

An Iterative Optimizing Framework for Radiology Report Summarization with ChatGPT

Chong Ma, Zihao Wu, Jiaqi Wang, Shaochen Xu, Yaonai Wei, Fang Zeng, Zhengliang Liu, Xi Jiang, Lei Guo, Xiaoyan Cai, Shu Zhang, Tuo Zhang, Dajiang Zhu, Dinggang Shen *Fellow, IEEE*, Tianming Liu *Senior Member, IEEE*, Xiang Li

Abstract—The “Impression” section of a radiology report is a critical basis for communication between radiologists and other physicians. Typically written by radiologists, this part is derived from the “Findings” section, which can be laborious and error-prone. Although deep-learning based models, such as BERT, have achieved promising results in Automatic Impression Generation (AIG), such models often require substantial amounts of medical data and have poor generalization performance. Recently, Large Language Models (LLMs) like ChatGPT have shown strong generalization capabilities and performance, but their performance in specific domains, such as radiology, remains under-investigated and potentially limited. To address this limitation, we propose ImpressionGPT, leveraging the contextual learning capabilities of LLMs through our dynamic prompt and iterative optimization algorithm to accomplish the AIG task. ImpressionGPT initially employs a small amount of domain-specific data to create a dynamic prompt, extracting contextual semantic information closely related to the test data. Subsequently, the iterative optimization algorithm automatically evaluates the output of LLMs and provides optimization suggestions, continuously refining the output results. The proposed ImpressionGPT model achieves superior performance of AIG task on both MIMIC-CXR and OpenI datasets without requiring additional training data or fine-tuning the LLMs. This work presents a paradigm for localizing LLMs that can be applied in a wide range of similar application scenarios, bridging the gap between general-purpose LLMs and the specific language processing needs of various domains.

Impact Statement—With the advent of Artificial General Intelligence (AGI) and Large Language Model (LLM) such as ChatGPT, we envision that a series of medical text data processing methodologies and the corresponding data management

solutions can be replaced by LLMs. Thus, in this work, we leveraged the text understanding and summarization capability of ChatGPT for the task of generating the “Impression” section of a radiology report, which is a critical basis for communication between radiologists and other physicians. Here we propose ImpressionGPT, which leverages the in-context learning capability of LLMs by constructing dynamic prompts using domain-specific, individualized data via an iterative optimization approach. We envision that the proposed framework can become a paradigm for similar works in the future to adapt general-purpose LLMs to specific domains via an in-context learning approach.

Index Terms—Radiology Report Summarization, Dynamic Prompt, Iterative Optimization, ChatGPT.

I. INTRODUCTION

TEXT summarization is the process of compressing a large amount of text data into a shorter summary while maintaining its coherence and informative properties. It has long been a critical and well-studied research area within the field of Natural Language Processing (NLP). As the volume of digital textual information is growing at an extraordinary rate in both the general and medical domains, the need for efficient and accurate text summarization models grows correspondingly. In earlier studies, Luhn [1] proposed the first automatic summarization algorithm based on statistical methods. Later, a variety of alternative approaches have been proposed, such as rule-based methods [2], latent semantic analysis [3], and graph-based techniques [4]. Although traditional methods significantly advanced the research and application of text summarization, they often lack the ability to capture complex semantics and contextual information to generate human-level summarization performance [5].

The introduction of neural networks and deep learning methods, especially the sequence-to-sequence models that employ encoder-decoder architectures for generating summaries [6] conveyed promising results. These approaches enabled the creation of more fluent and contextually relevant summaries compared with rule-based and statistical methods. Recently, the field of NLP, including text summarization, has experienced drastic changes with the emergence of large-scale, pre-trained foundational models, such as BERT [7] and GPT [8]. These models are trained on massive volumes of text data, which enables them to learn rich contextual representations and generate human-like languages. Study [9] has been conducted to demonstrate that fine-tuning these foundational models on text summarization tasks can lead to state-of-the-art performance, outperforming earlier models by a wide margin.

C. Ma, Y. Wei, L. Guo, X. Cai, and T. Zhang are with the School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China. (e-mail: {mc-npu, rean_wei}@mail.nwpu.edu.cn, {lguo, xiaoyan, tuozhang}@nwpu.edu.cn).

J. Wang and S. Zhang are with the School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China. (e-mail: jiaqi.wang@mail.nwpu.edu.cn, shu.zhang@nwpu.edu.cn).

X. Jiang is with the Clinical Hospital of Chengdu Brain Science Institute, MOE Key Lab for Neuroinformation, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, 611731, China. (e-mail: xijiang@uestc.edu.cn).

D. Shen is with School of Biomedical Engineering, ShanghaiTech University, Shanghai 201210, China, and Department of Research and Development, Shanghai United Imaging Intelligence Co., Ltd., Shanghai 200030, China, and also with Shanghai Clinical Research and Trial Center, Shanghai, 201210, China. (e-mail: Dinggang.Shen@gmail.com).

Z. Wu, S. Xu, Z. Liu, and T. Liu are with the school of computing, University of Georgia, Athens, GA 30602, USA. (e-mail: {zw63397, sx76699, zl18864, tliu}@uga.edu).

D. Zhu is with Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington 76019, USA, (e-mail: dajiang.zhu@uta.edu).

F. Zeng and X. Li are with the Department of Radiology, Massachusetts General Hospital, Boston 02114, USA, (e-mail: {fzeng1, xli60}@mgh.harvard.edu).

Radiology reports are pivotal in clinical decision-making since they can provide crucial diagnostic and prognostic information to healthcare professionals [10]. The volume of imaging studies and the complexity of radiology data are both growing at an increasing rate, thus raising an urgent need for efficient language processing, including extracting key information from radiology reports. Text summarization can address this challenge by automatically generating concise, informative, and relevant summaries of radiology reports, thus can significantly enhance clinical workflows, reduce the workload of healthcare professionals, and improve patient care [9]. With the help of automatic text summarization methods, healthcare professionals can efficiently identify essential information, which leads to faster decision-making, optimized resource allocation, and improved communication among multidisciplinary teams [5].

Compared with general NLP tasks, radiology report summarization has its own unique challenges. It would be difficult for general-purpose NLP models to accurately capture the key information due to the highly specialized and technical nature of the language used. The potential risks associated with misinterpretation of crucial findings, and the significance of maintaining the contextual and relational aspects of such findings further complicate the task [11]. In response, multiple specialized language processing methods have been developed for medical text summarization, which can be broadly categorized into three groups: traditional, deep learning-based, and large language model-based. Traditional methods such as [2] lay the foundation for text summarization research. However, they often lacked the ability to capture complex semantics and contextual information. The introduction of deep learning techniques, such as CNNs, demonstrated superior performance in capturing the unique language features and context of radiology reports [11]. However, these techniques often require large volumes of annotated data for training and are limited to specific tasks and/or domains. Subsequently, with the introduction of the Transformer model, its distinctive multi-head global attention mechanism has led to its widespread application in medical image analysis [12], [13]. Moreover, Transformer-based models, such as BERT [7] and GPT [8], brought the emergence of Large Language Models (LLMs) and opened new possibilities for radiology text processing. Fine-tuning BERT on domain-specific data, such as chest radiology reports, demonstrating better capability in capturing clinical context and generating high-quality summaries [10]. Nonetheless, utilizing pre-trained language models such as BERT still needs a significant volume of annotated data for the downstream tasks (e.g., text summarization). Furthermore, training LLMs like ChatGPT [14] and GPT-4 [15] still demands a substantial amount of data, even when this data is unannotated. However, in certain specialized domains, the available data is often extremely limited. Utilizing such limited data is insufficient to train an effective large language model. Even with access to an adequate amount of data, training a model with an immense number of parameters necessitates a considerable amount of resources. Hence, the high demands on data quantity and hardware resources pose significant obstacles to the application of large language models in specialized

domains.

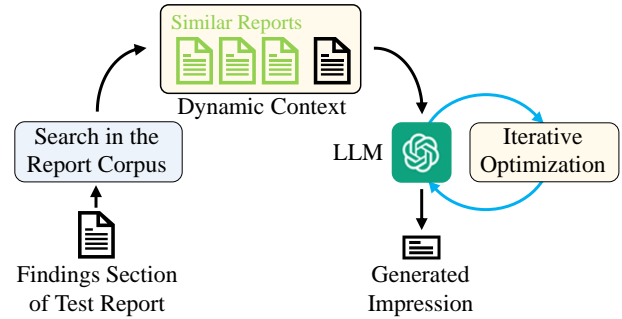


Fig. 1. Overview of ImpressionGPT. Initially, ImpressionGPT selects similar reports from the corpus based on the findings section of test report. Subsequently, a dynamic context is created and fed into ChatGPT. An iterative optimization algorithm is then employed to fine-tune the generated response in an interactive way. Finally, an optimized impression that conforms to the findings section of test report is produced.

It is noteworthy that models such as ChatGPT and GPT-4 possess strong in-context learning abilities, allowing them to extract useful semantic information based on provided prompts [16]. Coupled with their inherent text generation capabilities, even in scenarios with limited training data, these large language models can adapt and generate domain-specific answers through the optimization of prompts. In this study, we utilized ChatGPT and optimized its generated result for radiology report summarization. An iterative optimization algorithm is designed and implemented via prompt engineering to take advantage of ChatGPT's in-context learning ability while also continuously improving its response through interaction. Specifically, as shown in Fig. 1, we use similarity search techniques to construct a dynamic prompt to include semantically- and clinically-similar existing reports. These similar reports are used as examples to help ChatGPT learn the text descriptions and summarizations of similar imaging manifestations in a dynamic context. We also develop an iterative optimization method to further enhance the performance of the model by performing automated evaluation on the response generated by ChatGPT and integrating the generated text and the evaluation results to compose the iterative prompt. In this way, the generated response from ChatGPT is continually updated for optimized results under pre-defined guidance. We evaluate our ImpressionGPT on two public radiology report datasets: MIMIC-CXR [17] and OpenI [18]. Our experimental results show that ImpressionGPT performs substantially better than the current methods in radiology report summarization, with only a small dataset (5-20 samples) used for optimizing. Overall, the main contributions of this work are:

- In-context learning of Large Language Model (LLM) with limited samples is achieved by similarity search. Through the identification of the most similar examples in the corpus, a dynamic prompt is created that encompasses the most useful information for LLM.
- An iterative optimization algorithm is developed with a dynamic prompt scheme to further optimize the generated result. The iterative prompt provides feedback on the responses generated by LLM and the corresponding

evaluation, followed by further instructions to iteratively update the prompts.

