Automatic prompt optimization for medical prompts

Wen Zirui

September 27, 2024

Abstract

The application of large language models in the medical field is currently in a phase of continuous development, exploring its boundless potential. The latest research focuses on utilizing various methods to unlock the potential of large language models. Among them, prompt engineering plays a crucial role. By carefully designing prompts, large language models can handle more complex medical tasks. In the past, prompt engineering often focused on shorter prompts. However, single short prompts often struggle to cover multiple and complex task requirements. Therefore, the demand of longer and more specific prompts to guide large language models in completing tasks is important. However, traditional manually crafted prompts suffer from various deficiencies. To address this, the paper propose a method called "Automatic Prompt Optimization" to optimize manually crafted long prompts. This method utilizes text-based gradient descent as an optimizer to tailor prompts more closely to the specific task requirements. Additionally, the paper introduce Momentum methods in optimization field, helping with selecting the best prompt. Meanwhile, this paper also proposes a Bayesian method-based reverse validation mechanism to effectively evaluate the quality of medical text question-answering. This mechanism helps to verify whether prompts can effectively guide large language models to generate accurate medical answers. Finally, this paper combines various prompt engineering methods with the RAG method, significantly improving the performance of large language models on the medical test set CMB-clin. The successful application of these methods signifies a significant advancement in the application of large language models in the medical field, providing strong support for the development of intelligent medical question-answering systems.

Keywords: LLM; Prompt Engineering; Gradient descent; Momentum; Bayes

Contents

1 Introduction

Long-term AI research has led to the emergence of a large number of large language models in recent years. The most outstanding foundational models, such as OpenAI's GPT-3 [Brown et al., 2020], GPT-4 [Achiam et al., 2023], and Llama [Touvron et al., 2023], have demonstrated astonishing capabilities in various fields [Ahn et al., 2024, Wei et al., 2022a, Bubeck et al., 2023], etc. These foundational large language models are based on the transformer framework [van Zandvoort et al., 2023], with their core ability being to predict the next word given an input. In order to improve the accuracy of next-word prediction, researchers have employed various methods, including fine-tuning the parameters of the base model [Ziegler et al., 2019], enhancing training efficiency with methods like LoRA [Hu et al., 2021], as well as utilizing prompt tuning [Li and Liang, 2021, Schick and Schütze, 2020, Brown et al., 2020]. However, the ultimate result of these methods still relies on prompts, which are the input information provided by users and serve as the key to using these foundational models. Researchers have found that by altering the composition of prompts, foundational models can be endowed with the capability to tackle complex problems. As a result, research on prompts is currently progressing rapidly.

However, in specific domains such as the medical field, solving problems can be particularly challenging due to the complex background knowledge, vast amounts of information, and numerous uncertainties. Despite the emergence of many large models fine-tuned for the medical domain, such as those by Google [Singhal et al., 2023b, Singhal et al., 2023a], many of these models are fine-tuned or pre-trained on medical datasets, requiring extensive training time and resources.

On the other hand, some researchers have explored simpler prompt techniques, such as few-shot prompting [Touvron et al., 2023] and methods like COT [Wei et al., 2022b] and its extension, TOT [Yao et al., 2024]. GPT-4 has been shown to perform well on medical challenge tasks using various prompting strategies [Nori et al., 2023a]. Meanwhile, Medprompt [Nori et al., 2023b] combines multiple prompt types for medical tasks, often employing very short prompts consisting of just a sentence or two, such as "let's think step by step." However, shorter prompts often fail to effectively stimulate the large language models.

Therefore, longer prompts can effectively break down tasks and guide the completion of complex medical tasks through more precise and detailed instructions.

However, manually crafted long prompts often encounter various issues, such as potential spelling errors. Researchers have demonstrated that large language models (LLMs) exhibit poor robustness [Zhu et al., 2023], where small perturbations in prompts can significantly affect the generated outputs. Additionally, unclear expressions in prompts may also impact the final outputs of the base model.

Therefore, the use of Automatic Prompt Optimization for crafting long prompts not only addresses minor errors but also optimizes the final outputs of the base model. Moreover, Automatic Prompt Optimization relies solely on calling APIs. Researchers have shown that LLMs can serve as prompt optimizers themselves [Yang et al., 2023, Pryzant et al., 2023, Zhou et al., 2022], searching for optimal prompts within specific solution spaces. The following figure shows the basic flowchart of prompt optimization.

