or to read about significant global topics like disease outbreaks. However, the technical language and complex concepts can be difficult to understand for those without a background in the field. Our goal is to build an efficient model that can simplify complex biomedical text, making it easier for the general public to grasp the main points while preserving the original intent of the research. Additionally, providing background information further enhances the overall comprehensibility of the summary and ensures a coherent representation of the key points. To achieve this, we introduce custom adapter blocks into the transformer architecture that leverage domain-specific knowledge to enhance the encoded information in the language models. We conducted experiments using different pre-training techniques and compared the results with pre-trained language models (PLMs) and large language models (LLMs). Our extensive experiments on two benchmark datasets show that our proposed method improves over the pre-trained language models in terms of readability-based automatic metrics and human evaluations.

CHAPTER 1

INTRODUCTION

The COVID-19 pandemic has greatly heightened public interest in biomedical research (1), underscoring its vital role in understanding global health and disease dynamics. Insights and discoveries from this field are essential not only for scientists and medical professionals but also for journalists and the general public. The biomedical field is rapidly expanding, with over 3,500 new documents added daily to various journals (2). Navigating this vast knowledge base is challenging and time-consuming, especially given the extensive length of these articles. This situation calls for effective text summarization to manage the information overload.

Text summarization condenses lengthy texts while preserving their key information, making documents more comprehensible. This process involves thoroughly analyzing and processing extensive texts to distill their essence, enhancing readability and understanding without compromising the document's overall meaning and significance. There are two primary methods for text summarization: abstractive and extractive summarization (3).

Abstractive summarization generates a brief summary that may include new words, phrases, or sentences not found in the original text. This method depends on comprehending the context and creating natural, human-like language to express the main ideas. Conversely, extractive summarization focuses on identifying and extracting key sentences or phrases directly from the source text to create the summary, without rewording or producing new sentences (4).

In the biomedical field, text summarization is critical for experts such as researchers, medical professionals, and scientists, as it condenses complex biomedical texts while retaining technical terminology and specialized concepts (5). It provides a concise overview of research findings, methodologies, and conclusions for those already well-versed in the subject matter.

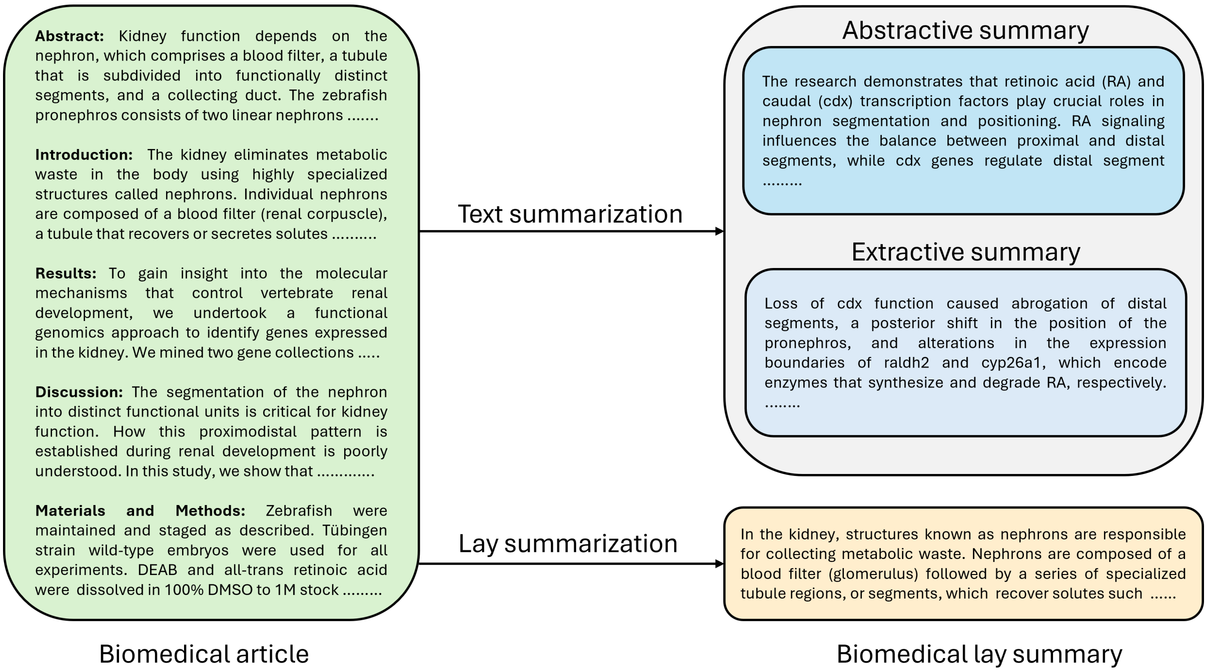
However, there is also a need for biomedical lay summarization, which makes biomedical texts accessible to non-experts by simplifying language, explaining technical concepts, and adding necessary background information. It ensures that individuals without specialized knowledge can understand the content. Due to the nature of the task, this work focuses on abstractive lay summarization, aiming to make biomedical research more accessible to a broader audience.

Biomedical lay summarization transforms complex technical language and concepts into understandable terms while preserving the original intent and significance of the research (6) as shown in Figure 1. By providing context and background knowledge, these summaries enhance clarity and coherence, ensuring that key points are effectively communicated to a broader audience.

Let $\mathcal{C}$ be a biomedical corpus composed of $\mathcal{D}$ documents. Each document $d \in \mathcal{C}$ contains m sentences, represented as $d = \{s_1, \ldots, s_m\}$. The reference (lay) summary for document d is denoted by $t_d = \{t_d^1, t_d^2, \ldots, t_d^n\}$, where n represents the number of words in the summary.

The model aims to maximize the log-likelihood of the target words in the reference summary:

$$\log p(t \mid \mathcal{C}; \theta) = \sum_{d \in \mathcal{C}} \sum_{i=1}^{n} \log p(t_d^i \mid t_d^{<i}, d; \theta)$$



Abstract: Kidney function depends on the nephron, which comprises a blood filter, a tubule that is subdivided into functionally distinct segments, and a collecting duct. The zebrafish pronephros consists of two linear nephrons ....

Introduction: The kidney eliminates metabolic waste in the body using highly specialized structures called nephrons. Individual nephrons are composed of a blood filter (renal corpuscle), a tubule that recovers or secretes solutes ....

Results: To gain insight into the molecular mechanisms that control vertebrate renal development, we undertook a functional genomics approach to identify genes expressed in the kidney. We mined two gene collections .....

Discussion: The segmentation of the nephron into distinct functional units is critical for kidney function. How this proximodistal pattern is established during renal development is poorly understood. In this study, we show that ....

Materials and Methods: Zebrafish were maintained and staged as described. Tübingen strain wild-type embryos were used for all experiments. DEAB and all-trans retinoic acid were dissolved in 100% DMSO to 1M stock ....

The research demonstrates that retinoic acid (RA) and caudal (cdx) transcription factors play crucial roles in nephron segmentation and positioning. RA signaling influences the balance between proximal and distal segments, while cdx genes regulate distal segment ....

Loss of cdx function caused abrogation of distal segments, a posterior shift in the position of the pronephros, and alterations in the expression boundaries of raldh2 and cyp26a1, which encode enzymes that synthesize and degrade RA, respectively. ....

In the kidney, structures known as nephrons are responsible for collecting metabolic waste. Nephrons are composed of a blood filter (glomerulus) followed by a series of specialized tubule regions, or segments, which recover solutes such ......

Figure 1: Types of text summarization.

Here, t_d^i represents the i-th word in the reference summary t_d for document d, and it holds that $n \ll m$ (3).