- A new paradigm for optimizing LLMs' generation using domain-specific data. The proposed framework can be applied to any scenarios involving the development of domain-specific models from an existing LLM in an effective and resource-efficient approach. And the corresponding code is available on GitHub¹.

II. BACKGROUND AND RELATED WORKS

A. Text Summarization in Natural Language Processing

Text summarization is to extract, summarize or refine the key information of the original text to obtain the main content or general meaning of the the original text. There are two major categories of text summarization: extractive summarization and abstractive summarization. Among them, extractive summarization [2]–[4], [19] is to take one or more sentences from a text or text set to construct a summary. An advantageous aspect of this approach lies in its simplicity, and its results exhibit a low tendency towards deviations from the essential message conveyed in the text. Before the emergence of artificial neural network, the practice of text summarization predominantly relied on the technique of extractive summarization. For example, Hongyan [2] developed filtering rules based on prior knowledge to remove unimportant parts of the text to obtain a summary. LexRanK [4] represented the words and sentences in a graph and search the text with the highest similarity in the graph. [3] used algebraic statistics to extract latent semantic information from text and generates text summaries based on latent semantic similarities. After the emergence of artificial neural networks and deep learning, methods such as BertSUM [19] and TransEXT [9] were training based on the BERT [7] model. However, extractive summarization suffers from incoherent generation of summaries, uncontrollable length, and the quality of the results is severely dependent on the original text.

For the task of abstractive summarization [6], [9], [19]–[21], there is no issue as mentioned previously. Abstractive summarization task is an end-to-end generative task, and it is necessary to understand the meaning of the original text and generate a new summarization. Compared with extractive summarization, abstractive summarization is more challenging, but it is also more in line with the daily writing habits of human beings, so it has gradually become a significant research focus in the field of text summarization since the introduction of artificial neural networks and deep learning. For example, PGN (LSTM) [6] used a pointer-generator network to copy words from the original text and also retains the ability to generate new words, thus improving the seq2seq+attention model architecture. Similarly, CGU [20] proposed a global encoding framework for summary generation, which uses a combination of CNN and self-attention to filter the global encoding of text, solving the alignment problem between source text and target summary. With the emergence of BERT [7], a milestone model within the NLP field, the previous training method

was changed to use the strategy of pre-training + fine-tuning. TransABS [9] accomplished generative summarization based on the BertSUM [19] used a two-stage fine-tuning method and achieved optimality on three datasets. CAVC [21] used a Mask Language Modeling (MLM) strategy based on the BERT model to further improve the performance.

B. Radiology Report Summarization

With the development of text summarization in NLP, text processing related to the medical field is also gaining attention. In the standard radiology report, the impression section is a summary of the entire report description. Therefore, Automatic Impression Generation (AIG) has become the focus of NLP research in the medical field [10], [22]–[26]. Earlier studies have focused on the use of seq2seq methods. For example, [22] trained a bi-directional LSTM based model as a summary extractor using the method of Multi-agent Reinforcement Learning.

After the emergence of BERT [7], BioBERT [23] pre-trained BERT using large-scale biomedical corpora, surpassing previous methods in a variety of medical-related downstream tasks such as named entity recognition, relationship extraction, and question answering. This work explored the path of using pre-trained language models within the biomedical domain. Similar, ClinicalRadioBERT [24] pre-trained a BERT model and proposed a knowledge-infused few-shot learning (KI-FSL) approach that leverages domain knowledge for understanding radiotherapy clinical notes. ChestXrayBERT [10] pre-trained BERT using a radiology-related corpus and combined it as an encoder with a Transformer decoder to perform the diagnostic report summarization task. And WGSUM [25] constructed a word graph from the findings section of radiology report by identifying the salient words and their relations, and proposed a graph-based model WGSUM to generate impressions with the help of the word graph. [26] utilized a graph encoder to encode the word graph during pre-training to enhance the text extraction ability of the model, and introduced contrast learning to reduce the distance between keywords. Its results on AIG outperform the previous methods. All the above methods have achieved good results in the medical text domain based on the pre-trained language model, but still have the problem of poor generalization due to the low complexity of the model.

C. Large Language Model

With the advent of BERT [7] model based on Transformer architecture, an increasing number of studies related to Natural Language Processing (NLP) have incorporated pre-training + fine-tuning methodologies. The approach that first pre-training on a large amount of unlabeled data and then fine-tuning on a small portion of labeled data has proven to achieve more outstanding results. For example, before BERT, GPT-1 [27] with 117 million parameters has been initially trained using self-supervised pre-training + supervised fine-tuning. It directly used the Transformer decoder to achieve excellent results on natural language inference and question-and-answer tasks. Later, Google proposed the landmark model BERT,

¹<https://github.com/MoMarky/ImpressionGPT>

which introduced Transformer encoder and further improved the performance by using Mask Language Modeling and Next Sentence Prediction methods in the pre-training stage, where the number of parameters in BERT-Large has reached 340 million. Four months after the release of BERT, GPT-2 [8] was introduced, which further extended the model parameters and training data set based on GPT-1, with Extra Large of GPT-2 model reaching 1.5 billion parameters. In addition, the researchers [8] found that with the expanded training dataset, outstanding results of large language model could be achieved in downstream tasks without using fine-tuning. GPT-3 [28] further expanded the data size and parameter size based on GPT-2, and the maximum parameter reached 175 billion, and its performance on downstream tasks was significantly improved. And they first proposed a training paradigm of unsupervised pre-training + few-shot prompt.

In comparison to small Pre-trained Language Models (PLMs), LLMs possess superior generalization capability. They can accurately learn potential features of input text and perform effectively across different downstream tasks, even without fine-tuning. One prominent foundational model of a large language model is ChatGPT [14], based on the GPT-3.5 model, which employs training data in conversation mode to facilitate user-friendly human-machine interaction. ChatGPT has been widely integrated into various applications such as education and healthcare, and performs well in tasks such as text classification, data expansion, summarization, and other natural language processing [29]–[34]. Although ChatGPT performs well in most tasks, its performance in specialized domains is still unsatisfactory. Hence, we propose ImpressionGPT, an iterative optimization algorithm that enables ChatGPT to achieve excellent performance on the radiology report summarization task.

D. Prompt Engineering

Prompt engineering is a burgeoning field that has garnered significant attention in recent years due to its potential to enhance the performance of LLMs. The fundamental idea behind prompt engineering is to utilize prompts as a means to program LLMs, which is a vital skill required for effective communication with these models [35], such as ChatGPT. Recent research has demonstrated that designing prompts to guide the model toward relevant aspects of input can lead to more precise and consistent outputs. This is particularly crucial in applications such as language translation and text summarization, where the quality of the output is paramount.

In general, prompt engineering is a new paradigm in the field of natural language processing, and although still in its early stages, has provided valuable insights into effective prompt patterns [36]. These patterns provide a wealth of inspiration, highlighting the importance of designing prompts to provide value beyond simple text or code generation. However, crafting prompts that are suitable for the model can be a delicate process. Even a minor variation in prompts could significantly impact the model's performance. Therefore, finding the most suitable prompts remains an important challenge. Typically, there are two main types of prompts: manual prompts and automated template prompts.

1) *Manual Prompt*: Manual prompts are designed manually to guide LLMs towards specific inputs. These prompts provide the model with explicit information about what type of data to focus on and how to approach the task at hand [36]. Manual prompts are particularly useful when the input data is well-defined and the output needs to adhere to a specific structure or format [28]. For example, in the medical field where interpretability is crucial, manually created prompts are often employed to guide the model toward focusing on specific aspects of the input data. In the area of medical text security, for instance, manual prompts were utilized to guide the model toward identifying and removing private information in medical texts, effectively solving the ethical issues associated with medical data [33]. Overall, manual prompts play a vital role in improving model performance in various domains by providing the model with a more structured and focused approach to the task.

2) *Automated Template Prompt*: While manual prompts are a powerful tool for addressing many issues, they do have certain limitations. For example, creating prompts requires time and expertise, and even minor modifications to prompts can result in significant changes in model predictions, particularly for complex tasks where providing effective manual prompts is challenging [37]. To address these issues, researchers have proposed various methods for automating the process of prompt design, different types of prompts can assist language models in performing specific tasks more effectively. Prompt mining involves extracting relevant prompts from a given dataset [37], while prompt paraphrasing improves model performance and robustness by increasing prompt diversity. Gradient-based search [38] helps identify optimal prompts in a model's parameter space, and prompt generation [39] can create new prompts using techniques such as generative models. These discrete prompts are typically automatically searched for in a discrete space of prompt templates, often corresponding to natural language phrases. Other types of prompts, such as continuous prompts [40] that construct prompts in a model's embedding space and static prompts that create fixed prompt templates for input can also aid in task performance.

These different prompt types can be used independently or in combination to help language models perform various tasks, including natural language understanding, generation, machine translation, and question-answering.