In this paper, the work abstracts the prompt optimization problem into a mathematical

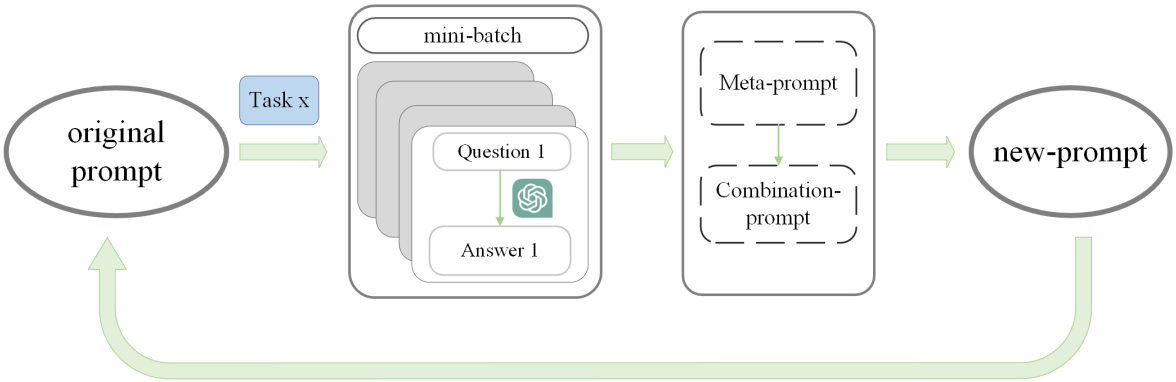

Figure 1: The process of prompt optimization

optimization problem [Guo et al., 2023, Yang et al., 2023, Tang et al., 2024]. Starting from convex optimization in operations research, the paper defines the gradient in the text space [Pryzant et al., 2023] and utilizes gradient descent in the text space [Pryzant et al., 2023, Tang et al., 2024] to search for the optimal solution of the objective function and thus the optimal prompt for the target problem. In addition, this article applies a momentum method commonly used in optimization to enhance the traditional gradient descent algorithm. For long prompt texts, the paper segments the prompts based on tasks and sentences and integrates the evaluations of segmented prompts to form a cohesive long prompt.

Additionally, when evaluating the generated text from the base model, the paper typically uses metrics such as BLEU [Papineni et al., 2002], Bertscore [Zhang et al., 2019]. However, automated evaluation metrics often suffer from serious issues. To address this, the paper employ the "expert in the loop" method during evaluation, where human experts manually score and evaluate the prompts.

Furthermore, the paper employed a Bayesian approach for prompt validation, known as the Bayes Prompt Validation method, to validate the use of long prompts in solving complex medical problems. Often, LLMs may not provide intuitive or easily understandable explanations for medical domain issues. Additionally, the diagnostic process in medicine heavily relies on past experiences and examples, aligning well with another statistical paradigm, the Bayesian approach, which calculates posterior probabilities based on prior information or distributions. The paper designed a prompt generation method based on the Bayesian approach, mimicking the diagnostic process used by human doctors to diagnose medical problems.

Finally, the paper combined the methods from the paper [Nori et al., 2023b], utilizing KNN as a few-shot prompt strategy and incorporating the methods from RAG [Lewis et al., 2020]. The paper conducted experiments on the datasets and found that the optimized prompts outperformed those without optimization. Furthermore, the research transferred the prompt optimization task to other complex downstream tasks and achieved similarly promising results.

2 Related work

2.1 from Transformer to GPT

Transformer [Vaswani et al., 2017] is a deep learning model based on attention mechanisms, originally proposed by researchers at Google and widely used in natural language processing, especially for tasks like machine translation. The Transformer model demonstrates excellent performance in handling sequence data, characterized by several key aspects:

The GPT model was proposed before the Google BERT [Devlin et al., 2018] model, and the biggest difference between them lies in their pre-training approaches. GPT adopts the traditional language model method for pretraining, where it predicts the next word using the preceding context of the word, whereas BERT utilizes bidirectional context information to predict words jointly. It is precisely due to this difference in training methods that GPT excels at natural language generation tasks (NLG), while BERT performs better in natural language understanding tasks (NLU). The pretraining task of GPT is predicting the next word, which is much more challenging than the cloze task of BERT because it requires predicting an open-ended outcome. GPT is a decoder-only structure; it does not have an encoder.

2.2 LLM in Medical domain

In the era of first-generation base models, due to limitations in model size and computational resources, domain-specific pre-trained models have significant advantages. Models such as PubMedBERT [Gu et al., 2021], BioLinkBERT [Yasunaga et al., 2022b], DRAGON [Yasunaga et al., 2022a], BioGPT [Luo et al., 2022], etc., demonstrate outstanding performance in biomedical natural language processing tasks by leveraging domain-specific data sources for self-supervised pre-training, such as the PubMed corpus and the UMLS knowledge graph. Although these models are smaller in scale and have limited computational resources, they exhibit powerful performance in handling natural language tasks in the biomedical field.

Powerful and versatile domain base models show significantly improved performance in medical challenges without the need for domain-specific pre-training. Some studies have begun to explore the performance of general base models on medical challenge problems. For example, ChatGPT-3.5 has been evaluated on questions from the United States Medical Licensing Examination (USMLE) and reached or approached the passing threshold without receiving any specialized training. In addition, GPT-4 demonstrated a passing score of over 20 points on the USMLE with simple 5-shot prompts. Other studies focus on fine-tuning base models to apply specialized medical knowledge.