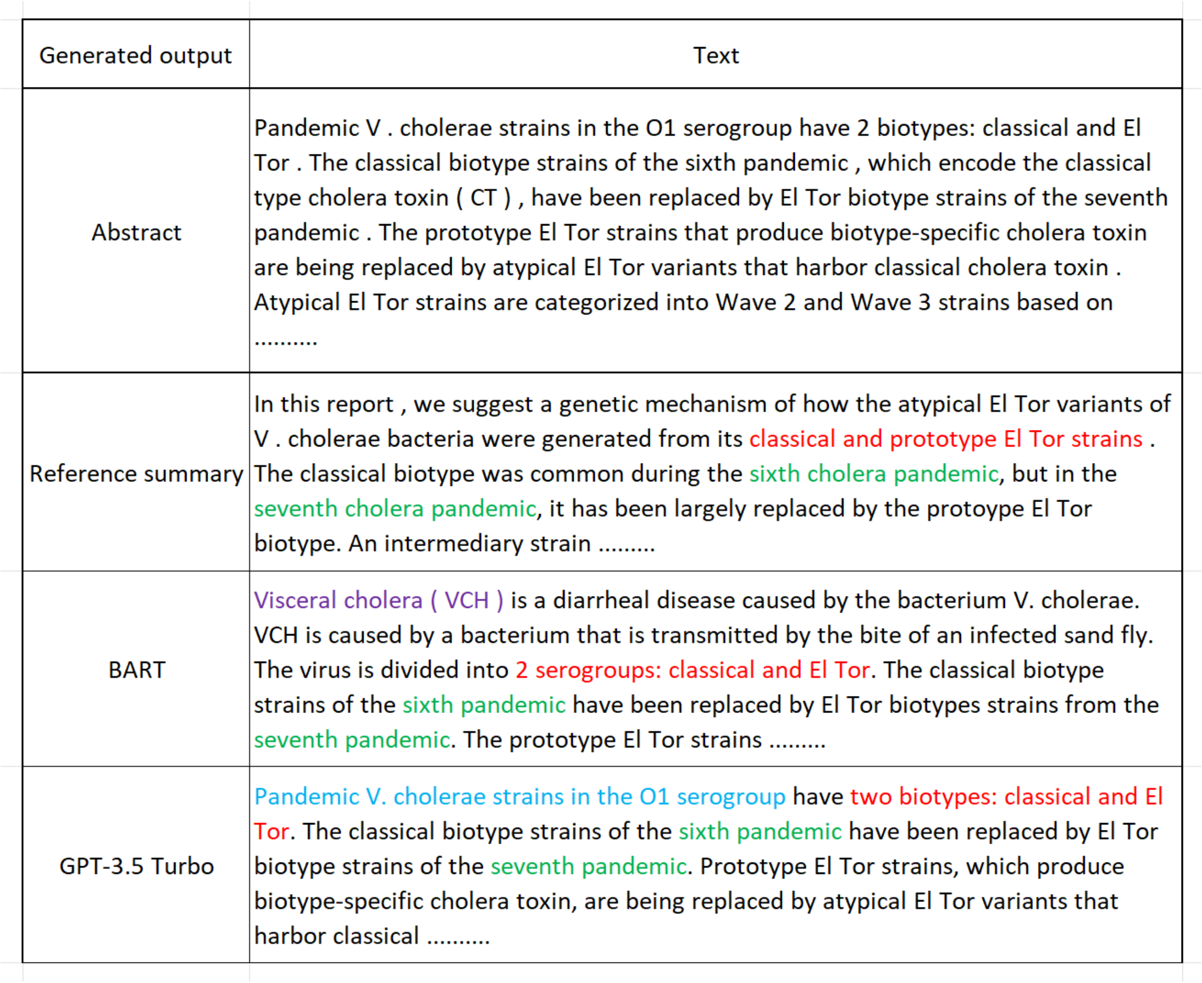
The field of automatic text summarization has witnessed significant advancements recently, driven by the development of core Natural Language Processing (NLP) technologies, including Pre-trained Language Models (PLMs) and Large Language Models (LLMs) (7; 8). PLMss are language models trained on extensive amounts of unlabeled data using self-supervised learning techniques, enabling them to grasp a certain degree of common sense and lexical knowledge embedded within the training data (9). Leveraging this knowledge, PLMss have substantially improved the performance of various NLP tasks through fine-tuning (10). Researchers have further boosted the capabilities of PLMss by enlarging both the model size and the volume of training data, leading to the advent of LLMss (11). LLMss exhibit exceptional proficiency in understanding and generating natural language (8). Additionally, LLMss demonstrate an emergent capability known as in-context learning, which allows them to execute a range of tasks by simply following natural language prompts, without requiring supervised training (8). For example, OpenAI created a Large Language Model called GPT-4 which has demonstrated human-like capabilities in zero-shot scenarios across multiple domains such as computer vision, programming, mathematics, and healthcare (3).

Despite advancements in general English automatic text summarization driven by large language models (LLMs) (12), progress in the biomedical domain has been limited (3). Challenges include knowledge grounding, establishing correct relationships between entities, and discerning between abbreviations, synonyms, homographs, and hyponyms specific to the biomedical domain. In Figure 2, we provide an example comparing the gold lay summary and lay summaries generated by an LLM and a PLM. The BART model incorrectly identifies the entity, referring to "*V. cholerae*" as "*Visceral cholera*" instead of its correct term, "*Vibrio cholerae*". The GPT-3.5 Turbo model complicates the opening sentence. Both the BART and GPT-3.5 Turbo models fail to adequately address the *"prototype El Tor strains"* and do not clearly mention the "*cholera pandemic*".

Motivated by these findings, in this thesis, we hypothesize that incorporating biomedical knowledge in the transformer block can overcome these shortcomings. Towards this, we propose a method called "Biomedical lay summarization using pre-trained adapters", which employs three adapter modules into PLM for biomedical lay summarization. Each adapter is pre-trained to address specific challenges:

- Adapter 1: We hypothesize that pre-training on masked-language modeling will improve knowledge grounding and handle homographs. This adapter is designed to understand the context and predict the missing parts of the text, thereby improving the model's capability to understand the underlying biomedical knowledge and correctly interpret words with multiple meanings based on context.
- Adapter 2: We hypothesize that pre-training on entity relations will address the challenge of establishing correct relationships between entities. By focusing on how different biomedical entities interact, this adapter aims to ensure that the summarization process maintains accurate and meaningful connections within the text.
- Adapter 3: We hypothesize that pre-training on entity synonyms will handle the issue of discerning between abbreviations, synonyms, and hyponyms. This adapter is intended to help the model recognize and correctly interpret various terms that might be used interchangeably or in hierarchical structures in biomedical literature, ensuring clarity and consistency in the summarized text.

We utilized the Plain Language Adaptation of Biomedical Abstracts (PLABA) and Public Library of Science (PLOS) datasets to evaluate the effectiveness of our models. During the pre-training phases, we incorporated the Unified Medical Language System (UMLS) knowledge base and the Multi-Scale Interactome (MSI) knowledge graph. The results obtained from our best model demonstrate that integrating adapters enhances the performance of models on biomedical lay summarization. We further conducted human evaluation to validate the improvements observed in the automatic metrics and to ensure that the generated summaries are understandable by laypeople.



Generated output	Text
Abstract	Pandemic V . cholerae strains in the O1 serogroup have 2 biotypes: classical and El Tor . The classical biotype strains of the sixth pandemic , which encode the classical type cholera toxin (CT) , have been replaced by El Tor biotype strains of the seventh pandemic . The prototype El Tor strains that produce biotype-specific cholera toxin are being replaced by atypical El Tor variants that harbor classical cholera toxin . Atypical El Tor strains are categorized into Wave 2 and Wave 3 strains based on ....
Reference summary	In this report , we suggest a genetic mechanism of how the atypical El Tor variants of V . cholerae bacteria were generated from its classical and prototype El Tor strains . The classical biotype was common during the sixth cholera pandemic, but in the seventh cholera pandemic, it has been largely replaced by the protoype El Tor biotype. An intermediary strain ....
BART	Visceral cholera (VCH) is a diarrheal disease caused by the bacterium V. cholerae. VCH is caused by a bacterium that is transmitted by the bite of an infected sand fly. The virus is divided into 2 serogroups: classical and El Tor. The classical biotype strains of the sixth pandemic have been replaced by El Tor biotypes strains from the seventh pandemic. The prototype El Tor strains ....
GPT-3.5 Turbo	Pandemic V. cholerae strains in the O1 serogroup have two biotypes: classical and El Tor. The classical biotype strains of the sixth pandemic have been replaced by El Tor biotype strains of the seventh pandemic. Prototype El Tor strains, which produce biotype-specific cholera toxin, are being replaced by atypical El Tor variants that harbor classical ....

Figure 2: Comparing the errors made by GPT-3.5 and BART.
🟥 Red: Denotes incompleteness
🟩 Green: Represents lack of references
🟦 Blue: Highlights less readability
🟪 Purple: Indicates incorrect abbreviations

CHAPTER 2

RELATED WORK

This section will discuss the existing works in open-domain text summarization, biomedical domain text summarization, and biomedical lay summarization.

2.1 Text summmarization

As discussed in Section 1, there are two main types of text summarization: extractive and abstractive. Extractive summarization involves selecting and using existing sentences or phrases from the original text to create a summary, ensuring that the summary consists of the most critical parts of the original document (13). Abstractive summarization, on the other hand, generates new sentences that convey the main ideas, potentially using different words and phrasing than the original text (14). This method often results in more coherent and human-like summaries by interpreting and reformulating the content.

Significant progress has been made in abstractive summarization through the use of advanced neural network models in recent years. One of the earliest neural network models for this task was the Seq2seq (sequence to sequence) model with attention mechanisms (15), which was originally developed for machine translation. This model architecture enabled the generation of more fluent and coherent summaries. The Pointer-Generator Network (16) further enhanced this approach by allowing the model to copy words from the source text while generating new phrases (17). PLMs such as BERT (7), BART (18), and T5 (19) have also been adapted for abstractive summarization, achieving state-of-the-art performance. For instance, PEGASUS (20) employs a transformer-based approach specifically designed for generating abstractive summaries by masking and predicting important sentences within a document.

Extractive summarization has also seen advancements with the advent of neural network models. Traditional methods, such as Term Frequency-Inverse Document Frequency (TF-IDF) (21), TextRank (22), and LexRank (23), laid the foundation for identifying and ranking important sentences. These methods relied on statistical and graph-based techniques to determine sentence importance. More recently, neural architectures have been employed to frame extractive summarization as a binary classification problem (24), where each sentence is evaluated for its relevance to the summary. Models like BERTSUM (25) utilize a document-level encoder based on BERT to generate extractive summaries. By leveraging the contextual information from pre-trained language models, these approaches can more effectively filter and select salient sentences from the source material, resulting in concise and informative summaries.

2.2 Biomedical text summarization

In the realm of abstractive summarization for biomedical texts, significant advancements have been made through various approaches. Sotudeh et al. (26) improved the summarization of radiology reports by incorporating medical ontology into a Seq2Seq model, enhancing the relevance and coherence of the generated summaries and Wallace et al. (27) explored the BART model using domain-specific pre-training techniques and input enhancements for summarizing multiple documents from randomized controlled trials (RCTs), (28)demonstrating the model's ability to handle complex biomedical texts . The availability of in-domain corpora has also spurred progress in this field. Cohan et al. (29) and Wang et al. (30) compiled extensive biomedical literature datasets with abstracts serving as summaries, providing valuable resources for training and evaluating summarization models. DeYoung et al. (31) investigated summarizing systematic reviews from their cited clinical trials, highlighting the potential of abstractive methods in distilling essential information from comprehensive biomedical studies. Guo et al. (32) used the PubMed dataset to pre-train BERT and BART models and subsequently fine-tuned them to create lay language summaries of biomedical reviews (33), bridging the gap between technical content and general understanding.