III. METHOD

In this section, we first illustrate the pipeline of our ImpressionGPT. Then we elaborate on the dynamic prompt generation in Sec. III-A and the iterative optimization in Sec. III-B. Fig. 2 shows the pipeline of our ImpressionGPT. Firstly, as shown in the first step of mainstream (part *a* in Fig. 2), we use a labeler to categorize the “Findings” section of the report and extract disease labels. Then, based on the disease category, we search for similar reports in the existing diagnostic report corpus, as shown in part *b* of Fig. 2. And we designed a dynamic prompt (shown in part *c* of Fig. 2) to construct a context environment with similar diagnostic reports, so that

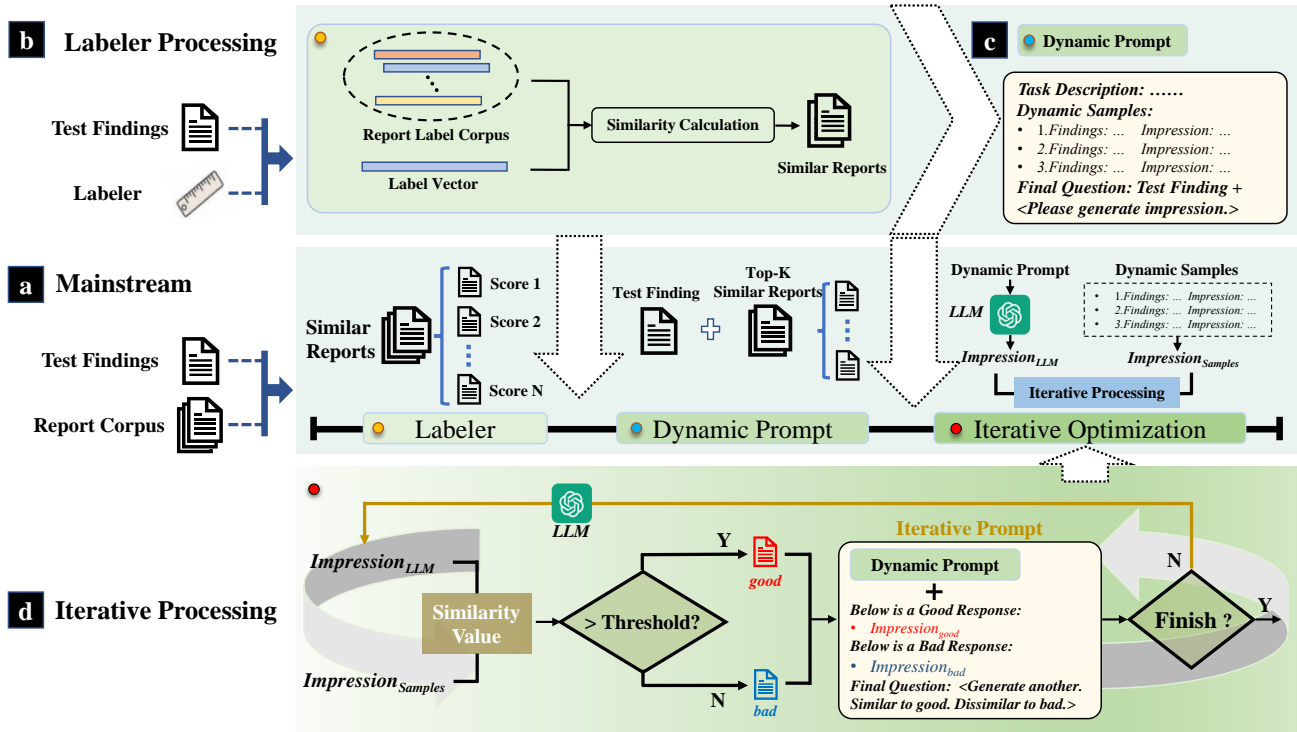


Fig. 2. The pipeline of our ImpressionGPT. Part *a* in the middle is the mainstream of our method. We first use a labeler to categorize the diseases of test report and obtain the similar reports in the corpus (part *b*), and then construct a dynamic prompt in part *c*. Part *d* accomplishes the iterative optimization of LLM’s generated result through interaction with positive (*good*) and negative (*bad*) responses.

ChatGPT can learn to summarize diagnostic reports related to the current disease. We refer to this as “dynamic context”. Based on the dynamic context, we utilized an iterative optimization method, as shown in the part *d* of Fig. 2, to optimize the response of ChatGPT. During the iterative optimization process, we compare the generated “Impression” from ChatGPT with examples in dynamic prompt to obtain good and bad responses. These evaluated responses are inserted to the iterative prompt with further guidance to ensure that the next response is closer to the former good response while avoiding the former bad response. Overall, our method requires a small number of examples, facilitating ChatGPT’s acquisition of excellent domain-specific processing capabilities. More details of dynamic prompt generation and iterative optimization can be found in the following subsections.

A. Dynamic Prompt Generation

In previous manual-designed prompts, fixed-form prompts were frequently employed for simple tasks that were easily generalized, such as translation, Q&A, and style rewriting. However, these fixed-form prompts were found to be insufficient in providing prior knowledge for more intricate tasks and datasets that are peculiar to specific domains, like processing medical diagnosis reports, resulting in poor performance of ChatGPT. Consequently, we propose a hypothesis which suggests that constructing dynamic prompts by utilizing similar examples from relevant domain-specific corpora can enhance the model’s comprehension and perception. In this work, dynamic prompt generation primarily comprises two main

components: similarity search and prompt design. Below is a detailed introduction to these two components.

1) *Similarity Search*: The purpose of similarity search is to find diagnostic reports within the corpus that are similar to the Test Findings. The process for similarity search mainly encompasses two phases. Initially, a disease classifier is employed to extract the disease categories appearing in the input radiology report. Subsequently, relying on these categories, a similarity calculation is conducted on the report corpus to obtain examples similar to the input radiology report.

In previous studies, corpus search typically involve two approaches: text-based [41] and feature-based [42]. However, text-based methods can be time-consuming and significantly increase search time, particularly with large corpora. Feature-based methods require feature extraction and storage of each sample, making it more space-demanding. Therefore, in this study, we utilize diagnostic report labels for similarity search, requiring only prior identification and local label storage of the report sample. Our approach substantially reduces time and space costs. Moreover, using tag values to calculate the similarity enables the identification of samples with similar diseases, which provides domain-specific and even sample-specific knowledge that can be leveraged by ChatGPT’s contextual processing capabilities. Specifically, we use CheXpert labeler [43] as the disease classifier, which is a rule-based labeler to extract observations from the free text radiology reports. The observations contain 14 classes based on the prevalence in the chest radiology reports. As shown in Fig. 3, each observation is represented by a letter from ‘a’ to ‘n’, and

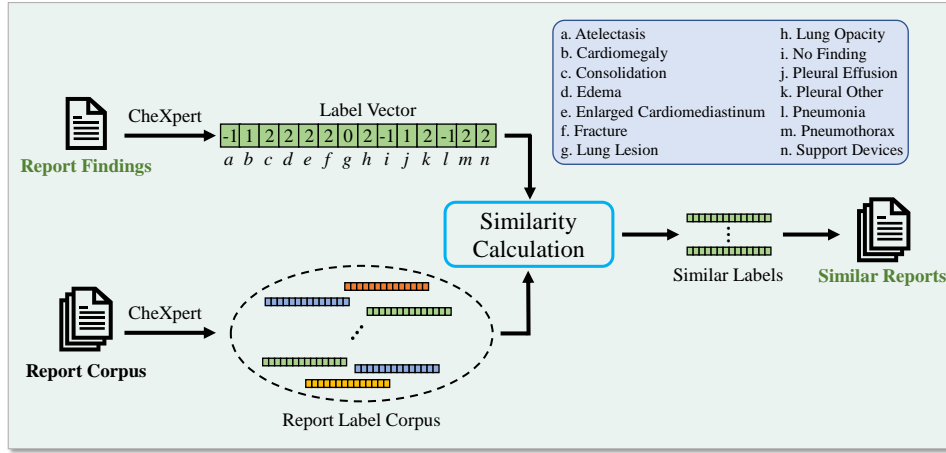


Fig. 3. Details of similarity search process. CheXpert is initially employed to obtain the label vector for the radiology reports present in the corpus. Then the similarity between the label vector of the test report and the label corpus is calculated, allowing for the identification of the similar radiology reports.

each contains four categories. The categories correspond to the clearly presence ('1') or absence ('0') of the observation in the radiology report, the presence of uncertainty or ambiguous description ('-1'), and the unmentioned observation (*blank*), which is replaced by the number '2' to facilitate similarity calculation. The labels of each radiology report in the corpus are extracted and saved as a one-dimensional vector of scale 1×14 . The label vectors of test reports are then compared to those in the corpus using Euclidean distance, and the radiology reports that are closest to each other are selected to generate dynamic prompts. This method ensures reliable and accurate disease classification of radiology reports. In this work, we directly use the split training set as a corpus and extract and preserve the corresponding disease labels for MIMIC-CXR [17] and OpenI [18] datasets. More details about dataset processing can be found at Section IV-A.

2) *Prompt Design*: Prior to introducing the design of our dynamic prompt, it is necessary to provide an overview of the input format when utilizing the ChatGPT API. The message that is fed into ChatGPT is accessed through the API in the form of a string, which is categorized into three distinct roles: *System*, *User*, and *Assistant*, as illustrated in Fig. 4. These roles are represented by the colors red, green, and blue, respectively. The *System* message initiates the conversation and provides information regarding the task while constraining the behavior of the *Assistant*. The *User* message instructs the *Assistant* and serves as the input provided by the user. Lastly, the *Assistant* component represents the response that is generated by the model. Notably, as the *Assistant* message can be artificially set, it enables the implementation of our dynamic prompt and facilitates iterative optimization.