Some studies have begun to explore the potential of relying on explicit medical knowledge adjustments. For example, Med-PaLM [Singhal et al., 2023a] and Med-PaLM 2 [Singhal et al., 2023b], fine-tuned 540B-parameter Flan-PaLM [Chung et al., 2022] and adjusted using instruction prompts. In Med-PaLM, the authors invited five clinical doctors to prepare fine-tuning datasets for instruction prompts. Similar to PaLM 2, Med-PaLM 2 achieved state-of-the-art performance on medical question-answering datasets, relying on comprehensive fine-tuning guided by instructions.

The paper reexamine the capabilities of general base models without relying on extensive fine-tuning. This paper explores diverse prompting strategies to best guide powerful general base models to demonstrate outstanding performance in specialized domains.

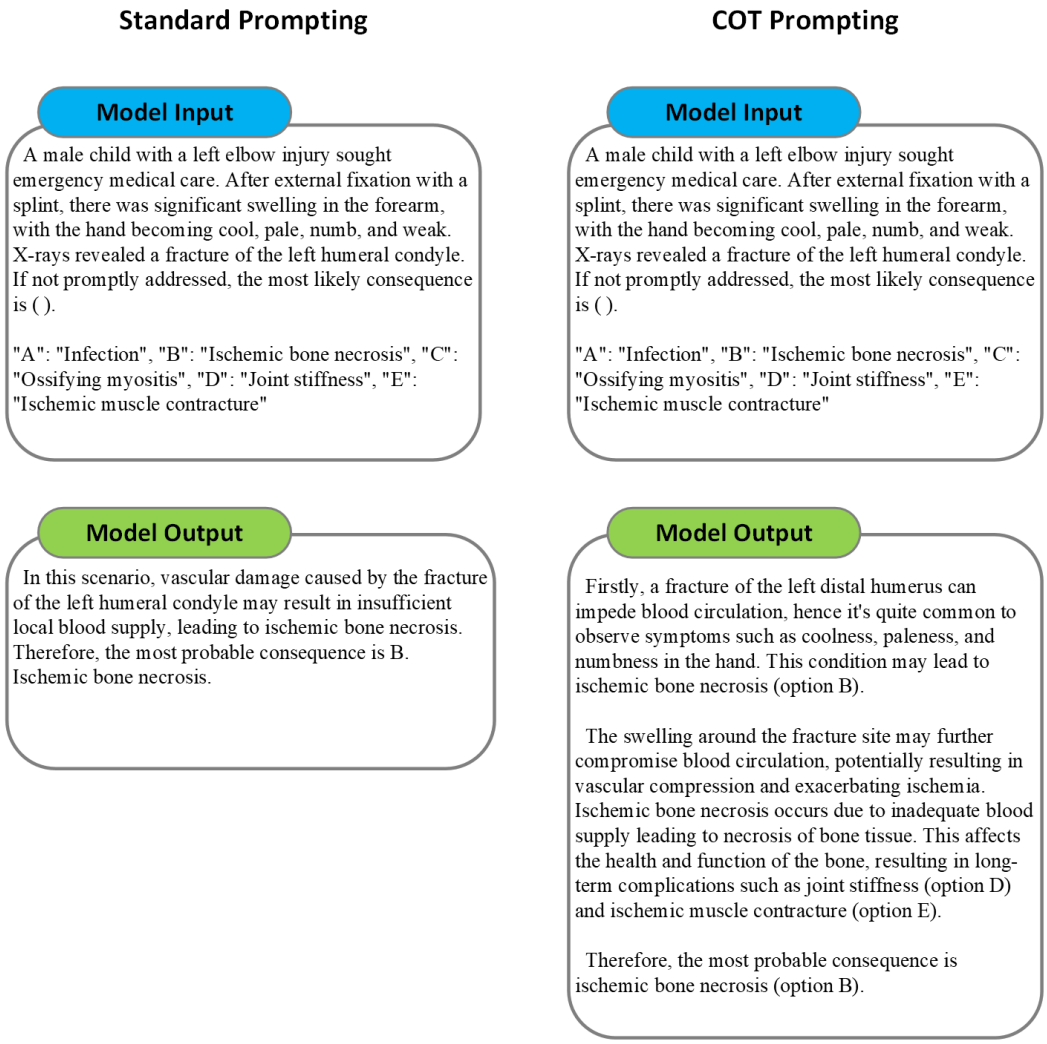

Figure 2: COT example

2.3 Prompt Engineering

Prompt engineering is a relatively new discipline applied to developing and optimizing prompts to help users effectively utilize language models in various applications and research domains. Mastering skills related to prompt engineering will assist users in better understanding the capabilities and limitations of large language models. Researchers can use prompt engineering to enhance the ability of large language models to handle complex task scenarios, such as question answering and arithmetic reasoning. Developers can leverage prompt engineering to design and develop powerful techniques that efficiently integrate with large language models or other ecosystem tools.