In contrast, extractive summarization in the biomedical domain has focused on leveraging pre-trained language models and domain-specific knowledge. Du et al. (34) proposed fine-tuning the BioBERT model for biomedical extractive summarization, achieving notable performance improvements by utilizing domain-specific embeddings. Xie et al. (35) incorporated medical knowledge from PICO (Population, Intervention, Comparison, Outcome) into pre-trained language models for extractive summarization of biomedical literature, enhancing the models' ability to identify and extract relevant information. Esteva et al. (36) and Su et al. (37) utilized BERT and BioBERT as encoders, which they fine-tuned for the task of question answering-based multi-document summarization specifically focused on COVID-19 literature (33), showcasing the versatility and efficacy of extractive approaches in handling large-scale biomedical data. Cai et al. (38) improved the SciBERT encoder by incorporating word co-occurrence information, advancing the performance of abstractive summarization for COVID-19 literature.

2.3 Biomedical lay summarization

The creation of a dataset for biomedical text simplification was first attempted for the CL-LaySumm 2020 shared task (6). This private dataset comprised 572 full-text papers, including abstracts and author-written lay language summaries (LLS) from journals published by Elsevier. The method for creating such datasets remains in use today. For instance, (32) and (39) introduced datasets containing approximately 5,000 pairs of scientific abstracts and LLS extracted from the Cochrane database. Similarly, (40) presented around 30,000 pairs of biomedical literature abstracts from PLOS and eLife, and (28) developed a dataset with about 28,000 biomedical abstract pairs from PLOS. A high-quality dataset called PLABA was created by (41), which includes 750 pairs of abstracts with sentence-aligned adaptations generated by human authors. MedEasi, another dataset akin to PLABA, contains 1,697 pairs of human-annotated sentences characterized by shorter lengths and fewer sentences (42). The CELLS dataset, introduced by (43), comprises 62,886 pairs of diverse scientific abstracts along with their corresponding LLS. Additionally, adapter modules for encoding domain-specific information for entity linking and text-pair classification were utilized by (44).

To generate LLS, pre-trained BART on biomedical data was employed by (32) and (40), with subsequent fine-tuning on their respective datasets. Different variants of unlikelihood loss to enhance simplification were used by (39) and (45). Controlling techniques such as prompts and multi-heads were proposed by (28) to adjust readability during summarization in both extractive and abstractive methods. Article-specific knowledge graphs detailing the technical concepts and relationships discussed in the articles, were leveraged by (46), who explored three distinct methods to integrate graph-based information into lay summarization models. Experiments on definition-based explanation retrieval and embedding-based explanation retrieval using the Retrieval-Augmented Generation (RAG) model were conducted by (43). They utilized UMLS entity-definition pairs for three keywords from the documents, concatenating them with the source document for definition-based explanation retrieval. For embedding-based explanation retrieval, Dense Passage Retrieval (DPR) (47) was used with BART as the generator.

2.4 Metrics

In evaluating biomedical lay summarization, three main criteria are considered - Relevance, Readability, and Factuality; each criterion is composed of one or more automatic metrics (46) (48).

- **Relevance**: ROUGE-1, ROUGE-2, and ROUGE-L and BERTScore.
- **Readability**: Flesch-Kincaid Grade Level (FKGL), Dale-Chall Readability Score (DCRS) and Coleman-Liau Index (CLI).
- **Factuality**: BARTScore.

2.4.1 ROUGE Scores

ROUGE-1, ROUGE-2, and ROUGE-L, (49) are prominent metrics for assessing the quality of summaries. ROUGE-1 evaluates the match of individual words (unigrams) between a generated summary and a reference, while ROUGE-2 examines the overlap of pairs of consecutive words (bigrams). These metrics are useful for determining how well the key terms and phrases are included in the automated summary. Conversely, ROUGE-L gauges the longest common subsequence (LCS) present in both the generated and reference summaries. This score is particularly insightful as it considers the order of the words, thus providing a good indication of how well the summary maintains the flow and structure of the reference content.

2.4.2 BERT Score

BERTScore is a metric designed to evaluate the quality of text by measuring the semantic similarity between pairs of text segments. This metric leverages the power of BERT, a deep learning model known for its ability to capture contextual embeddings of words. BERTScore calculates the cosine similarity by comparing the contextual embeddings of tokens from the candidate text with those from the reference text (50). This method allows it to effectively measure the relevance of generated text in tasks such as summarization, translation, or any content generation task where semantic accuracy is crucial. The BERTScore metric is mathematically defined by the following expression:

$$\text{BERTScore} = \frac{1}{|C|} \sum_{c \in C} \max_{r \in R} \text{cosine}(\mathbf{e}_c, \mathbf{e}_r)$$

In this formula, C denotes the set of tokens in the candidate text, R represents the set of tokens in the reference text, $\mathbf{e}_c$ and $\mathbf{e}_r$ are the BERT embeddings for tokens c and r respectively, and cosine refers to the cosine similarity between the embeddings.

2.4.3 FKGL

The FKGL (Flesch-Kincaid Grade Level) metric, developed by Kincaid et al. (51), is used to assess the readability of written material. It evaluates text readability by analyzing both the average word length and the average sentence length. The FKGL formula is expressed as:

$$\text{FKGL} = 0.39\left(\frac{\text{Number of Words}}{\text{Number of Sentences}}\right) + 11.8\left(\frac{\text{Number of Syllables}}{\text{Number of Words}}\right) - 15.59$$

This score reflects the U.S. school grade level required for comprehension of the text. A higher FKGL score corresponds to a more challenging text, indicating a direct relationship between the score and the text's complexity.

2.4.4 DCRS

DCRS (52) is another lexical metric, similar in purpose to the FKGL. It uses a list of approximately 3000 words that are considered understandable for fourth-grade American students. Words not included in this list are classified as difficult. The DCRS is calculated by taking the percentage of difficult words (those not on the list) in the text, and the average sentence length in words, then applying the following formula:

$$\text{DCRS} = 0.1579\left(\frac{\text{Number of Difficult Words}}{\text{Total Words}} \times 100\right) + 0.0496\left(\frac{\text{Total Words}}{\text{Number of Sentences}}\right)$$

2.4.5 CLI

CLI is a readability formula designed to gauge the understandability of English texts. Unlike some other readability metrics it depends solely on characters per word and words per sentence

(53). This index predicts the U.S. school grade level necessary to comprehend a text. The Coleman-Liau Index can be calculated using the formula:

$$\text{CLI} = 0.0588 \cdot L - 0.296 \cdot S - 15.8$$

where L represents the mean number of letters per 100 words, and S denotes the mean number of sentences per 100 words.

2.4.6 BARTScore

BARTScore utilizes the capabilities of the BART model as it estimates the likelihood of a generated summary being contextually and factually aligned with the source document by calculating the probability of the summary conditioned on the source text (54). This approach is particularly effective when BARTScore is fine-tuned on domain-specific datasets, as it allows the model to better understand and evaluate the nuances and specialized terminology inherent in different fields, leading to more accurate assessments of factuality. The BARTScore can be expressed as follows:

$$\text{BARTScore} = \frac{1}{T} \sum_{j=1}^{T} \log P(\text{summary}_j \mid \text{source}_j)$$

where T denotes the total number of tokens in the summary, and $P(\text{summary}_j \mid \text{source}_j)$ signifies the conditional probability of each summary token given the corresponding source text.

CHAPTER 3

METHODOLOGY

Entity descriptions in biomedical knowledge bases (KBs) often provide complete information about various entities. For instance, the Unified Medical Language System (UMLS) alone contains over 100 million pairs of concepts along with their corresponding definitions or descriptions. Biomedical documents frequently include numerous synonyms, full forms and short forms, hyponyms, homographs, etc. Hence, we leveraged the descriptions found for biomedical terms in KBs to aid in understanding biomedical documents. Knowledge graphs are an abundant source of valuable information in the biomedical field. For entity relationships, we use MSI, an existing biomedical knowledge graph. MSI includes a network of diseases, proteins, genes, drug targets, and biological functions. The relations are drug-protein interactions, disease-protein associations, protein-protein interactions, protein-function mappings, function-function relationships, and drug-disease treatments, and it has 484,654 positive triples. For synonyms knowledge, we again leverage UMLS. It has over 10 million synonyms derived from 187 distinct source vocabularies. In Section 3.2, we discuss in detail the pre-training process to incorporate these knowledge sources.

As discussed in Section 1, we integrate three distinct forms of biomedical knowledge: descriptions, relations, and synonyms. In the Table I below, we provide detailed examples. We leverage PubMed-BART as the backbone model for integrating these knowledge sources because it is fine-tuned on the PubMed corpus. Models trained on domain-specific data often

Knowledge	**Source**	**Examples**
Descriptions	UMLS	Aspirin (Drug) A medication used to reduce pain, fever, or inflammation. Hypertension (Disease) A condition in which the force of the blood against the artery walls is too high. Hemoglobin (Protein) The protein in red blood cells that carries oxygen. Diabetes Mellitus (Disease) A group of diseases that result in too much sugar in the blood. Insulin (Hormone) A hormone that regulates blood sugar levels.
Relations	MSI	p53 protein interacts with MDM2 protein. BRCA1 gene regulates RAD51 gene. Ibuprofen inhibits Cyclooxygenase-2 (COX-2) enzyme. Diabetes is associated with increased thirst and frequent urination. Glycolysis pathway includes Glucose-6-phosphate.
Synonyms	UMLS	Acute Myocardial Infarction, Acute Coronary Syndrome Aspirin, Acetylsalicylic Acid Glucose, Blood Sugar Hypertension, High Blood Pressure Neoplasm, Tumor

TABLE I: Knowledge types and examples

demonstrate superior performance (32). Figure 3 illustrates the general architecture of the model's encoder. In this architecture, we incorporate three adapter modules, each consisting of a feed-forward up layer and a feed-forward down layer, placed after the Add & Norm layer. The input from the Add & Norm layers is fed into these adapter modules. A fusion layer is introduced on top of the three adapters to consolidate the knowledge. Additionally, a residual connection is included from the feed-forward layer (preceding the Add & Norm layer) to the fusion layer.