In this work, we employ the prefix form for designing the dynamic prompt. The dynamic prompt consists of three modules: task description, dynamic samples, and final query, as illustrated in Fig. 4. The task description module specifies the role of the ChatGPT as a chest radiologist and provides a brief overview of the radiology report summary task along with a simple rule that serves as the foundation for the entire prompt. Subsequently, the dynamic prompt integrates

similar reports obtained in Sec. III-A1. The dynamic samples module, depicted in the central part of Fig. 4, utilizes a question-and-answer format to provide the prompt in each sample. Specifically, the question part consists of a pre-defined question sentence and the "Findings" section of dynamic sample and is treated as the message of the *User* role. Then the "Impression" section of the dynamic sample is treated as the following message of the *Assistant* role. At the end of our dynamic prompt, the same pre-defined question is used as in the previous samples, and the "Findings" section of the test report is inserted. Overall, the resulting dynamic prompt uses a question-answering approach and provides multiple examples that have similar content to the target test sample, thus creating a data-specific dynamic context.

B. Iterative Optimization of Response

In Sec. III-A, we constructed a dynamic context within the prompt in order to facilitate the model learning relevant prior knowledge from semantically similar examples. However, the use of this fixed form of prompt on ChatGPT produced a one-off effect, as we cannot guarantee that the generated response is appropriate. Therefore, based on the dynamic prompt, we utilized an iterative optimization operation to further optimize the generated result of ChatGPT. The specific optimization process is presented in Algorithm 1.

The first input of the algorithm is the dynamic prompt ($Prompt_{Dy}$) constructed in Sec. III-A. In line 6 of Algorithm 1, we receive a initial *response* from ChatGPT based on our dynamic prompt. Then, we evaluate it with other impression section of similar report and calculate an evaluation *score* at line 10. In lines 11-15, we compare above *score* with a predefined threshold ($Thres$) and construct our iterative prompt. Finally we feed this iterative prompt into ChatGPT to generate an optimized response (*response* in line 8) and repeat the above procedures. The purpose of the iterative prompt is to enable ChatGPT to optimize its response during the iterative process, so that the responses are more similar to those considered good and avoid the ones that are considered bad.

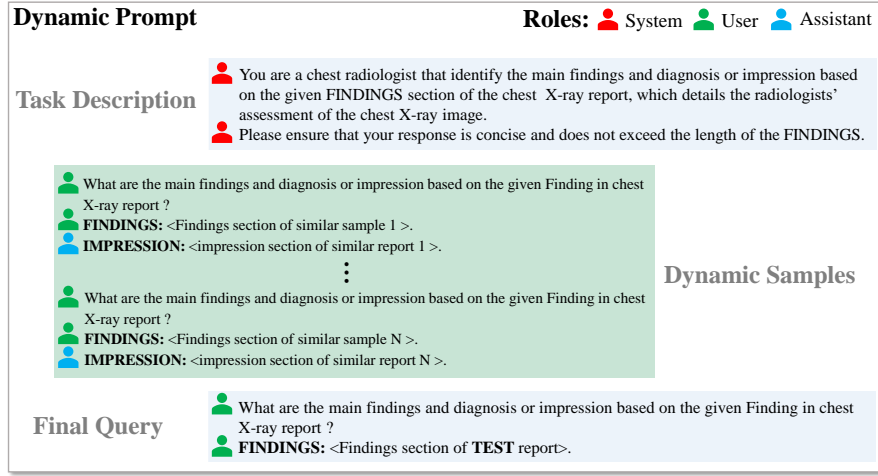


Fig. 4. Details of dynamic prompt. The dynamic prompt is composed of task description, dynamic samples, and final query. Each component contains message of multiple roles. The flags in front of each sentence represent the role of the current message, red for System, green for User, and blue for Assistant.

Algorithm 1 Iterative Optimization Algorithm

```

1: Input: Dynamic prompt with  $N_s$  similar reports
2: Initialize:  $I$ =Iteration times,  $Thres$ =Threshold of evaluation score,  $iter=0$ 
3: Definition:  $GPT$  means ChatGPT,  $Prompt_{Dy}$  means our dynamic prompt,  $Prompt_{Iter}$  means our iterative prompt,  $L$  means evaluate function,  $M_{good}$  and  $M_{bad}$  represent prompt merging with good and bad response
4: while  $iter < I$  do
5:   if  $iter = 0$  then
6:      $response = GPT(Prompt_{Dy})$ 
7:   else
8:      $response = GPT(Prompt_{Iter})$ 
9:   end if
10:   $score = \frac{1}{N_s} \sum_{i=0}^{N_s} L(response, Impression_i)$ 
11:  if  $score > Thres$  then
12:     $Prompt_{Iter} = M_{good}(Prompt_{Dy}, response)$ 
13:  else
14:     $Prompt_{Iter} = M_{bad}(Prompt_{Dy}, response)$ 
15:  end if
16:   $iter++$ 
17: end while

```

Through this method, ChatGPT becomes iterative and self-optimizing. A detailed description of the evaluation method and iterative prompt design is described below.

1) *Response Evaluation:* During the process of iterative optimization, it is crucial to assess the quality of a response. In this study, we employ the Rouge-1 score, a prevalent metric for evaluating summarization models. The similarity between the generated results and ground-truth is measured on a word-by-word basis by the Rouge-1 score, thus enabling the evaluation of the outputs at a more refined scale. The dynamic prompt incorporates N radiology reports having the highest similarity in the corpus, with the impression section deemed of high reference value for the ChatGPT response

evaluation. As depicted in line 10 of Algorithm 1, we calculate the Rouge-1 score of ChatGPT's response with the impression section ($Impression_i$) of each similar report in the dynamic prompt, and average it to get the final evaluation result ($score$ in line 10). Subsequently, we derive "good response" and "bad response" based on the pre-defined threshold and merge the outcomes into the iterative prompt ($Prompt_{Iter}$). Specifically, if the evaluated score is higher than the previously defined threshold ($score > Thres$), it is considered as a good response. Conversely, if the score is lower than the threshold, the response is considered as a bad response. We update the iterative prompt by integrating the feedback through our merging function (M_{good} and M_{bad}). After constructing the iterative prompt, we feed it again into ChatGPT. Note that, this process is repeated until the maximum number of iterations ($iter = I$). Details of iterative prompt design are described in the following subsection.

2) *Prompt Design:* The subsequent and crucial stage after identifying the positive and negative feedback is to utilize the evaluation results to instruct ChatGPT in generating enhanced responses. As shown in Fig. 5, the initial part of the iterative prompt is the dynamic prompt described in Sec. III-A2, which delivers the fundamental context for subsequent enhancement. For the responses evaluated in Sec. III-B1, a pair of *User* and *Assistant* roles are included in the iterative prompt. The *User* initiates the message with "Below is an excellent impression of the FINDINGS above" for the positive feedback and "Below is a negative impression of the FINDINGS above" for the negative feedback. The response produced by ChatGPT is positioned after the corresponding *User* message, adopting the "Instruction + Response" form to enable ChatGPT to learn the relevant content from the good and bad samples. As shown in the bottom "Final Query" in Fig. 5, the final query that concludes the iterative prompt provides specific optimization rules to regenerate a response that is consistent with the good response and avoids the bad response. Additionally, a length limit instruction is included to avoid ChatGPT from generating overly verbose responses. Note that after extensive

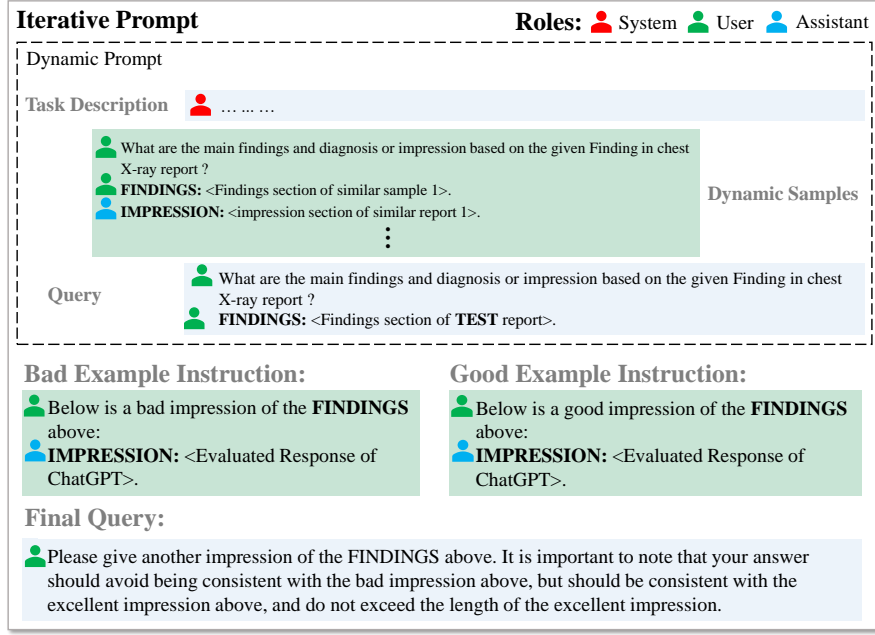


Fig. 5. Details of iterative prompt. We construct our iterative prompt by including the bad and good example instructions along with the final query, in addition to applying a “Instruction + Response” approach in the example instruction for optimizing the response.

experimental validation, we finally insert one good and n bad responses in the iterative prompt. The number of bad examples (n) is indefinite and can be appended all the time as long as the input limit of ChatGPT is not exceeded. We conducted ablation experiments and discussion on the setting of the number of good and bad responses in Section I of supplementary materials. In the process of each iteration, good response or bad response will be updated, which enables our prompt to optimize ChatGPT’s responses in time.