Prompt engineering in LLMs involves providing inputs to guide LLMs in generating outputs.

Experiments have shown that the application of prompts in vertical domains can significantly influence the performance of base large models. Moreover, employing specific prompt engineering techniques can effectively stimulate the capabilities of large language models. Here are several prompt engineering methods used in our model:

1. Zero-shot prompt: The most common form where no reference examples are provided directly to the large model when performing a task. This is considered one of the crucial scenarios for evaluating the capabilities of large models.

2. Few-shot prompt: In contrast to zero-shot, a few examples are provided in the prompt when interacting with the large model. The large language model can learn from its context and provide similar responses to the final question.

3. Role prompt: Playing a "role-playing" game with the large model, where it imagines itself as an expert in a particular field, thereby achieving better task performance.

4. Instruction prompt: Prompt in the form of instructions that effectively break down the problem, enabling the large language model to perform better by dividing the task.

5. Chain-of-thought prompt: Commonly used in reasoning tasks, guiding the large model to "think step by step" to gradually solve more challenging reasoning problems. One example is shown on figure 2.2.

2.4 automatic prompt optimization

Our work is based on LLM-based prompt optimization. LLM foundational models possess versatile capabilities, and initially, the APO paper [Pryzant et al., 2023] described how LLMs can optimize prompts based on "gradients" in the text space. The OPRO method proposed by Google Research [Yang et al., 2023] demonstrated that LLMs can act as optimizers, explaining from a mathematical perspective. Additionally, the concept of "Momentum" was introduced, which was also utilized in the GPO paper [Tang et al., 2024]. Furthermore, heuristic algorithms such as genetic algorithms [Guo et al., 2023, Yang and Li, 2023] have been employed for prompt optimization. In medical summarizing field [Yao et al., 2023], this work use "gradient" with forwards and backwards process to help clinicians summarize notes from conversations. Our work found that these optimization methods perform well for short prompts, and based on these related works, our work researched a method for optimizing long prompts.

3 Long prompt optimization

In this section, the paper presents a text-based Gradient Descent (GD) method as a prompt optimizer.

3.1 Optimization Objectives

Automatic prompt optimization aims to find an optimal prompt that exists within the space of human natural language and enables the Large Language Model (LLM) to maximize its score/accuracy on the dataset. Current methods such as OPRO [Yang et al., 2023] and

GPO [Tang et al., 2024] illustrate that this process can be represented as an optimization task. It can be formulated with the following equation:

$$p^* = \arg\max_{p \sim \mathcal{M}_O} \mathbb{E}_{\langle x,y \rangle \in \mathcal{D}} \left[L \left(\mathcal{M}_T(x; p), y \right) \right], \tag{1}$$

Explain the meaning of the symbols in this optimization function. p^*The optimal prompt that we aim to find through the optimization process. $\mathcal{M}_O$ The entire text space, representing all possible prompts within the space of human natural language. $\mathcal{M}_T(x; p)$ denotes the content of the output of the LLM under the input of the condition x and the prompt word p, L computes the difference between the output of $\mathcal{M}_T$ and the standard answer, given some evaluation criteria y between them. Our work requires the highest rated prompt under these criteria.

3.2 derivative representation

By observing the above expressions, it can be noted that solving this optimization problem has been abstracted into a convex optimization problem, although whether the optimization function is convex remains to be studied, and proving that the abstracted function is differentiable is necessary. OPRO [Yang et al., 2023] demonstrates that LLM can act as its own prompt engineer, as it can autonomously identify potentially optimal prompt words within the corresponding textual space. APO [Pryzant et al., 2023] further illustrates that LLM can analyze the disparities between the current prompt-generated outputs and the standard answers, thereby providing relevant optimization suggestions. Hence, our work defines this "derivative" as $\nabla \mathcal{F}$. By symbolically representing the derivative in the abstracted textual space, it can be expressed as the disparities or errors between the content generated by LLM through the current prompt and the standard content, utilizing these disparities generated by LLM as "derivative" information for the optimization process. In mathematics, when derivative information is unavailable, in many cases, the secant method is chosen to approximate the derivative at a certain point.

$$\frac{d}{dx} f(x) \approx \frac{f(x+h) - f(x)}{h}, \tag{2}$$

The concept of using the difference between abstraction and actualization as a derivative is very similar. LLM is essentially a black-box model, and it is impossible to access its internal parameters through API calls. Therefore, this paper provides first-order derivative information for this optimization problem through analogical derivatives.

META-PROMPT: Based on the above discussion, this paper refers to the process of LLM autonomously analyzing as the LLM optimizer and adopts the concepts from articles such as [Yang et al., 2023] and [Ye et al., 2023]. Typically, another prompt is needed to generate this gradient, called a meta-prompt. For example, for a mathematical reasoning task, the human-provided prompt could be "let's solve this problem step by step," while the meta-prompt could be "Improve the prompt to help the model better perform mathematical reasoning". Detailed meta-prompts can be found in the appendixA.1.