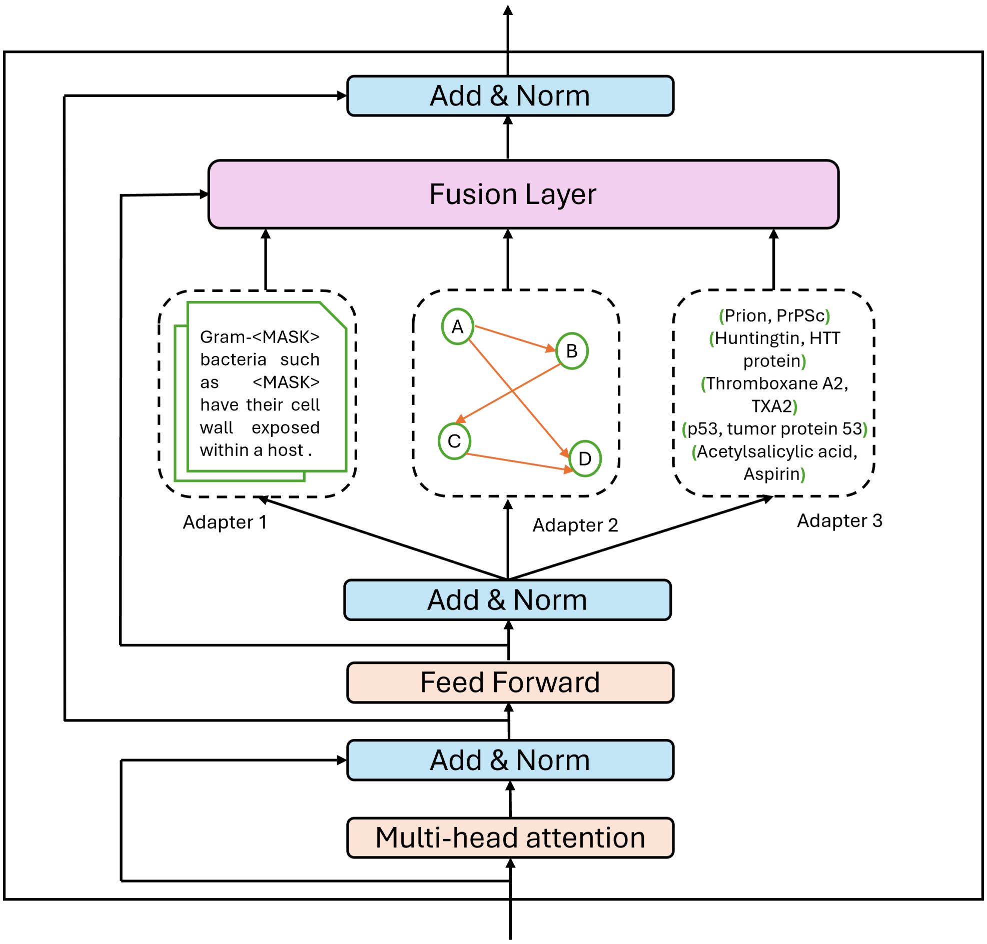


Figure 3: Adapters pre-training and knowledge fusion.

3.1 Adapters

Adapters are compact neural networks inserted between layers of a pre-trained language model (PLM). Typically, during the fine-tuning process for a specific downstream task, only the parameters of these adapters are adjusted while the original PLM's weights remain unchanged. This method ensures that adapter-based tuning introduces only a minimal number of additional parameters for each downstream task. As adapters are lightweight feed-forward network modules they can be inserted into the backbone PLM. Additionally, these methods are highly efficient, consuming less than 5% of a model's parameters. They are modular and require approximately 20MB of file size. Furthermore, multiple adapters can be grouped together depending on the task.

Bottleneck adapters incorporate feed-forward layers within each Transformer model layer. These layers are designed to include a down-projection matrix W_{down}, which compresses the hidden states to a smaller dimension $d_{\text{bottleneck}}$, followed by a non-linear function f, and an up-projection matrix W_{up} that restores the dimensions to match the original hidden layers. Additionally, a residual connection r is added:

$$y \leftarrow W_{\text{up}} \cdot f(W_{\text{down}} \cdot x) + r$$

These adapter layers can be strategically placed at various points within a Transformer block. The design allows for adjustments in residual connections, layer norms, activation functions, and the dimensions of the bottleneck as shown in Figure 4

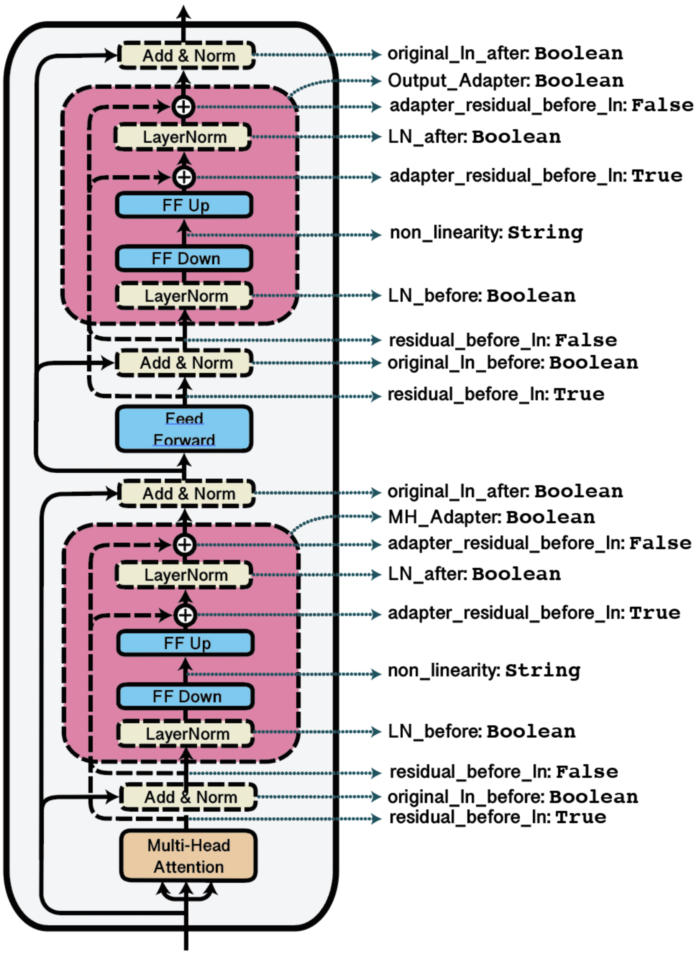

original_ln_after: Boolean
Output_Adapter: Boolean
adapter_residual_before_ln: False
LN_after: Boolean
adapter_residual_before_ln: True
non_linearity: String
LN_before: Boolean
residual_before_ln: False
original_ln_before: Boolean
residual_before_ln: True
original_ln_after: Boolean
MH_Adapter: Boolean
adapter_residual_before_ln: False
LN_after: Boolean
adapter_residual_before_ln: True
non_linearity: String
LN_before: Boolean
residual_before_ln: False
original_ln_before: Boolean
residual_before_ln: True
Add & Norm
LayerNorm
FF Up
FF Down
Feed Forward
Multi-Head Attention

Figure 4: Types of possible adapter configurations.

(55)

A critical hyperparameter in this setup is the bottleneck dimension $\mathrm{d}_{\mathrm{bottleneck}}$, which is defined using the *reduction factor*. This factor determines the ratio between the hidden dimension of the model and the bottleneck dimension:

$$\textit{reduction factor} = \frac{\mathrm{d}_{\mathrm{hidden}}}{\mathrm{d}_{\mathrm{bottleneck}}}$$

This setup allows for significant customization and optimization in the model's architecture, adapting it to various specific needs and improving efficiency. Several bottleneck adapter architectures have been proposed in the literature:Houlsby et al. (56) suggest placing adapter layers after both the multi-head attention block and the feed-forward network in each Transformer layer, while Pfeiffer et al. (57) propose adding an adapter layer solely after the feed-forward block in each Transformer layer. He et al. (58) introduce adapter layers in parallel to the original Transformer layers. Inspired by this, we used the bottleneck adapter (56) in our study.

3.2 Pretraining Adapters

Each type of knowledge is used to pretrain an adapter module using a self-supervised learning objective tailored to its specifics. Only the adapter module parameters are updated throughout the pretraining phase, with the weights of the underlying backbone PLM remaining fixed. The strategy for creating the objective function focuses on guiding the model to accurately predict and digest the information from a relevant knowledge source. Based on the kind of knowledge at hand, the exact nature of the objective varies which we discuss in detail in the sub-sections below. On top of the three adapters we deploy a fusion layer that combines the knowledge from all the adapters. In Table II, we provide the detailed format of input data.

Knowledge	**Source**	**Format**
Descriptions	UMLS	Enitity Name (Entity Type) Entity Definition
Relations	MSI	Entity1 relationship with Entity2
Synonyms	UMLS	Entity1, Entity2

TABLE II: Knowledge types and input formats

3.2.1 Descriptions knowledge

We employ the masked language modeling (MLM) objective. Specifically, for a given textual description of a biomedical entity, a percentage of its tokens are randomly masked to create a corrupted input sequence. In Table II, we provide the detailed input data format. Subsequently, the model is tasked with predicting the masked tokens. The masked language modeling (MLM) objective function aims to predict masked tokens in a corrupted input sequence. Formally, given a sequence of tokens $X = (x_1, x_2, ..., x_n)$, where each x_i represents a token, the MLM objective (59) is defined as:

$$L_{\text{MLM}}(X; \theta) = \sum_{i \in \text{Masked Tokens}} -\log P(x_i \mid x_{<i}, x_{>i}; \theta) \tag{3.1}$$

Here, X is the input sequence of tokens, x_i denotes the i-th token in the sequence, and $x_{<i}$ and $x_{>i}$ refer to the tokens before and after x_i, respectively. "Masked Tokens" denotes the set of tokens that have been randomly masked out in X. $P(x_i \mid x_{<i}, x_{>i}; \theta)$ represents the probability assigned by the model with parameters θ to token x_i given its context $x_{<i}$ and $x_{>i}$. The natural logarithm log is used, and $\mathcal{L}_{\text{MLM}}(X; \theta)$ is the MLM loss function, which is minimized during training to enhance the model's ability to reconstruct the original sequence from corrupted inputs.