To summarize, our approach involves utilizing a dynamic prompt to establish a contextual understanding that is highly relevant to the semantics of the given test case. This context is then fed into ChatGPT to obtain an initial response, which is further evaluated and incorporated into an iterative prompt. The iterative prompt is used to elicit subsequent responses that are specific to the task domain, thereby enabling self-iterative updates of ChatGPT’s generated result with limited examples. Our experimental results demonstrate that ImpressionGPT delivers superior results in the summarization of radiology reports, as elaborated in Sec. IV-D.

IV. EXPERIMENT AND ANALYSIS

A. Dataset and Evaluation Metrics

We evaluated our ImpressionGPT on two public available chest X-ray datasets: MIMIC-CXR [17] and OpenI [18]. MIMIC-CXR dataset contains of 227,835 radiology reports, which are amalgamated into our corpus after applying the official split that includes both the training and validation sets. The test set is reserved solely for evaluating the effectiveness of our ImpressionGPT. In line with the objective of medical report summarization, we excluded ineligible reports by implementing the following criteria: (1) removal of incomplete

reports without finding or impression sections, (2) removal of reports whose finding section contained less than 10 words, and (3) removal of reports whose impression section contained less than 2 words. Consequently, we filtered out 122,014 reports to construct our corpus and included 1,606 test reports. OpenI dataset contains 3,955 reports, and we use the same inclusion criteria as above and finally obtained 3419 available reports. Since the official split is not provided, we follow [25] to randomly divided the dataset into train/test set by 2400:576. Similar to the MIMIC-CXR dataset, we use the training set to build our corpus, while the test set is reserved for evaluating the model.

TABLE I
THE STATISTICS OF THE TWO BENCHMARK DATASETS WITH OFFICIAL SPLIT FOR MIMIC-CXR AND RANDOM SPLIT FOR OPENI AFTER PRE-PROCESSING. WE REPORT THE AVERAGED SENTENCE-BASED LENGTH (AVG. SF, AVG. SI), THE AVERAGED WORD-BASED LENGTH (AVG. WF, AVG. WI) OF BOTH “FINDINGS” AND “IMPRESSION” SECTIONS.

Type	MIMIC-CXR [17]		OpenI [18]	
	Train	Test	Train	Test
Report Num	122,014	1,606	2,400	576
AVG. WF	55.78	70.67	37.89	37.98
AVG. SF	6.50	7.28	5.75	5.77
AVG. WI	16.98	21.71	10.43	10.61
AVG. SI	3.02	3.49	2.86	2.82

Table I shows the statistics of MIMIC-CXR and OpenI datasets after pre-processing. The code to process radiology reports is publicly available on GitHub². In our experiments, we employ Rouge metrics to evaluate the generated impression from our ImpressionGPT, and we reported F1 scores

²<https://github.com/MoMarky/radiology-report-extraction>

TABLE II
ABLATION STUDY OF IMPRESSIONGPT ON MIMIC-CXR AND OPENI DATASETS. **BOLD** DENOTES THE BEST RESULT.

Dataset	Method	R-1↑	R-2↑	R-L↑	BS-P↑	BS-R↑	BS-F1↑
MIMIC-CXR [17]	Fixed Prompt	33.29	16.48	28.35	86.07	88.15	87.07
	Dynamic Prompt	45.14	25.75	38.71	87.25	90.04	88.60
	Fixed Prompt + Iterative Opt	35.68	20.30	31.69	88.73	88.46	88.55
	Dynamic Prompt + Iterative Opt	54.45	34.50	47.93	91.56	91.33	91.41
OpenI [18]	Fixed Prompt	33.48	19.80	31.33	87.42	90.61	88.94
	Dynamic Prompt	42.07	27.02	39.78	88.92	91.76	90.27
	Fixed Prompt + Iterative Opt	46.63	33.17	44.74	89.28	92.28	90.71
	Dynamic Prompt + Iterative Opt	66.37	54.93	65.47	93.59	94.15	93.83

for Rouge-1 (R-1), Rouge-2 (R-2), and Rouge-L (R-L) that compare word-level unigram, bi-gram, and longest common sub-sequence overlap with the reference impression of the test report, respectively. In addition to using Rouge metrics, we incorporate the factual consistency (FC) metric, as introduced in previous works [25], [26]. The FC metric measures similarity between reference impressions and generated impressions by incorporating generic disease-related observations. Precision, recall, and F1 score are subsequently utilized to assess the performance. Given that existing studies only provide FC metrics for the MIMIC-CXR dataset, we also compute the Precision (FC-P), Recall (FC-R), and F1 score (FC-F1) for ImpressionGPT specifically on the MIMIC-CXR dataset.

B. Implementation Details

In our framework, dynamic prompt construction and iterative optimization of response are essential components, and therefore, the related parameters are crucial. Specifically, in dynamic prompt construction, we define the number of similar reports as N_s , which determines the richness of dynamic semantic information in our prompt. In the iterative optimization of response, we define the evaluation threshold as T , the number of good response and bad response after evaluation as G_d and B_d , and the number of iterations as I . The parameter T determines the quality of ChatGPT’s response during the optimization process. G_d and B_d determine the richness of positive and negative feedback information in the iterative prompt. And I determines the degree of optimization of ChatGPT’s response in our framework.

We conducted detailed ablation experiments on all parameters in supplementary materials and obtained the optimal values of $N_s = 15$, $T = 0.7$, $G_d = 1$, $B_d = n(n > 1)$, $I = 17$. It should be noted that there is no upper limit set for the value of B_d , and the value of n is flexibly adjusted based on the length of the actual input information during the optimization process. Detailed experimental analysis can be found in Section I of the supplementary material.

C. Main Property Studies

In this section, we ablate our ImpressionGPT in two aspects: the effectiveness of dynamic prompt and iterative optimization. In addition to employing Rouge metrics, we introduced BERTScore [44] to further evaluate these two modules. As shown in the Table II, the upper section displays the results of four methods on the MIMIC-CXR dataset, while the lower

section showcases experimental outcomes on OpenI dataset. BS-P, BS-R, and BS-F1 respectively represent the Precision, Recall, and F1 metrics of BERTScore. As shown in the rows of “Fixed Prompt”, we construct the prompt using fixed examples, which means that for each report in the test set, the prefix instruction in the fixed prompt is consistent. In the rows of “Dynamic Prompt” in Table II, we first search the corpus for examples similar to the test report, and then construct the dynamic prompt. It can be seen that the experimental results for Rouge-1, Rouge-2, and Rouge-L are improved by %11.85, %9.27, and %10.36 on MIMIC-CXR dataset, and by %8.59, %7.22, and %8.45 on OpenI dataset. And there is also a noticeable improvement in the BERTScore metrics. In the rows of “Fixed Prompt + Iterative Opt”, we continue to introduce our iterative optimization algorithm on the fixed prompt and find that the results are further improved on both datasets. And for the OpenI dataset, the results of “Fixed Prompt + Iterative Opt” even outperform “Dynamic Prompt”. We believe this is due to the fact that OpenI’s corpus is relatively small and cannot provide a richer and more similar examples as context, whereas the results after using the iterative optimization method are significantly improved, and the improvement is larger than that on MIMIC-CXR, which has a much larger corpus. This further illustrates the advantages of our iterative optimization algorithm. In the end, we combine dynamic prompt and iterative optimization, our ImpressionGPT, to achieve the optimal result. The above results show that ChatGPT learns more relevant prior knowledge through the dynamic context we designed, which significantly improves the quality of the generated results. This demonstrates the feasibility of using a small number of samples to optimize the response of LLMs instead of training the model parameters. With the introduction of iterative optimization, the model can learn how to generate impression correctly with good responses, while learning how to avoid similar writing styles with bad responses. The model completes self-iterative updates in an interactive way, thus further optimizing the generated results. We also present a case study of our ablation experiments in Section II of supplementary materials.

D. Comparison with Other Methods

Table III presents a comparison of our ImpressionGPT with other radiology report summarization methods on MIMIC-CXR [17] and OpenI [18] datasets, including graph-based [4], sequence-based [6], [20], pre-trained language model-based

TABLE III

COMPARISON RESULTS OF IMPRESSIONGPT WITH OTHER PROMINENT METHODS ON MIMIC-CXR AND OPENI DATASET. **BOLD** DENOTES THE BEST RESULT AND UNDERLINE DENOTES THE SECOND-BEST RESULT.

Method	MIMIC-CXR [17]						OpenI [18]		
	R-1↑	R-2↑	R-L↑	FC-P↑	FC-R↑	FC-F1↑	R-1↑	R-2↑	R-L↑
LexRank [4]	18.11	7.47	16.87	-	-	-	14.63	4.42	14.06
PGN [6]	46.41	32.33	44.76	54.72	45.37	49.61	63.71	54.23	63.38
CGU [20]	46.50	32.61	44.98	-	-	-	61.60	53.00	61.58
TransEXT [9]	31.00	16.55	27.49	-	-	-	15.58	5.28	14.42
TransAbs [9]	47.16	32.31	45.47	56.18	49.08	52.39	59.66	49.41	59.18
CAVC [21]	43.97	29.36	42.50	-	-	-	53.18	39.59	52.86
WGSUM (LSTM) [25]	47.48	33.03	45.43	55.82	47.13	51.11	64.32	55.48	63.97
WGSUM (Trans) [25]	48.37	33.34	46.68	56.83	51.22	53.88	61.63	50.98	61.73
Jinpeng et al. [26]	<u>49.13</u>	<u>33.76</u>	<u>47.12</u>	<u>58.85</u>	<u>52.33</u>	<u>54.52</u>	<u>64.97</u>	55.59	<u>64.45</u>
ChatGPT [14]	20.48	9.96	17.02	42.83	57.44	45.23	12.03	3.70	10.52
GPT-4 [15]	19.95	8.58	15.75	41.90	58.04	44.06	11.71	3.43	9.75
Radiology-Llama2 [45]	48.34	32.40	44.27	43.00	49.86	44.44	41.85	25.69	40.87
Fine-tuned GPT-3 [28]	-	-	-	-	-	-	53.33	41.36	52.39
ImpressionGPT (Ours)	54.45	34.50	47.93	79.30	80.98	80.09	66.37	54.93	65.47