3.3 Optimization methods

For differentiable convex optimization problems, gradient descent is the most widely used method in the world. By moving along the direction of the negative gradient with a certain step size, the paper derive the abstracted formula for gradient descent.

$$p_{k+1} = p_k - \alpha_k \nabla \mathcal{F}, \tag{3}$$

The objective of this text is to find the optimal prompt, which is equivalent to finding the prompt that minimizes the objective function. According to the iteration formula of gradient descent and the definition of "derivative," the minimum point can be found through iteration. From the above formula, it can be seen that the update of gradient descent requires the selection of step size and direction. Next, this text will discuss step size and direction in detail.

Step Size: In traditional operations research, the step size in the iterative process is commonly referred to as the learning rate in deep learning. A good learning rate can accelerate the convergence speed of the function, while a poor learning rate can affect convergence, potentially leading to saddle points where convergence to the minimum value is not possible, or convergence speed is very slow. Therefore, for the learning rate in the text space, this text incorporates constraints such as "You are allowed to change up to X words in the current prompt" in the meta-prompt, as a method to limit the modification of the prompt's word count. Such constraints are roughly equivalent to the step size in operations research. However, it was found during experimentation that the convergence speed is very fast, with convergence typically occurring within 3 to 4 iterations for different tasks. To ensure that the selection of step size for each iteration is appropriate, the step size for each iteration needs to be adjusted. First, exponential decay is used as a method for adjusting the step size.

$$\alpha_{k+1} = \alpha_k \times \gamma^{iteration}, \tag{4}$$

Direction: The previous section introduced the derivative in the text space. In operations research, the direction of the negative gradient is defined as the direction of iteration. In the text space, the direction of the previously described "derivative" is defined as the direction of optimization iteration. Now, a method is needed to update our old prompts with the "derivative." Therefore, a prompt refinement strategy is defined.

Prompt refinement strategy: This text defines a combination prompt, for example, "modify the current prompt in conjunction with the suggestion," in combination with the meta-prompt to generate a new prompt in this way. Detailed prompts can be found in the appendixA.2.

Gradient descent has some issues when solving optimization problems in practice because of the fixed learning rate and inappropriate choices. The convergence speed of ordinary gradient descent can become slow and sometimes even fall into local optima. Therefore, the momentum method in optimization algorithms can address these issues. By considering historical gradients, the parameters are guided to converge to the optimal value more quickly, which is the basic idea of the momentum algorithm. Introducing the basic formula:

$$v_t = \beta v_{t-1} + \eta \nabla_\theta J(\theta_i), \tag{5}$$

$$\theta_t = \theta_{t-1} - v_t. \tag{6}$$

The formula indicates that each time parameters are updated, the previous velocity is taken into account. The magnitude of movement of each parameter in various directions not only depends on the current gradient but also on whether the past gradients in different directions are consistent. If a gradient consistently updates along the current direction, the magnitude of each update increases over time. Conversely, if a gradient changes continuously in a direction, its update magnitude will be attenuated. This allows us to use a larger learning rate for faster convergence, while directions with larger gradients will have their update magnitudes reduced due to momentum.

Therefore, in prompt optimization, the optimization method needs to incorporate previous "gradient" information. When using the combination prompt, our task involves merging the previous "gradients" and iterating accordingly.

3.4 Evaluation

Researchers evaluate the results generated by LLM (Large Language Models) using various metrics. This paper combines multiple evaluation metrics to automatically evaluate the content generated by LLM, and then combines this with human evaluation to assess its quality.

3.4.1 automation

LLM automatic evaluation is a common and popular evaluation method. Typically, metrics or evaluation tools such as accuracy, BLEU, ROUGE, and BERTScore [Zhang et al., 2019] are used to assess the content generated by the model. Due to its subjectivity, automatic calculation, and simplicity, most existing evaluation works adopt this evaluation protocol. Therefore, most deterministic tasks, such as natural language understanding and mathematical problems, usually use this evaluation protocol. In the field of Automatic Prompt Optimization, articles like [Ye et al., 2023] use LLM-eval to score or evaluate generated content from multiple dimensions using LLM. In this paper, we use BERTScore to score the generated text.

BERTScore utilizes a pre-trained BERT model to assess the similarity between generated text and reference text. It is a widely used evaluation metric for text comparison tasks. BERTScore leverages the contextual embeddings of the BERT model and matches candidate sentences and reference sentences using cosine similarity.