Here, θ represents the PLM which includes the corresponding task adapter module. Only the task adapter module and the feed-forward layer are active for training, while the rest of the backbone model parameters remain frozen. In the context of biomedical entity descriptions, applying the MLM objective helps models effectively utilize the textual information provided by knowledge bases such as the UMLS, enhancing comprehension and utilization of specialized biomedical terminology.

3.2.2 Relations Knowledge

In general, a knowledge graph is a collection of triples in the format $K = \{(h, r, t) \mid h, t \in E, r \in R\}$, where E and R denote the sets of entities and relations, respectively (60). In Table II, we provide the detailed format of input data. During pre-training phase we train a model to assign a score of 1 to correct positive triples in K and a score of 0 to incorrect triples. We use the margin ranking loss function:

$$L_{\text{MR}}(x) = \frac{1}{N} \sum_{i=1}^{N} \max\left(0, f(x_1) - f(x_2) + \lambda\right) \tag{3.2}$$

where x_1 is a positive triple, x_2 is a negative triple, λ is the margin hyperparameter, and N is the number of negative samples per positive sample (61).

Consider a triple (h, r, t). We concatenate the words in h, r, and t to form a text $T = \text{concat}(h, r, t)$. We then use a pretrained language model (PLM) to convert this concatenated text T into a hidden representation, producing an output $X = \theta(\text{concat}(h, r, t))$. Finally, we feed this vector X into a feed-forward layer F to obtain the final output Y:

$$Y = F(\theta(\text{concat}(h, r, t)))$$

Here, θ represents the PLM which includes the corresponding task adapter module. Only the task adapter module and the feed-forward layer are active for training, while the rest of the backbone model parameters remain frozen.

3.2.3 Synonyms knowledge

To leverage this rich repository in our model, we employ a contrastive loss function. This type of loss is particularly useful for models that require emphasizing the similarity measurement between similar objects rather than dissimilar ones. Consider a knowledge base (KB) comprising a set of entities denoted by $E = \{e_1, e_2, \ldots, e_n\}$. Each entity e is associated with a set of synonyms represented as $S(e)$. Define N as the union of all terms in the KB, formally expressed as $N = \bigcup_{i=1}^{n} S(e_i)$. Our goal is to develop a mapping function $y : N \rightarrow \mathbb{R}^d$ that assigns a feature vector to each term in N.

In each batch, we have multiple sets of synonyms $S(e_i)$, where each set provides positive examples for any sample within the set, and negative examples are those outside the set. Each textual sample s_i is processed through a pre-trained language model (PLM) to obtain hidden representations. Specifically, for a sample text $T = s_i$, the PLM generates a hidden representation X. We then apply a feed-forward layer F to X to obtain the final feature vector Y:

$$Y = F(\theta(T))$$

where θ represents the PLM which includes the corresponding task adapter module. In this configuration, only the adapter module and the feed-forward layer are trained, while the remaining PLM parameters are kept fixed.

To train the function y, we employ the multi-similarity loss L_{MS} as defined in (62). The loss function is formulated as:

$$L_{MS} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{\alpha}\log\left(1+\sum_{k\in P_i} e^{-\alpha(C_{ik}-\lambda)}\right)+\frac{1}{\beta}\log\left(1+\sum_{k\in N_i} e^{\beta(C_{ik}-\lambda)}\right)\right)$$

where m denotes the number of examples in a mini-batch, α and β are scaling factors that adjust the impact of positive and negative terms in the loss, respectively, λ is a margin parameter used to distinguish positive from negative similarities, C_{ik} is the cosine similarity score between the i-th and k-th examples, and P_i and N_i are the sets of indices for positive and negative examples relative to i, respectively. Cosine similarity is used to compute C_{ik} based on the feature vectors Y.

3.2.4 Knowledge Fusion

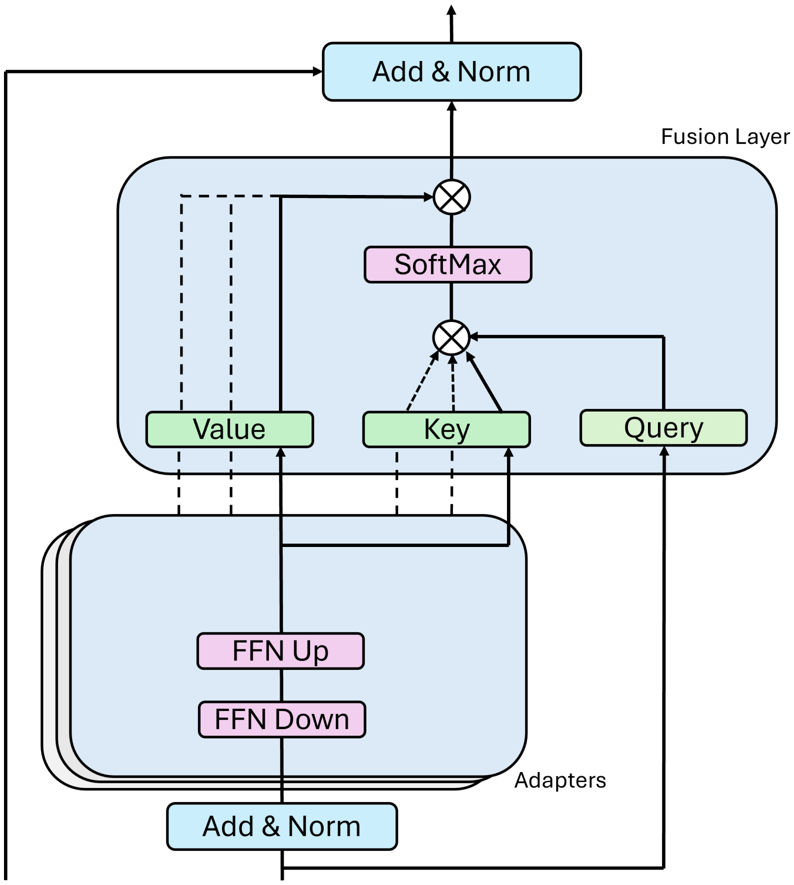


Figure 5: Fusion Mechanism.

After pretraining is complete, we employ fusion layers to integrate the learned knowledge. These layers operate by selecting the most suitable adapter for each specific text during downstream task training. The fusion mechanism we implement, as outlined in (63), consists of learnable components: Query, Key, and Value. The Query processes the output from the pretrained transformer weights, while the Key and Value handle outputs from their respective adapters. The Query's interaction with all Keys involves a dot product that passes through a softmax function, thus learning to adjust the weights of each adapter contextually.

Mathematically, this is expressed using Key, Value, and Query matrices at each transformer layer l, denoted K^l, V^l, and Q^l. For each layer l and time-step t, the feedforward sub-layer's output at layer l acts as the query vector. The outputs from each adapter at $z_{l,t}$ serve as inputs for both the Key and Value transformations. The mathematical operations involved are as follows:

$$s_{l,t} = \text{softmax}(h_{l,t} \cdot Q^l \odot z_{l,t,n} \cdot K^l), \quad n \in \{1, \dots, N\}$$

$$z'_{l,t,n} = z_{l,t,n} \cdot V^l, \quad n \in \{1, \dots, N\}$$

$$Z'_{l,t} = [z'_{l,t,1}, \dots, z'_{l,t,N}]$$

$$o_{l,t} = s_{l,t} \cdot Z'_{l,t}$$

where $\odot$ symbolizes the dot product, and $[\cdot, \cdot]$ indicates the concatenation of vectors. A clear illustration is provided in Figure 5.

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1 Datasets

We considered using 3 existing datasets – PLABA, eLife and PLOS. We discarded eLife due to the differences in source-target lexical similarity and length found upon manual inspection.

PLABA dataset contains 921 manually adapted summaries for 750 abstracts sourced from PubMed. The selection of these abstracts was based on the top-10 search results generated from 75 questions taken from the list of the most frequently asked consumer health questions in MedlinePlus. The adaptation process adhered to the following guidelines: Replacing arcane words with common synonyms and splitting sentences, converting passive voice constructions to active voice and omitting confidence intervals, p-values, and sentences deemed irrelevant to consumer understanding.

PLOS (Public Library of Science) dataset comprises 24,773 biomedical articles, each accompanied by an abstract and author summary written by the authors themselves. The creation followed these guidelines: Emphasizing how the work fits into the broader context and clearly present the research's significance, avoiding acronyms and complex terminology (28). In our manual inspection, we find the authors' adherence to be inconsistent with the guidelines, and each author has their own style of writing the author summary. We only used the abstract as the input and considered the author summary as the target simplification.

The dataset statistics for PLABA and PLOS are shown in Table III.

Data	**Metric**	**Statistic**	**Train**	**Val**	**Test**
PLABA	Abstract	Total Samples	635	138	148
		Max. Length	5273	3826	2692
		Avg. Length	1593	1625	1445
		Min. Length	385	465	338
	Lay Summary	Total Samples	635	138	148
		Max. Length	4930	4521	3028
		Avg. Length	1550	1550	1441
		Min. Length	311	463	249
PLOS	Abstract	Total Samples	24773	1376	1376
		Max. Length	4094	3387	3492
		Avg. Length	1684	1701	1687
		Min. Length	23	583	32
	Lay Summary	Total Samples	24773	1376	1376
		Max. Length	3353	2366	2531
		Avg. Length	1222	1223	1220
		Min. Length	32	379	343

TABLE III: Dataset statistics for PLABA and PLOS.