methods [9], [21], [25], [26], recent GPT-based models [14], [15], and two fine-tuned LLMs, Radiology-Llama2 [45] and GPT-3 [28]. The evaluation is based on the Rouge-1, Rouge-2, Rouge-L, and FC metrics, comparing the impressions generated by our model with handwritten impressions. As shown in Table III, our method outperforms other methods in all metrics except for the Rouge-2 score on the OpenI dataset, where it performs slightly lower than Jinpeng et al. [26] and WGSUM (LSTM) [25]. Pre-trained language models perform better than earlier studies, as they are trained with a large amount of medical text data, enabling them to learn the prior knowledge of the medical domain adequately. And our method is even better than the pre-trained language models, with the ability to construct a smaller number of relevant samples in the prompt for the LLM to learn. In the lower part of Table III, we compared the original ChatGPT [14] and GPT-4 [15] models on MIMIC-CXR and OpenI datasets. It can be seen that the original ChatGPT outperforms the GPT-4 model slightly, indicating that GPT-4 is not as effective as ChatGPT for text summarization tasks. We also compared Radiology-Llama2 [45], a large language model fine-tuned on MIMIC-CXR and OpenI datasets. Recently, the GPT-3 model released a fine-tuning interface, allowing us to utilize the most potent available GPT-3 model, Davinci [28]. To save costs, we solely conducted fine-tuning and testing on the relatively small-scale OpenI dataset. Our ImpressionGPT model surpasses Radiology-Llama2, ChatGPT, GPT-4 and the fine-tuned GPT-3. It's important to note that we achieved this improvement solely through our designed dynamic prompt and iterative optimization framework without training the parameters of the ChatGPT model. Thus, we successfully transfer generic knowledge learned by LLMs in pre-training to domain-specific tasks, such as radiology report summarization, at a lower cost than previous pre-trained language models.

In summary, we conclude that incorporating semantically similar examples as context in prompt is beneficial in using LLMs in specific domains. Moreover, the generated output of

LLMs can be optimized further with iterative interaction.

V. DISCUSSION AND CONCLUSION

In this work, we explore the applicability of Large Language Models (LLMs) in the task of radiology report summarization by optimizing the input prompts based on a few existing samples and an iterative scheme. Specifically, relevant examples are extracted from the corpus to create dynamic prompts that facilitate in-context learning of LLMs. Additionally, an iterative optimization method is employed to improve the generated results. The method involves providing automated evaluation feedback to the LLM, along with instructions for good and bad responses. Our approach has demonstrated state-of-the-art results, surpassing existing methods that employ large volumes of medical text data for pre-training. Furthermore, this work is a precursor to the development of other domain-specific language models in the current context of artificial general intelligence [16].

While developing the iterative scheme of ImpressionGPT, we noticed that evaluating the quality of responses generated by the model is a crucial yet challenging task. In this work, we employed the Rouge-1 score, a conventional metric for calculating text similarity, as the criterion for evaluating the results. We also compared the evaluation criteria using Rouge-1, Rouge-2, and Rouge-L scores and finally found that the performance is sensitive to the set threshold and achieved optimal results using the Rouge-1 score. We speculate that the differences caused by the scores used is due to the fact that the expression of words or phrases in a specific domain differs greatly from the general-domain text used for training the LLMs. Thus, using fine-grained evaluation metrics (i.e., Rouge-1) is better for evaluating the details of the generated results. We also envision that better evaluation criteria that can capture higher-level semantic information from the text will be highly needed with the advancement of LLMs.

The ethical concerns surrounding the clinical application of LLMs are a significant focus [32]. Regarding data privacy, our ImpressionGPT was tested on de-identified data,

ensuring patient privacy. Therefore, for clinical applications, employing de-identification algorithms in data pre-processing suffices to safeguard patient privacy without compromising confidentiality. Regarding model biases, our iterative optimization algorithm automatically constrains the model's generated outputs through automatic evaluations, significantly reducing the possibility of biased or fabricated outputs. Lastly, and most importantly, our model serves to provide valuable reference for clinical experts and does not replace their decision-making authority, thus leaving the final decisions in the hands of healthcare professionals or clinical practitioners.

In the future, we will continue to optimize the prompt design to better incorporate the domain-specific data from both public and local data sources while at the same time addressing the data privacy and safety concerns involved, especially in a multi-institution scenario. We are also investigating the utilization of knowledge graph in the prompt design to make it more conformed to existing domain knowledge (e.g., the relationship among different diseases). Finally, we will introduce human experts, e.g., radiologists, into the prompt optimization iterations, adding human input to evaluate the generated results when adding them to the prompts. In such a human-in-the-loop approach, we can better optimize the generated results of LLMs with decisions and opinions from human experts interactively.

REFERENCES

- [1] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.
- [2] H. Jing, "Sentence reduction for automatic text summarization," in *Sixth applied natural language processing conference*, 2000, pp. 310–315.
- [3] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using latent semantic analysis," *Journal of Information Science*, vol. 37, no. 4, pp. 405–417, 2011.
- [4] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.
- [5] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert systems with applications*, vol. 165, p. 113679, 2021.
- [6] 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)*, 2017, pp. 1073–1083.
- [7] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- [8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [9] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3730–3740.
- [10] X. Cai, S. Liu, J. Han, L. Yang, Z. Liu, and T. Liu, "Chestxraybert: A pretrained language model for chest radiology report summarization," *IEEE Transactions on Multimedia*, 2021.
- [11] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote Jr, E. T. Moseley, D. W. Grant, P. D. Tyler *et al.*, "Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives," *PLoS one*, vol. 13, no. 2, p. e0192360, 2018.
- [12] C. Ma, L. Zhao, Y. Chen, S. Wang, L. Guo, T. Zhang, D. Shen, X. Jiang, and T. Liu, "Eye-gaze-guided vision transformer for rectifying shortcut learning," *IEEE Transactions on Medical Imaging*, 2023.
- [13] C. Ma, L. Zhao, Y. Chen, L. Guo, T. Zhang, X. Hu, D. Shen, X. Jiang, and T. Liu, "Rectify vit shortcut learning by visual saliency," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [14] OpenAI, "Introducing ChatGPT — openai.com," <https://openai.com/blog/chatgpt>, [Accessed 28-08-2023].
- [15] OpenAI, "GPT-4 technical report," 2023.
- [16] L. Zhao, L. Zhang, Z. Wu, Y. Chen, H. Dai, X. Yu, Z. Liu, T. Zhang, X. Hu, X. Jiang *et al.*, "When brain-inspired ai meets agi," *arXiv preprint arXiv:2303.15935*, 2023.
- [17] A. E. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng, "Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports," *Scientific data*, vol. 6, no. 1, p. 317, 2019.
- [18] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald, "Preparing a collection of radiology examinations for distribution and retrieval," *Journal of the American Medical Informatics Association*, vol. 23, no. 2, pp. 304–310, 2016.
- [19] Y. Liu, "Fine-tune bert for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.
- [20] J. Lin, X. Sun, S. Ma, and Q. Su, "Global encoding for abstractive summarization," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 163–169.
- [21] K. Song, B. Wang, Z. Feng, R. Liu, and F. Liu, "Controlling the amount of verbatim copying in abstractive summarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8902–8909.
- [22] S. K. Karn, N. Liu, H. Schütze, and O. Farri, "Differentiable multi-agent actor-critic for multi-step radiology report summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1542–1553.
- [23] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.
- [24] S. Rezayi, H. Dai, Z. Liu, Z. Wu, A. Hebbbar, A. H. Burns, L. Zhao, D. Zhu, Q. Li, W. Liu *et al.*, "Clinicalradiobert: Knowledge-infused few shot learning for clinical notes named entity recognition," in *Machine Learning in Medical Imaging: 13th International Workshop, MLMI 2022, Held in Conjunction with MICCAI 2022, Singapore, September 18, 2022, Proceedings*. Springer, 2022, pp. 269–278.
- [25] J. Hu, J. Li, Z. Chen, Y. Shen, Y. Song, X. Wan, and T.-H. Chang, "Word graph guided summarization for radiology findings," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 4980–4990.
- [26] J. Hu, Z. Li, Z. Chen, Z. Li, X. Wan, and T.-H. Chang, "Graph enhanced contrastive learning for radiology findings summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 4677–4688.
- [27] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.
- [28] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [29] Z. Liu, A. Zhong, Y. Li, L. Yang, C. Ju, Z. Wu, C. Ma, P. Shu, C. Chen, S. Kim *et al.*, "Tailoring large language models to radiology: A preliminary approach to llm adaptation for a highly specialized domain," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2023, pp. 464–473.
- [30] Y. Shi, S. Xu, Z. Liu, T. Liu, X. Li, and N. Liu, "Mededit: Model editing for medical question answering with external knowledge bases," *arXiv preprint arXiv:2309.16035*, 2023.
- [31] H. Dai, Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, W. Liu, N. Liu *et al.*, "Auggpt: Leveraging chatgpt for text data augmentation," *arXiv preprint arXiv:2302.13007*, 2023.
- [32] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu *et al.*, "Summary of chatgpt/gpt-4 research and perspective towards the future of large language models," *arXiv preprint arXiv:2304.01852*, 2023.
- [33] Z. Liu, X. Yu, L. Zhang, Z. Wu, C. Cao, H. Dai, L. Zhao, W. Liu, D. Shen, Q. Li *et al.*, "Deid-gpt: Zero-shot medical text de-identification by gpt-4," *arXiv preprint arXiv:2303.11032*, 2023.
- [34] C. Tang, Z. Liu, C. Ma, Z. Wu, Y. Li, W. Liu, D. Zhu, Q. Li, X. Li, T. Liu *et al.*, "Policygpt: Automated analysis of privacy policies with large language models," *arXiv preprint arXiv:2309.10238*, 2023.