Write the formula for cosine similarity:

$$cos_{similarity} = \frac{\mathbf{x}_i^\top \hat{\mathbf{x}}_j}{||\mathbf{x}_i||||\hat{\mathbf{x}}_j||}, \tag{7}$$

First, embeddings are computed for both the generated text and the reference text. Then, the BERT model is used to calculate the similarity scores between these embeddings. These scores reflect the semantic similarity between the two texts. The specific idea is straightforward: tokenize

both the generated and reference sentences using word piece, extract features using BERT for each token, and then calculate the dot product for each word of the two sentences to obtain a similarity matrix. Based on this matrix, we can compute the cumulative maximum similarity score for both the reference and generated sentences, normalize them, and derive BERTScore precision, recall, and F1 scores.

$$R_{\text{BERT}} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}. \tag{8}$$

Meanwhile Bertscore adds importance weights, using the formula:

$$\text{idf}(w) = -\log \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\left[w \in x^{(i)}\right], \tag{9}$$

$$R_{\text{BERT}} = \frac{\sum_{x_i \in x} \text{idf}\left(x_i\right) \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{x_i \in x} \text{idf}\left(x_i\right)}. \tag{10}$$

3.5 Expert in the loop

Undeniably, the automatic evaluation of the LLM is still lacking in some aspects, unable to fully comprehend the generated content like humans do. Therefore, this paper introduces an Expert in the Loop into the rating system, inviting experts in the field of medicine to evaluate the generated content of the model. The rating system categorizes the generated content into six different types, and human experts classify each sentence accordingly. We refer to articles such as [van Zandvoort et al., 2023] and [Liévin et al., 2023] for the classification of model-generated content. The diagram in Figure 3.5 illustrates the method we used for Expert in the Loop.

3.6 Long prompt

In addressing the issue of long prompt words during gradient descent (GD), optimizing the entire prompt word space often proves time-consuming and labor-intensive. Moreover, the answers tend to converge towards incorrect directions. Therefore, in the process of optimizing long prompt words, following the advice provided by the meta-prompt, we optimize prompt words based on different pairs. Each time, only a few sets are selected for optimization. This approach is adopted because previous methods applied to long prompts have encountered the following situations.

Over-optimization towards specificity: The gap between the generated text and the standard answer is too large, resulting in overly specific modifications that render the prompt words lacking in generalizability. Our work mitigates this by adjusting the step size and modifying the content of the 'meta-prompt' to reduce such occurrences, limiting the changes in the number of words modified each time to achieve better results.

Over-optimization towards abstraction: The reason for over-optimization towards abstraction lies in selecting a relatively large batch size for each group, which fails to identify common characteristics. Thus, adjusting the appropriate batch size as the initial training set helps in addressing this issue.

USMLE [test:731] Three days after induction chemotherapy, a 9-year-old boy is admitted to the hospital for fever and neutropenia. He was initially diagnosed with t(9;22)-positive acute T cell lymphoblastic leukemia following workup for fatigue and pancytopenia. He is fully conscious. His blood pressure is 110/65 mm Hg, temperature is 39.5°C (103.1°F), pulse is 98/min, and respiratory rate is 14/min. Physical examination including examination of the central venous catheter site shows no source of infection. Broad-spectrum antibiotics are initiated. Which of the following factors about this patient most strongly indicates a poor prognosis?
A) Age **B) Cytogenetics** C) Fever and neutropenia D) Initial presentation

✓ Let's use step by step inductive reasoning, given the medical nature of the question.
The first thing that stands out is that the patient is 9 years old. In general, children have a better prognosis than adults when it comes to leukemia.
The next thing to consider is the patient's cytogenetics. The fact that the patient has t(9;22)-positive acute T cell lymphoblastic leukemia indicates that the patient has a specific chromosomal abnormality that is associated with a poor prognosis.
The next thing to consider is the patient's fever and neutropenia. Fever and neutropenia are both signs of infection, and infection is a serious complication in leukemia patients, particularly in those who are immunocompromised. Finally, we need to consider the patient's initial presentation. The fact that the patient was initially diagnosed with leukemia following workup for fatigue and pancytopenia indicates that the patient's leukemia was advanced at the time of diagnosis.
All of these factors indicate a poor prognosis for this patient.
Therefore, among A through D, the answer is B, cytogenetics.

✗ Let's think step by step like a medical expert. The first step is to gather information about the patient. In this case, we are given the following information: -The patient is 9 years old -The patient has t(9;22)-positive acute T cell lymphoblastic leukemia -The patient is currently experiencing fever and neutropenia
The next step is to analyze the information and look for any patterns or clues that could help us answer the question.
In this case, the most important factor that indicates a poor prognosis is the patient's age. Children under the age of 10 have a much higher chance of relapsing after treatment and a lower chance of achieving remission.
Therefore, among A through D, the answer is A.

✗ Let's derive the differential diagnosis step by step:
1) The patient has fever and neutropenia.
2) The patient is 9 years old.
3) The patient has t(9;22)-positive acute T cell lymphoblastic leukemia.
4) The patient's initial presentation was fatigue and pancytopenia.
The most likely cause of the patient's fever and neutropenia is infection. The patient's age, cytogenetics, and initial presentation are all risk factors for infection.
Therefore, the factor that most strongly indicates a poor prognosis is the patient's age.
Therefore, among A through D, the answer is A.