4.2 Experimentation

We fine-tuned existing Pre-trained Language Models (PLMs) as baselines and also fine-tuned leading open-source Large Language Models (LLMs) like Gemma2 and BioMistral for comparison. In Experiment 1, Experiment 2, and Experiment 3, we used one adapter module in each encoder/decoder layer, pre-training only these adapters while freezing the rest of the model. Experiment 1 utilized definitions knowledge, Experiment 2 employed relations knowledge, and

Experiment 3 applied synonyms knowledge, each with their respective task-specific objective functions discussed in Section 3.2. In the following sections, we refer to the model trained in Experiment 1 as $\text{Adapter}_{\text{MLM}}$, in Experiment 2 as $\text{Adapter}_{\text{MR}}$, in Experiment 3 as $\text{Adapter}_{\text{MS}}$, and in Experiment 4 as Adapter Fusion.

For Experiment 1, we pre-trained the adapter modules with Masked Language Modeling (MLM) using a probability of 0.15 on hard biomedical terms identified using ScispaCy Named Entity Recognition (NER). In Experiment 2, we used 4 negative samples for every positive sample with $\lambda = 0.5$, noting that a higher number of negative samples might be beneficial. Experiment 3 used $\alpha = 2$, $\beta = 50$, and $\lambda = 0.5$. Experiment 4 incorporated all three adapters in each encoder/decoder layer, training them on their respective knowledge types with their corresponding task-specific objective functions. Additionally, a fusion layer was added in each encoder/decoder module, trained in MLM style while freezing the rest of the model and adapters. For all experiments, including fine-tuning, we utilized mixed precision training, a learning rate of $2e^{-5}$, and the efficient 8-bit AdamW optimizer. In some experiments, we added UMLS definitions for the biomedical NER terms obtained during fine-tuning. To select the most feasible NER method, we first manually annotated 700 examples with the following guidelines:

1. Mark hard words that a layperson would not understand and require simplification in the lay summary.
2. Exclude basic biomedical terminology that a person with a degree in a non-health-related field would know.

3. Mark the reappearance of jargon in different formats, e.g., annotating both “esophagus” and “mid-esophagus.”

Given the impracticality of annotating all examples in our datasets, we tested various NER methods. ScispaCy NER demonstrated the highest accuracy of 50.5% when compared to our annotations, even outperforming GPT-3.5 Turbo, which had an accuracy of 29%.

4.3 Results

We present the results for PLABA and PLOS in Table IV and Table V, respectively. First, we will discuss the results obtained on PLABA dataset, followed by the PLOS dataset.

PLABA dataset: In terms of baseline models, PubMed-BART consistently performs better than T5 across all cases. Among the fine-tuned LLMs, BioMistral-7b consistently outperforms Gemma2-9b across all metrics in the PLABA dataset, suggesting that BioMistral-7b generalizes better even on smaller datasets and also obtains the highest ROUGE-L score (0.59). Among the adapter models, $\text{Adapter}_{\text{fusion}}$ consistently performs better across all metrics and achieves the highest scores in FKGL (12.57) and DCRS (8.97), indicating that it generates simpler and easier-to-understand terms, whereas $\text{Adapter}_{\text{MLM}}$ specifically performs better on the ROUGE scores, obtaining the highest ROUGE-1 (0.62) and ROUGE-2 (0.38) scores. We believe that due to the masked language modeling as the pre-training objective, $\text{Adapter}_{\text{MLM}}$ achieves more n-gram overlap, leading to higher ROUGE scores. An increase in performance across all metrics is observed when definitions are included during fine-tuning, compared to vanilla fine-tuning for both PubMed-BART and $\text{Adapter}_{\text{fusion}}$. PubMed-BART (definitions) obtains the highest CLI score (14.5).

Model	R-1↑	R-2↑	R-L↑	BERTs↑	FKGL↓	DCRS↓	CLI↓	BARTs↑
T5	0.6	0.35	0.58	0.9	13.16	11.25	14.87	-2.39
Pubmed-BART	0.6	0.35	0.58	0.9	13.19	11.28	14.98	-2.39
BioMistral-7b	0.61	**0.37**	0.59	**0.91**	13.6	10.83	15.46	-2.23
Gemma2-9b	0.55	0.34	0.51	0.88	12.61	10.8	15.03	-3.03
Adapter$_{\text{MLM}}$	**0.62**	0.38	**0.6**	0.9	12.76	10.62	15.46	-2.22
Adapter$_{\text{MR}}$	0.46	0.28	0.44	0.86	16.22	9.67	15.09	**-2.21**
Adapter$_{\text{MS}}$	0.44	0.27	0.43	0.87	14.48	10.47	15.17	-2.23
Adapter$_{\text{fusion}}$	0.56	0.34	0.54	0.86	12.97	10.8	15.07	-2.22
Pubmed-BART (definitions)	**0.62**	0.36	**0.6**	0.9	12.84	10.9	**14.5**	-2.36
Adapter$_{\text{fusion}}$ (definitions)	0.58	0.36	0.56	0.86	**12.57**	**8.97**	14.68	-2.32

TABLE IV: Performance metrics for various models on PLABA dataset.

PLOS dataset: For the PLOS dataset, the T5 model has better FKGL (14.95), DCRS (12.03), and BARTs (-3.59) scores compared to PubMed-BART, whereas PubMed-BART has higher ROUGE scores (R-1: 0.49, R-2: 0.18, R-L: 0.45) compared to T5, likely due to it being fine-tuned on in-domain PubMed data. BioMistral-7b performs better than Gemma2-9b, except on BERTScore (0.88) and CLI (15.03) where Gemma2-9b does better. Among the experiments, Adapter$_{\text{MLM}}$ model achieves higher ROUGE scores (R-1: 0.5, R-2: 0.19, R-L: 0.46) and Adapter$_{\text{fusion}}$ does better in FKGL (14.29) and DCRS (9.96) scores just like in the PLABA dataset. Overall, when the definitions are included during fine-tuning, the performance increases for the both datasets.

Model	**R-1↑**	**R-2↑**	**R-L↑**	**BERTs↑**	**FKGL↓**	**DCRS↓**	**CLI↓**	**BARTs↑**
T5	0.48	0.17	0.44	0.86	14.95	12.03	16.17	-3.59
Pubmed-BART	0.49	0.18	0.45	0.86	14.84	11.76	16.16	-3.52
BioMistral-7b	0.49	**0.2**	0.46	0.87	14.81	11.04	16.26	**-3.03**
Gemma2-9b	0.41	0.18	0.38	**0.88**	14.61	10.89	**15.03**	-3.19
Adapter$_{\text{MLM}}$	**0.5**	0.19	**0.46**	0.84	14.96	11.13	16.45	-3.35
Adapter$_{\text{MR}}$	0.38	0.15	0.35	0.82	15.55	10.96	17.88	-3.34
Adapter$_{\text{MS}}$	0.39	0.16	0.36	0.82	15.29	10.26	17.48	-3.26
Adapter$_{\text{fusion}}$	0.42	0.16	0.43	0.83	15.17	10.06	17.87	-3.28
Pubmed-BART (definitions)	0.49	0.18	0.45	0.86	14.85	11.77	16.16	-3.5
Adapter$_{\text{fusion}}$ (definitions)	0.43	0.17	0.44	0.83	**14.29**	**9.96**	17.43	-3.34

TABLE V: Performance metrics for various models on PLOS dataset.

4.4 Human Evaluation

To provide a comprehensive assessment of the lay summaries generated by our models, we conducted a human evaluation focusing primarily on readability. Four computer science graduates that don't have any experience in bio-medical or health related fields were tasked with rating the same 20 randomly sampled generated lay summaries. The evaluation used a 3-point scale based on the following simplification guidelines: 1 (Hard to understand), 2 (Partially understandable) and 3 (Easily understandable). The evaluators were instructed to read the abstract before rating each lay summary.

To measure inter-rater reliability, we calculated Krippendorff's alpha, which was 73.3% for the PLABA dataset and 72.5% for the PLOS dataset, indicating substantial agreement among evaluators. In Table VI and Table VII, we present the average human evaluation label percentages for all 20 samples from the PLABA and PLOS datasets, respectively. We observe that Adapter$_{\text{fusion}}$ (definitions) outperforms PubMed-BART (definitions) in human evaluations for the PLABA dataset. Adapter$_{\text{fusion}}$ (definitions) sees an increase in label 3 percentage and a decrease in label 1 and label 2 precentages. This suggests that Adapter$_{(}$definitions) produced more understandable lay summaries according to human evaluators. In PLOS, we observe that more lay summaries generated by PubMed-BART (definitions) are harder to understand. Overall, Adapter$_{\text{fusion}}$ (definitions) performs slightly better with higher scores for label 2 and label 3. This indicates that for the PLOS dataset, Adapter$_{\text{fusion}}$ (definitions) produce more easily understandable summaries, but the advantage is less clear than in the PLABA dataset. For both the datasets, vanilla Pubmed-BART under performs when compared to the other two models.

Label	**Pubmed-BART**	**Pubmed-BART (definitions)**	**Adapter$_{\text{fusion}}$ (definitions)**
1	0.17	0.11	0.06
2	0.40	0.43	0.37
3	0.43	0.46	0.57

TABLE VI: PLABA human evaluations

Label	**Pubmed-BART**	**Pubmed-BART (definitions)**	**Adapter fusion (definitions)**
1	0.43	0.36	0.25
2	0.33	0.27	0.29
3	0.24	0.37	0.46

TABLE VII: PLOS human evaluations

4.5 Case Study

We provide a sample of the lay summaries generated by Pubmed-BART (definitions) and Adapter$_{\text{fusion}}$ (definitions) models below in Table VIII and Table IX. We highlighted the differences in the generated summaries in color.