- [35] J. Wang, E. Shi, S. Yu, Z. Wu, C. Ma, H. Dai, Q. Yang, Y. Kang, J. Wu, H. Hu *et al.*, "Prompt engineering for healthcare: Methodologies and applications," *arXiv preprint arXiv:2304.14670*, 2023.
- [36] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.
- [37] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020.
- [38] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Auto-prompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.
- [39] E. Ben-David, N. Oved, and R. Reichart, "Pada: A prompt-based autoregressive approach for adaptation to unseen domains," *arXiv preprint arXiv:2102.12206*, 2021.
- [40] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 200–212, 2021.
- [41] M. Karthiga, S. Sountharajan, A. Bazila Banu, S. Sankarananth, E. Suganya, and B. Sathish Kumar, "Similarity analytics for semantic text using natural language processing," in *3rd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing*. Springer, 2022, pp. 239–248.
- [42] A. W. Qurashi, V. Holmes, and A. P. Johnson, "Document processing: Methods for semantic text similarity analysis," in *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2020, pp. 1–6.
- [43] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghighi, R. Ball, K. Shpankaya *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 590–597.
- [44] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2019.
- [45] Z. Liu, Y. Li, P. Shu, A. Zhong, L. Yang, C. Ju, Z. Wu, C. Ma, J. Luo, C. Chen *et al.*, "Radiology-llama2: Best-in-class large language model for radiology," *arXiv preprint arXiv:2309.06419*, 2023.

Chong Ma received his master degree of computer science from Northwestern Polytechnical University (Xi'an) in 2019. He is currently pursuing his Ph.D at school of automation of Northwestern Polytechnical University. His main research interests are deep learning, medical image analysis and natural language processing.

Zihao Wu received his B.E. degree from the School of Microelectronics, Tianjin University, Tianjin, China, in 2017 and Master degree in Electrical Engineering and Computer Science Department, Vanderbilt University, Nashville, USA, in 2020. Currently he is pursuing a PhD in computer science at University of Georgia under the supervision of Dr. Tianming Liu. His current research interests include brain inspired AI and deep learning-based medical image analysis.

Jiaqi Wang received her master degree of Statistics from Henu University (Kai'feng) in 2020. She is currently pursuing her Ph.D. at School of Computer Science of Northwestern Polytechnical University. Her main research interests are deep learning, EEG signal analysis and human-computer interaction.

Shaochen Xu received his B.S. degree in Computer Science from the University of Georgia in 2019. He is currently pursuing his PhD degree at the University of Georgia under the supervision of Dr. Tianming Liu with a research interest in deep learning, natural language processing, and vision transformers.

Yaonai Wei received his master degree of Control science and engineering from Northwestern Polytechnical University (Xi'an) in 2021. He is currently pursuing his Ph.D at school of automation of Northwestern Polytechnical University. His main research interests are deep learning, brain science and artificial general intelligence.

Zhengliang Liu received his B.A. and M.S. degrees in computer science from the University of Wisconsin-Madison and Washington University in St. Louis, in 2018 and 2021, respectively. He is a PhD student in the School of Computing, University of Georgia, Athens, GA. His areas of research include biomedical natural language processing, biomedical image analysis and the intersection of machine learning and radiation oncology.

Fang Zeng is a Data Scientist working at the Massachusetts General Hospital,

leading the development of the Large Language Model and multi-agent solutions for medical text processing in healthcare.

Prof. Xi Jiang received the B.E. degree in automation from Northwestern Polytechnical University, Xi'an, China, in 2009, and the Ph.D. degree in computer science from the University of Georgia, Athens, GA, USA, in 2016. Since 2017, he has been an Associate Professor with the School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, China. His research interests include machine learning/deep learning-based medical image analysis. Dr. Jiang was a recipient of the Li Foundation Heritage Prize, USA, for "outstanding research and contributions in the interdisciplinary field of brain science" in 2019.

Prof. Lei Guo received the Ph.D. degree from Xidian University, Xi'an, China, in 1994. He is currently a Professor with Northwestern Polytechnical University, Xi'an. His current research interests include computer vision, pattern recognition, and medical image processing.

Prof. Xiaoyan Cai is an associate professor in School of Automation, Northwestern Polytechnical University. She was a research associate in department of computing, the Hong Kong Polytechnic University, Hong Kong, from June 2009 to June 2011. She received the PhD degree from Northwestern Polytechnical University, China, in 2009. Her current research interests include document summarization, information retrieval and machine learning.

Prof. Shu Zhang received the Ph.D. degree in Computer Science from the University of Georgia, USA, in 2018. He is currently a professor at School of Computer Science from the Northwestern Polytechnical University, Xi'an, China. His research interests include biomedical image analysis, brain image analysis, deep learning and machine learning algorithms, artificial intelligence.

Prof. Tuo Zhang received the B.E. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2007 and 2015, respectively. He is currently a Research Associate with Northwestern Polytechnical University. His research interests include machine learning and medical image analysis.

Prof. Dajiang Zhu received the Ph.D. degree from Computer Science at the University of Georgia in U.S. in 2014. His current research interests include machine learning, neuroimaging and computational neuroscience.

Prof. Dinggang Shen (IEEE Fellow, AIMBE Fellow, IAPR Fellow and MICCAI Fellow) is the Founding Dean of the School of Biomedical Engineering at ShanghaiTech University. Before joining ShanghaiTech, he was a tenured Professor of Radiology, Biomedical Research Imaging Center (BRIC), Computer Science, and Biomedical Engineering at the University of North Carolina at Chapel Hill (UNC), USA. Professor Shen has been involved in the application of machine learning and artificial intelligence in medical image computing for a long time, including early brain development, early diagnosis, and the prediction of Alzheimer's disease, as well as diagnosis, prognosis and radiotherapy of brain tumor, breast cancer and prostate cancer. He is a pioneering scientist carrying out imaging AI research all over the world and is one of the first to apply deep learning to medical imaging (2012).

Prof. Tianming Liu is a Distinguished Research Professor of Computer Science at UGA. Dr. Liu's research interests are Brain Imaging, Computational Neuroscience, and Brain-inspired Artificial Intelligence. Dr. Liu has published over 400 papers in these areas, his Google Scholar citations are over 10,000+, and his H-index is 54. Dr. Liu is the recipient of the NIH Career Award and the NSF CAREER Award. Dr. Liu serves on the editorial boards of multiple journals including Medical Image Analysis, IEEE Transactions on Medical Imaging, IEEE Reviews in Biomedical Engineering, IEEE/ACM Transactions on Computational Biology and Bioinformatics, and IEEE Journal of Biomedical and Health Informatics. Dr. Liu is an elected Fellow of the American Institute for Medical and Biological Engineering.

Prof. Xiang Li is an Assistant Professor at the Massachusetts General Hospital and Harvard Medical School. He received his Ph.D. degree from the Department of Computer Science at the University of Georgia. Dr. Li's research focuses on the Artificial General Intelligence (AGI) and foundation models in healthcare, to tackle the practical challenges of applying AI in a complex clinical context. He has led the development of multiple medical imaging, language processing, EHR analysis, and multi-modal fusion projects, as well as the medical informatics systems for smart data management and AI deployment in the clinical workflow.

APPENDIX A

STUDY OF HYPER-PARAMETER SELECTION

We ablate our ImpressionGPT in five aspects of preset parameters: the number of examples in fixed or dynamic prompt (N_s), the threshold for response judgement in iterative prompt (T), the number of good and bad responses in iterative prompt (Gd and Bd), and iteration times (I). For cost-saving, we select 500 samples in the test set of MIMIC-CXR dataset for our ablation study. This allows us to complete the search for the optimal parameters of the model at a smaller cost. After determining the optimal parameters, we employ the complete test set to get the final results. Table IV~VII show the ablation results of our hyper-parameter study.

TABLE IV
ABLATION STUDY OF N_s IN FIXED AND DYNAMIC PROMPT. **BOLD** DENOTES THE BEST RESULT

Type	N_s	R-1↑	R-2↑	R-L↑	Cost
Fixed	5	29.39	13.68	23.74	\$2
	10	30.21	13.88	24.54	\$3
	15	32.57	15.97	26.13	\$3
	18	32.46	15.28	25.81	\$4
Dynamic	5	41.01	22.90	33.45	\$3
	10	42.53	24.71	35.63	\$4
	15	43.97	24.75	36.13	\$4
	18	43.76	24.34	35.63	\$5

As shown in the top part of Table IV, we first studied the results of different numbers of examples in fixed prompt, where N_s is the number of examples, and R-1, R-2, R-L, and Cost represent the Rouge-1, Rouge-2, Rouge-L scores and the rough cost, respectively. It can be seen that the best results are obtained by inserting 15 examples in the prompt. When 18 examples are inserted, the results are saturated and the input text may exceed the token limit. We continue to investigate the number of examples in the dynamic prompt, as shown in the bottom part of Table IV. The experimental results are consistent with “Fixed Prompt” that using 15 similar radiology reports as examples in the dynamic prompt performs the best results, and the corresponding Rouge-1, Rouge-2, and Rouge-L scores increase by 11.4%, 8.78%, and 10.0%, respectively, compared to the “Fixed Prompt”. This demonstrates that dynamic context constructed using similar examples can help ChatGPT to better perform tasks in specific domain.