Pattern A	Correct reasoning step	Pattern D	Incorrect reasoning step
Pattern B	Correct recall of knowledge	Pattern E	Incorrect or insufficient knowledge
Pattern C	Correct reading comprehension	Pattern F	Incorrect reading comprehension

Figure 3: Expert in the loop

4 Forward validation of Bayes prompt

4.1 Why use Bayesian methods

Although large language models (LLMs) possess relatively strong explanatory analytical capabilities in the medical field, their explanations may not be intuitive enough in real medical scenarios. Physicians typically make diagnoses based on past experience or previous cases, which aligns with the principles of Bayesian inference. The Bayesian approach typically utilizes prior information and prior distributions to calculate posterior probabilities.

Physicians usually start with the patient's chief complaint and combine it with important physical examination findings to make a diagnosis. Therefore, we have designed a prompt word based on the Bayesian method to better simulate the diagnostic process of physicians in real life when diagnosing patients' illnesses.

4.2 How Bayesian prompts are verified

The positive verification method suggested by Bayesian inference, combined with the real situation during doctor-patient consultations, typically begins with the patient's chief complaint. The doctor then proceeds to conduct a series of examinations, including physical examinations, and ultimately makes an initial diagnosis of the patient's illness. By incorporating the Bayesian formula, the paper design prompt words similar to the following:

The paper try to analyze our above prompts through Bayesian formulas. The most basic Bayesian formula is given first.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}, \tag{11}$$

$$P(A|B_1, B_2, ..., B_n) = \frac{P(A) \cdot P(B_1|A) \cdot P(B_2|A, B_1) \cdot ... \cdot P(B_n|A, B_1, B_2, ..., B_{n-1})}{P(B_1) \cdot P(B_2) \cdot ... \cdot P(B_n)}, \tag{12}$$

The use of Bayesian methods, which are usually used to solve medical problems with multiple choice questions, employs Bayesian formulas to try to obtain the best option, therefore, in this paper, it define events such as A_1, A_2, A_3, A_4, etc. as the selection of option A_1, A_2, etc. under condition 1, and organize the information obtained into conditions 1, 2, 2 Condition 3, etc. Finally calculate the probability of choosing the option under all conditions. Simplifying the above (12)

$$P(A_i|B_l, B_{C-l}) = \frac{P(A_i) \cdot P(B_l|A_i) \cdot P(B_{C-l}|Ai, B_l)}{P(B_l) \cdot P(B_{C-l})} \tag{13}$$

$$= P(A_i) \cdot P(B_l|A_i) \cdot P(B_{C-l}|Ai, B_l). \tag{14}$$

Through the Bayesian formula, it is possible to visualize the results of judging the disease in combination with other conditions when certain important conditions are used as the main basis.

Bayes Prompt

Extract the patient's information from the text as conditions, in the form of conditions 1,2,3,4, condition 1 is the patient's chief complaint, and the rest are the other conditions, and list the questions as well as the options, do not give the correct answer. Step 1, you now have only conditions 1,2,3 and are not allowed to consider any other conditions, give a number between 0 and 0.5 indicating the probability of choosing each option given the information in condition 1 only, for each option explain the reason for giving that number, if the probability is very small give a number closer to 0, don't give it, if the probability is higher give a number closer to 0.5. Remember to prioritize serious diseases. In the second step, combine the knowledge you have with the perspective of a medical professional and tell me the possible reasons for each condition, don't leave out any condition as well as the data. Give a number between 0 and 1 that indicates the probability of choosing each option given all the conditions of the patient, for each option explain the reason for giving the number. Explain for each option the reason for giving that number, analyze each condition in as much detail as possible, if the probability is small give a number closer to 0 than 0, if the probability is large give a number closer to 1. Finally multiply these two probabilities obtained for each option and select the option with the highest probability.

Figure 4: Bayes prompt

4.3 Problems in the design of Bayesian prompts

Meanwhile, when designing Bayesian prompts, it is necessary to consider the following questions in order to achieve better results.

4.3.1 How to get the probability

In our research, probabilities are provided through the common sense stored in the LLM. However, the probabilities provided by LLM are often not accurate enough. It often lacks data support and strict mathematical formulas. In addition, each hospital usually has its own database. Trying to provide relevant probabilities by connecting to external databases.

4.3.2 How to define the primary condition

In the process of designing prompts, another issue that arises is how to define which conditions are primary and which are secondary. To address this issue, our work consulted experts in the medical field. Based on their recommendations, adjustments were made to the primary conditions and a reasonable allocation was determined. By inquiring about the patient's chief complaints and physical examination findings, primary conditions were identified, while the remaining conditions were categorized as secondary. This approach was used to design prompts.

4.4 More Discussions

Our work extensively utilizes the Bayesian formula to solve multiple-choice questions. In generative questions, Bayesian prompts also exhibit excellent performance. In medical cases, "differential diagnosis" is an area where Bayesian prompts can be effectively utilized.