In the PLABA example lay summary in Table VIII, we observe that Adapter$_{\text{fusion}}$ (definitions) performs better than Pubmed-BART (definitions). It simplifies biomedical terms by providing exact definitions, by providing an estimated or concise short definition, or by altogether removing that word.

- Highlighted in red: Pubmed-BART (definitions) removes “autosomal dominant disorder” and Adapter$_{\text{fusion}}$ (definitions) further replaces it with “genetic condition inherited due to changes in a tumour gene”.
- Highlighted in blue: Pubmed-BART (definitions) replaces “renal” with “kidney”, whereas Adapter$_{\text{fusion}}$ (definitions) simplifies “pulmonary cysts, recurrent spontaneous pneumothoraces, cutaneous fibrofolliculomas, and kidney tumours of various types” to“cysts in the lungs, skin growths called fibrofolliculomas, and various types of kidney tumors”.
- Highlighted in magenta: Pubmed-BART (definitions) replaces “pneumothorax” in the abstract with “lung disease”, whereas Adapter$_{\text{fusion}}$ (definitions) replaces “pneumothorax” in the abstract with “spontaneous lung collapses” and “chest computed tomography” with “CT scan”.

- Highlighted in brown: $\text{Adapter}_{\text{fusion}}$ deletes “lymphangioleiomyomatosis and pulmonary Langerhans cell histiocytosis” and simplifies “chronic respiratory insufficiency” to “difficulty in breathing”.

In the PLOS example lay summary in Table IX, we find that the $\text{Adapter}_{\text{fusion}}$ (definitions) generated summary is a bit more simplified.

- Highlighted in red: $\text{Adapter}_{\text{fusion}}$ (definitions) simplifies “epidemics of influenza virus” to “outbreaks of the flu virus”.
- Highlighted in blue: The reference lay summary and the $\text{Adapter}_{\text{fusion}}$ (definitions) generated summary remove biomedical jargon, whereas Pubmed-BART (definitions) does not.
- Highlighted in magenta: $\text{Adapter}_{\text{fusion}}$ (definitions) explains what T-cells are and simplifies the complex terms, whereas the reference lay summary removes them altogether, and Pubmed-BART (definitions) does not address this.
- Highlighted in brown: We find that both $\text{Adapter}_{\text{fusion}}$ (definitions) and Pubmed-BART (definitions) do not perform as well compared to the reference lay summary.

Model	**Text**
Reference lay summary	Birt-Hogg-Dubé syndrome is a rare autosomal dominant disorder caused by changes in the gene FLCN that makes a protein called folliculin, which helps suppress tumor growth. It is often associated with the development of multiple cysts (sacs of air) in the lungs, sudden collapse of the lung, non-cancerous tumors in hair follicles around the face, neck, and chest (fibrofolliculomas), and kidney tumors. Birt-Hogg-Dubé has no preference between male and female and often appears when people are in their 30s or 40s. Multiple cysts on both lungs are identified using x-rays in more than 80% of patients. More than half experience one or more collapsed lung events which is when air leaks into the space between the lung and chest wall. A family history of collapsed lung is an important clue and suggests a diagnosis of Birt-Hogg-Dubé. Unlike other cystic lung diseases, Birt-Hogg-Dubé does not lead to loss of lung function over time or to ongoing breathing problems. Kidney tumors affect about 30% of patients during their lifetime. There can be multiple number of tumors, and they can be recurring.....
Pubmed-BART (definitions)	Birt-Hogg-Dubé syndrome (BHD) is a rare inherited disorder caused by mutations in the tumour suppressor gene FLCN, encoding the protein folliculin. Its clinical expression typically includes multiple pulmonary cysts, recurrent spontaneous pneumothoraces, cutaneous fibrofolliculomas and kidney tumours of various types. BHD has no sex predilection and tends to manifest in the third or fourth decade of life. Multiple bilateral lung cysts are found on chest computed tomography in $>$ 80% of patients and more than half experience one or more episodes of lung disease. A family history of pneumothorax is an important clue, which suggests the diagnosis of BHD. Unlike other cystic lung diseases such as lymphangioleiomyomatosis and pulmonary Langerhans cell histiocytosis, BHD does not lead to progressive loss of lung function and chronic respiratory insufficiency. Renal tumours affect about 30% of the patients during their lifetime, and can be multiple and recurrent.....
Adapter fusion (definitions)	Birt-Hogg-Dubé syndrome (BHD) is a rare genetic condition inherited due to changes in a tumour gene called FLCN, encoding the protein folliculin. It is characterized by the development of multiple cysts in the lungs, skin growths called fibrofolliculomas, and various types of kidney tumors. BHD has no sex predilection and tends to manifest in the third or fourth decade of life. More than 80% of people with BHD will have multiple lung cysts visible on a CT scan, and more than half experience one or more episodes of spontaneous lung collapses. A family history of punctured lungs can be a significant indicator that someone may have BHD. Unlike other lung diseases, BHD does not typically cause loss of the lung function or difficulty in breathing. About 30% of individuals with BHD will develop kidney tumors during their lives, which may occur multiple times.....

TABLE VIII: Comparison of PLABA lay summaries

Model	**Text**
Reference lay summary	Influenza virus continues to pose a significant risk to global health and is responsible for thousands of deaths each year in the United States. This threat is largely due to the ability of the influenza virus to undergo rapid changes, allowing it to escape from immune responses elicited by previous infections or vaccinations. Certain internal determinants of the influenza virus are largely conserved across different viral strains and represent attractive targets for potential "universal" influenza vaccines. Here, we demonstrated that cross-subtype protection against the influenza virus could be obtained through simultaneous priming of multiple arms of the immune response against conserved elements of the influenza virus. These results suggest a novel strategy that could potentially form a primary component of a universal influenza vaccine capable of providing long-lasting protection.
Pubmed-BART (definitions)	Seasonal epidemics of influenza virus result in 36,000 deaths annually in the United States. Current vaccines against influenza virus elicit an antibody response specific for the envelope glycoproteins. However, high mutation rates result in the emergence of new viral serotypes, which elude neutralization by preexisting antibodies. T lymphocytes have been reported to be capable of mediating heterosubtypic protection through recognition of internal, more conserved influenza virus proteins. Here, we demonstrate that cross-subtype protection could be obtained through simultaneous priming of multiple arms of the immune response against conserved elements of the influenza virus. Our findings suggest that an influenza vaccine capable of eliciting both CD4+ T cell and antibodies specific for highly conserved influenza proteins may be able to protect humans against influenza, and act as the basis for a potential "universal" vaccine capable of providing long-lasting protection.
Adapter fusion (definitions)	Seasonal outbreaks of the flu virus cause around 36,000 deaths each year in the United States. Current influenza vaccines mainly help the body make antibodies that target specific virus proteins, but the flu virus changes rapidly, creating new versions that these antibodies can't recognize. Research has shown that T cells, a type of immune cell, can help protect against different flu strains by recognizing more stable, internal flu proteins. In our study, we used a special flu virus that included a specific T cell marker to show that relying solely on T cells or non-neutralizing antibodies doesn't provide strong protection against the flu. However, when these two immune responses work together, they provide much better protection. This enhanced immunity depends partly on the support of certain immune cells in the lungs. Our findings suggest that a flu vaccine that encourages both T cells and antibodies targeting these stable proteins could lead to better protection against various strains of the flu and might form the foundation for a "universal" vaccine.

TABLE IX: Comparison of PLOS lay summaries

CHAPTER 5

CONCLUSION

This thesis presents the first study on utilizing adapter modules by pre-training them on distinct knowledge sources and incorporating a knowledge consolidation phase to enhance lay summarization on the PLOS and PLABA datasets. We extend this approach by adding definitions during fine-tuning. We compare four different methods for integrating knowledge into encoder-decoder summarization models, focusing on the readability of the generated outputs. Our findings indicate that incorporating external knowledge can significantly improve lay summarization, especially in generating readable text and clarifying technical concepts. Future research should explore the use of pre-training mechanisms and the integration of adapters into open-source LLMs.

5.1 Limitations

Manual human evaluation is time-consuming, and significant work is needed to develop automatic evaluation metrics. Simplification is subjective, and using multiple metrics to judge a sentence is not always ideal. All the aforementioned metrics are reference-dependent, and their efficacy is limited due to the reliance on high-quality reference summaries, which are often challenging to obtain for lay summaries. Furthermore, these metrics struggle to accurately identify hallucinations, which is especially crucial for lay summaries in the health domain to ensure accurate health decisions. Although human evaluation offers thorough assessment, the high costs and time required impede scalability for larger datasets. There are many possible configurations with adapters, as discussed in Section 3.1. As seen in Table VII, even the lay summaries generated by the models are harder to understand. This is due to the nature of the author-written lay summaries in the PLOS dataset. PLABA is a manually hand-crafted dataset for lay summarization, whereas PLOS summaries are automatically extracted from the PLOS journal, with authors expected to follow certain submission guidelines. Therefore, the level of simplification is not consistent among authors, as the task is highly subjective, rendering the author-written summaries harder to understand for laypeople.

5.2 Future Work

We pre-trained the models on 2 data sources - UMLS and MSI, but there are a wide variety of biomedical knowledge bases like PubChem, DrugBank, Gene Ontology, and Kyoto Encyclopedia of Genes and Genomes (KEGG) that could also be utilized for further enriching the models with diverse biomedical knowledge. Due to limitations in computational power, we only placed the adapter module after the feed-forward layer, but there is a variant where you can place the adapter module before and after the feed-forward layer. Additionally, one can experiment with different combinations of residual connections, activation functions, and layer norms inside the adapter modules as shown in Figure 4. For experiment2, we used 4 negative samples for every positive sample, with the number decided based on computational constraints. Higher number of negative samples might yield better results and is definitely something to try. The human evaluation by people from non-biomedical fields can only capture the simplification of the generated summaries and their relevance to some extent. However, they can't capture the factuality. Therefore, we need biomedical domain experts to evaluate the factuality.