TABLE V
ABLATION STUDY OF T IN ITERATIVE OPTIMIZATION. **BOLD** DENOTES THE BEST RESULT

Type	T	R-1↑	R-2↑	R-L↑	Cost
Dynamic	0.50	50.34	30.35	44.06	\$20
$N_s=15$	0.65	51.06	31.25	44.16	\$20
$Gd=1, Bd=n$	0.70	51.53	31.95	44.79	\$20
$I=7$	0.75	51.11	31.76	44.33	\$20

After that, we investigate the value of the metric for evaluating the Impression generated by ChatGPT in iterative optimization, which we noted as T , as shown in Table V. N_s is the similar examples in dynamic prompt, Gd and Bd represent the number of good and bad examples in iterative

prompt, and I is the upper limit of the number of iterations. We find that after using interaction optimization, Rouge-1, Rouge-2, and Rouge-L improve by 7.56%, 7.2% and 8.66%, respectively when $T = 0.7$, although the corresponding cost increases about 5 times.

TABLE VI
ABLATION STUDY OF Gd AND Bd IN ITERATIVE OPTIMIZATION. **BOLD** DENOTES THE BEST RESULT

Type	Gd	Bd	R-1↑	R-2↑	R-L↑	Cost
Dynamic	0	1	50.68	30.42	44.08	\$20
	1	0	50.31	29.87	43.75	\$20
	1	1	50.93	31.15	44.39	\$20
	1	n	51.53	31.95	44.79	\$30
$T=0.7$	n	1	51.13	31.05	44.54	\$30
$I=7$	n	n	51.46	31.77	44.25	\$40

TABLE VII
ABLATION STUDY OF I IN ITERATIVE OPTIMIZATION. **BOLD** DENOTES THE BEST RESULT

Type	I	R-1↑	R-2↑	R-L↑	Cost
Dynamic	7	51.53	31.95	44.79	\$30
$N_s=15$	12	51.91	32.78	45.85	\$50
$T=0.7$	15	52.44	33.13	46.18	\$60
$Gd=1, Bd=n$	17	53.37	33.68	46.89	\$60
	20	53.07	33.54	46.36	\$70

We continued our study on the number of good and bad examples in the iterative prompt, as shown in Table VI, in the Gd and Bd columns, where 0, 1 and n represent no examples, one example and multiple examples, respectively. We find that the model achieves optimal results when one good example and multiple bad examples are given. When more good examples are inserted in the prompt, the results are not improved, but rather there are problems with exceeding the textual limit. We believe that with one good example the direction of model optimization can be determined and with multiple bad examples the model generated responses can be further constrained from multiple perspectives. It should be noted that the number of bad examples in the iterative prompt is indefinite and can be appended all the time as long as the input limit is not exceeded.

Finally, we investigate the number of iterations, as shown in Table VII, and find that the model achieves optimal performance when the upper limit number of iterations is set to 17, and its results for Rouge-1, Rouge-2, and Rouge-L improve by 1.84%, 1.73%, and 2.1%, respectively, compared to Table VI. Continuing to increase the number of iterations does not bring any performance improvement except for the additional cost. In summary, we determine a set of optimal parameters: $N_s = 15, T = 0.7, Gd = 1, Bd = n, I = 17$.

APPENDIX B

EXAMPLES IN OUR ABLATION STUDY

Table VIII presents a test case of our ablation study from Table II of our manuscript. The upper part of Table VIII is “Case Information”, which lists the Finding and Impression sections of test report. Our task is to generate an appropriate Impression based on the Finding section. The middle part of Table VIII is “Similar Samples”, which are similar examples

TABLE VIII
EXAMPLES IN OUR ABLATION STUDY

Case Information	
Findings	Moderate cardiomegaly is re- demonstrated. The aorta is tortuous. Pulmonary vasculature is not engorged. Patchy opacities are seen in the left lung base, potentially atelectasis but infection or aspiration cannot be excluded. Streaky atelectasis is also demonstrated in the left lung base. No pleural effusion or pneumothorax is present. No acute osseous abnormality is visualized.
Impression	Patchy left basilar opacity may reflect atelectasis, but infection or aspiration cannot be excluded in the correct clinical setting.
Similar Samples	
No.1	Finding: Heart size is normal. Mediastinal and hilar contours are unremarkable. The pulmonary vasculature is not engorged. Patchy opacities are demonstrated in both lung bases which may reflect atelectasis but infection is not excluded. No pleural effusion or pneumothorax is identified. There are no acute osseous abnormalities. Impression: Bibasilar patchy opacities may reflect atelectasis but infection is not excluded in the correct clinical setting.
No.2	Finding: Cardiac silhouette size is normal. Mediastinal and hilar contours are within normal limits. Pulmonary vasculature is normal. Lungs are hyperinflated. Patchy opacities noted in the right lung base. This could reflect atelectasis but infection is not excluded. Left lung is clear. No pleural effusion or pneumothorax is demonstrated. No acute osseous abnormalities seen. Remote left-sided rib fractures are noted. Extensive degenerative changes of the left glenohumeral joint are partially imaged. Impression: Patchy right basilar opacity may reflect atelectasis but infection cannot be excluded in the correct clinical setting.
No.3	Finding: Cardiac silhouette size is normal. The mediastinal and hilar contours are unremarkable. The pulmonary vasculature is not engorged. Minimal patchy opacity is demonstrated within the right lung base. No pleural effusion or pneumothorax is present. No acute osseous abnormality is demonstrated. Impression: Patchy right basilar opacity may reflect atelectasis however infection or aspiration is not excluded in the correct clinical setting.
...	...
No.14	Finding: Lung volumes are low. The heart size is top normal with a left ventricular predominance. The mediastinal and hilar contours are unremarkable. Streaky bibasilar airspace opacities could reflect atelectasis though infection cannot be excluded. There is no pleural effusion or pneumothorax. No acute osseous abnormalities are visualized. Impression: Streaky bibasilar airspace opacities could reflect atelectasis but infection cannot be excluded. Repeat PA and lateral views with improved inspiratory effort is recommended to better assess the lung bases.
Generation Results	
Fixed	
R-1 = 45.0	Moderate cardiomegaly. Aorta is tortuous. Patchy opacities in the left lung base suggestive of atelectasis, possibly due to infection or aspiration.
Dynamic	
R-1 = 57.78	Patchy opacities in the left lung base, possibly reflecting atelectasis. Infection or aspiration cannot be excluded. Streaky atelectasis is also seen in the left lung base.
Fixed + Iteration 17	
R-1 = 52.00	Cardiomegaly and a tortuous aorta, compatible with underlying cardiovascular disease. Patchy opacities in the left lung base, likely representing atelectasis. Infection or aspiration cannot be excluded. Further evaluation may be required.
Dynamic + Iteration	
$i = 3$ (R-1 = 53.33)	Patchy opacities in the left lung base, potentially atelectasis, though infection or aspiration cannot be excluded. Streaky atelectasis is also observed in the left lung base.
$i = 5$ (R-1 = 55.00)	Moderate cardiomegaly with patchy opacities in the left lung base, potentially representing atelectasis. Infection or aspiration cannot be completely ruled out.
$i = 10$ (R-1 = 61.58)	Patchy opacities and streaky atelectasis in the left lung base, potentially due to atelectasis. Infection or aspiration cannot be excluded.
$i = 15$ (R-1 = 63.83)	Moderate cardiomegaly with tortuous aorta. Patchy opacities in the left lung base, potentially atelectasis, with streaky atelectasis. Infection or aspiration cannot be excluded in the correct clinical setting.
$i = 17$ (R-1 = 68.42)	Moderate cardiomegaly with patchy opacities in the left lung base, likely reflecting atelectasis. Infection or aspiration cannot be excluded.

we constructed in dynamic prompt. These 14 examples are selected from the corpus based on their similarity distance to the Finding section of the test report. And these similar samples (from No.1 to No.14) are ordered from closest to furthest similarity distance to finding section of test report. The lower of Table VIII, titled “Generation Results”, presents the text results and corresponding Rouge-1 scores generated by four different methods in our ablation experiments. “Fixed” represents the output results obtained using a fixed prompt. It can be observed that the scores are relatively lower, as the contextual information provided by the fixed examples may not be closely related to the actual questions (Test Case), thus lacking assistance. “Fixed + Iteration” denotes the introduction of our iterative optimization algorithm based on the fixed prompt, where we present the results in the 17th iteration. Noticeable improvements can be observed in the generated results. “Dynamic” denotes the output result obtained using a dynamic prompt. The “Similar Samples” in the middle of Table VIII demonstrates the high similarity

between these examples and the Test Case, indicating that the dynamic prompt provides valuable reference information to ChatGPT, thereby further optimizing its generated results. Lastly, “Dynamic + Iteration” represents the use of the iterative optimization algorithm on the dynamic prompt, which is our proposed ImpressionGPT. We list the results and Rouge-1 scores for iteration numbers 3, 5, 10, 15, and 17. It can be observed that as the iteration increases, the results generated by ChatGPT also improve progressively.

APPENDIX C ABLATION STUDY ON CORPUS SIZE

In the similarity search module of our model, corpus size serves as a critical factor. Therefore, we contrasted three sets of different-sized corpus using the MIMIC-CXR dataset as an example. As illustrated in Table IX, the first row represents ImpressionGPT’s similarity search within a corpus of 10,000 samples, the second row presents the experiment with a corpus expanded to 20,000, and the third row indicates the results

obtained using the entire training set as the corpus. The experimental outcomes reveal a gradual enhancement in model performance with increasing corpus size. This trend aligns with expectations, as within ImpressionGPT’s similarity search module, employing larger corpora yields instances with higher semantic similarity, providing more valuable references for the model.

TABLE IX
COMPARISON RESULTS OF USING DIFFERENT SIZE OF CORPUS IN
SIMILARITY CALCULATION. **BOLD** DENOTES THE BEST RESULT.

Corpus Size	MIMIC-CXR		
	R-1↑	R-2↑	R-L↑
100,00 Samples	48.32	29.21	41.85
200,00 Samples	49.69	30.86	43.16
122,014 (Full set)	54.45	34.50	47.93