CITED LITERATURE

1. Henke, J.: Public engagement with COVID-19 preprints: Bridging the gap between scientists and society. Quantitative Science Studies, 5(2):271–296, 05 2024.

2. Kirmani, M., Kour, G., Mohd, M., Sheikh, N., Khan, D. A., Maqbool, Z., Wani, M. A., and Wani, A. H.: Biomedical semantic text summarizer. BMC Bioinformatics, 25(1):152, 04 2024.

3. Xie, Q., Luo, Z., Wang, B., and Ananiadou, S.: A survey for biomedical text summarization: From pre-trained to large language models, 2023.

4. Basyal, L. and Sanghvi, M.: Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models, 2023.

5. Wang, M., Wang, M., Yu, F., Yang, Y., Walker, J., and Mostafa, J.: A systematic review of automatic text summarization for biomedical literature and ehrs. Journal of the American Medical Informatics Association: JAMIA, 28(10):2287–2297, September 2021.

6. Chandrasekaran, M. K., Feigenblat, G., Hovy, E., Ravichander, A., Shmueli-Scheuer, M., and de Waard, A.: Overview and insights from the shared tasks at scholarly document processing 2020: Cl-scisumm, laysumm and longsumm. In Proceedings of the First Workshop on Scholarly Document Processing, pages 214–224, 2020.

7. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019, 2019.

8. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.

9. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog, 2019.

10. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

11. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D.: Scaling laws for neural language models, 2020.

12. Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., and Hashimoto, T. B.: Benchmarking Large Language Models for News Summarization. Transactions of the Association for Computational Linguistics, 12:39–57, 01 2024.

13. Wang, G. and Wu, W.: Surveying the landscape of text summarization with deep learning: A comprehensive review, 2023.

14. Žagar, A. and Robnik-Šikonja, M.: One model to rule them all: Ranking slovene summarizers. In Text, Speech, and Dialogue, eds. K. Ekštein, F. Pártl, and M. Konopík, pages 15–24, Cham, 2023. Springer Nature Switzerland.

15. Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.

16. See, A., Liu, P. J., and Manning, C. D.: Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368, 2017.

17. Chang, C., Zhou, J., Zeng, X., and Tang, Y.: Sumope: Enhanced hierarchical summarization model for long texts. In Advanced Data Mining and Applications, eds. X. Yang, H. Suhartanto, G. Wang, B. Wang, J. Jiang, B. Li, H. Zhu, and N. Cui, pages 307–319, Cham, 2023. Springer Nature Switzerland.

18. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.

19. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J.: Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

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

21. Luhn, H. P.: The automatic creation of literature abstracts. IBM Journal of research and development, 2(2):159–165, 1958.

22. Mihalcea, R. and Tarau, P.: Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 404–411, 2004.

23. Erkan, G. and Radev, D. R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research, 22:457–479, 2004.

24. Nallapati, R., Zhai, F., and Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

25. Liu, Y. and Lapata, M.: Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345, 2019.

26. Sotudeh Gharebagh, S., Goharian, N., and Filice, R.: Attend to medical ontologies: Content selection for clinical abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, eds. D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, pages 1899–1905, Online, July 2020. Association for Computational Linguistics.

27. Wallace, B. C., Saha, S., Soboczenski, F., and Marshall, I. J.: Generating (factual?) narrative summaries of rcts: Experiments with neural multi-document summarization, 2020.

28. Luo, Z., Xie, Q., and Ananiadou, S.: Readability controllable biomedical document summarization. arXiv preprint arXiv:2210.04705, 2022.

29. Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N.: A discourse-aware attention model for abstractive summarization of long documents. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), eds. M. Walker, H. Ji, and A. Stent, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

30. Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Burdick, D., Eide, D., Funk, K., Katsis, Y., Kinney, R. M., Li, Y., Liu, Z., Merrill, W., Mooney, P., Murdick, D. A., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A. D., Wang, K., Wang, N. X. R., Wilhelm, C., Xie, B., Raymond, D. M., Weld, D. S., Etzioni, O., and Kohlmeier, S.: CORD-19: The COVID-19 open research dataset. In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, eds. K. Verspoor, K. B. Cohen, M. Dredze, E. Ferrara, J. May, R. Munro, C. Paris, and B. Wallace, Online, July 2020. Association for Computational Linguistics.

31. DeYoung, J., Beltagy, I., van Zuylen, M., Kuehl, B., and Wang, L. L.: MSˆ2: Multi-document summarization of medical studies. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, eds. M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, pages 7494–7513, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

32. Guo, Y., Qiu, W., Wang, Y., and Cohen, T.: Automated lay language summarization of biomedical scientific reviews. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 160–168, 2021.

33. Xie, Q., Tiwari, P., and Ananiadou, S.: Knowledge-enhanced graph topic transformer for explainable biomedical text summarization. IEEE Journal of Biomedical and Health Informatics, 28(4):1836–1847, 2024.

34. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4):1234–1240, 2020.

35. Xie, Q., Bishop, J. A., Tiwari, P., and Ananiadou, S.: Pre-trained language models with domain knowledge for biomedical extractive summarization. Knowledge-Based Systems, 252:109460, 2022.

36. Esteva, A., Kale, A., Paulus, R., Hashimoto, K., Yin, W., Radev, D., and Socher, R.: Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. npj Digital Medicine, 4(1):68, April 2021.

37. Su, D., Xu, Y., Yu, T., Siddique, F. B., Barezi, E., and Fung, P.: CAiRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. In Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020, eds. K. Verspoor, K. B. Cohen,

M. Conway, B. de Bruijn, M. Dredze, R. Mihalcea, and B. Wallace, Online, December 2020. Association for Computational Linguistics.

38. Cai, X., Liu, S., Yang, L., Lu, Y., Zhao, J., Shen, D., and Liu, T.: Covidsum: A linguistically enriched scibert-based summarization model for covid-19 scientific papers. Journal of Biomedical Informatics, 127:103999, 2022.

39. Devaraj, A., Wallace, B. C., Marshall, I. J., and Li, J. J.: Paragraph-level simplification of medical texts. In Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting, volume 2021, page 4972. NIH Public Access, 2021.

40. Goldsack, T., Zhang, Z., Lin, C., and Scarton, C.: Making science simple: corpora for the lay summarisation of scientific literature. arXiv preprint arXiv:2210.09932, 2022.

41. Attal, K., Ondov, B., and Demner-Fushman, D.: A dataset for plain language adaptation of biomedical abstracts. Scientific Data, 10(1):8, 2023.

42. Basu, C., Vasu, R., Yasunaga, M., and Yang, Q.: Med-easi: Finely annotated dataset and models for controllable simplification of medical texts. arXiv preprint arXiv:2302.09155, 2023.

43. Guo, Y., Qiu, W., Leroy, G., Wang, S., and Cohen, T.: Cells: A parallel corpus for biomedical lay language generation. arXiv preprint arXiv:2211.03818, 2022.

44. Lai, T. M., Zhai, C., and Ji, H.: Keblm: Knowledge-enhanced biomedical language models. Journal of Biomedical Informatics, 143:104392, 2023.

45. Flores, L. J. Y., Huang, H., Shi, K., Chheang, S., and Cohan, A.: Medical text simplification: Optimizing for readability with unlikelihood training and reranked beam search decoding. arXiv preprint arXiv:2310.11191, 2023.

46. Goldsack, T., Luo, Z., Xie, Q., Scarton, C., Shardlow, M., Ananiadou, S., and Lin, C.: Overview of the biolaysumm 2023 shared task on lay summarization of biomedical research articles. arXiv preprint arXiv:2309.17332, 2023.

47. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906, 2020.

48. Team, B.: Biolaysumm — biomedical lay summarization. `https://biolaysumm.org/`, 2024. Accessed: 2024-07-26.

49. Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81, 2004.

50. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675, 2019.

51. Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., and Chissom, B. S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.

52. Dale, E. and Chall, J. S.: A formula for predicting readability: Instructions. Educational research bulletin, pages 37–54, 1948.

53. Coleman, M. and Liau, T. L.: A computer readability formula designed for machine scoring. Journal of Applied Psychology, 60(2):283, 1975.

54. Yuan, W., Neubig, G., and Liu, P.: Bartscore: Evaluating generated text as text generation. Advances in Neural Information Processing Systems, 34:27263–27277, 2021.

55. Team, A.: Adapter methods — adapterhub documentation. `https://docs.adapterhub.ml/methods.html#bottleneck-adapters`, 2024. Accessed: 2024-07-04.

56. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S.: Parameter-efficient transfer learning for nlp, 2019.

57. Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S.: Mad-x: An adapter-based framework for multi-task cross-lingual transfer, 2020.

58. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G.: Towards a unified view of parameter-efficient transfer learning, 2022.

59. Xu, Y. and Lapata, M.: Document Summarization with Latent Queries. Transactions of the Association for Computational Linguistics, 10:623–638, 05 2022.

60. Baumgartner, M.: How to compare apples to oranges: Integrating heterogeneous data sources with representation learning. ResearchGate, 01 2022.

61. Team, P.: Marginrankingloss — pytorch documentation. `https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html`, 2024. Accessed: 2024-07-26.

62. Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R.: Multi-similarity loss with general pair weighting for deep metric learning. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5017–5025, 2019.

63. Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I.: AdapterFusion: Non-destructive task composition for transfer learning. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, eds. P. Merlo, J. Tiedemann, and R. Tsarfaty, pages 487–503, Online, April 2021. Association for Computational Linguistics.

VITA

NAME	V. S. Sandeep Reddy Dwarampudi
EDUCATION	M.S., Computer Science, University of Illinois at Chicago, Chicago, Illinois, Summer 2024.
EXPERIENCE	Research Assistant (Fall 2023 - Summer 2024).