5 Experiment

5.1 Datasets

The evaluation benchmark of this article is conducted on several outstanding Chinese datasets and English datasets.

- **CMB** [Wang et al., 2023]: The dataset is a comprehensive medical model evaluation dataset, consisting of 6 major categories, 28 subcategories, and detailed explanations. Meanwhile, it includes 74 complex clinical cases for assessing the comprehensive capabilities of LLM.
- **CMExam** [Liu et al., 2023]:CMExam is a dataset derived from the Chinese National Medical Licensing Examination. It comprises over 60,000 multiple-choice questions and annotations for five additional attributes, including disease categories, clinical departments, medical disciplines, competency domains, and question difficulty levels.
- **MedQA** [Jin et al., 2020]:MedQA is a medical text question-answering dataset presented in the format of multiple-choice questions. The questions are sourced from examinations conducted by medical boards in the United States, mainland China, and Taiwan, designed to assess physicians' professional knowledge and clinical decision-making abilities. The questions cover a wide range of topics, and answering them typically requires a deep understanding of relevant medical concepts. The dataset comprises a total of 61,097 questions, with 12,723 questions in English, 34,251 questions in simplified Chinese, and 14,123 questions in traditional Chinese.
- **PubMedQA** [Jin et al., 2019]:PubMedQA is a dataset containing biomedical research questions. It requires answers in the form of yes, no, or maybe, based on the context provided by PubMed abstracts. The dataset has two settings: "with reasoning required" and "without reasoning required." In the "without reasoning required" setting, long answers including explanations from the abstracts are provided, and we report models can only use the context from the abstracts when answering questions. The dataset consists of 500 questions in total.

The paper use the CMB dataset as an example to visualize the dataset, as shown in the following figure.

5.2 Experimental Methods

Before proceeding to step 3, there are two initialization methods for the long prompt that needs optimization in our work. The first method involves manual crafting by experts in the

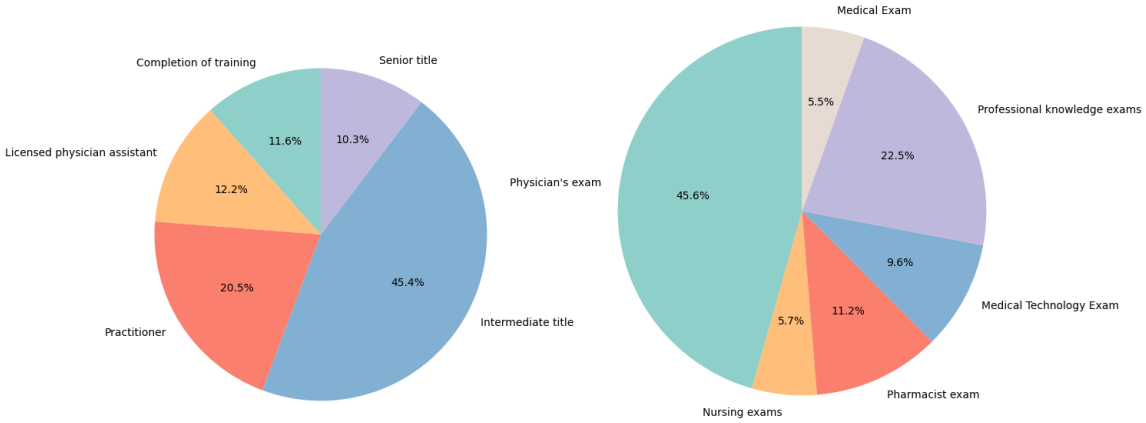

Figure 5: CMB dataset

medical field, who write long prompts for the corresponding questions following the guidelines proposed by [Bsharat et al., 2023]. During the prompt writing process, the problems to be addressed are broken down step by step. The second method utilizes LLM (Large Language Model) automatic generation, as indicated by [Zhou et al., 2022]. LLMs can serve as prompt engineers to automatically generate prompts. We use LLM to generate long prompts for solving specific medical issues.

After initializing the prompts, the next step involves optimizing them. Before using the meta-prompt, it's necessary to identify areas where the original generated results are incorrect. In this step, we evaluate the generated results. During the evaluation process, we segment the generated text and compare it with the true answers. Texts with low scores are then fed into the meta-prompt for iterative optimization.

5.3 Experimental algorithm steps

6 Results

6.1 Experimental setup

The experimental setup of this paper was conducted locally, utilizing the OPENAI API interface. The model used was "gpt-3.5-turbo-16k". Both "meta-prompt" and "combination-prompt" were experimented with using this model. The initial learning rate (step size) for the prompt was set to 200. All initial prompt settings can be found in Appendix A.2.

Algorithm 1 gradient descend algorithm

Input: p,x,y

Output: p^*

while end condition **do**
  choose a $\mathcal{K}$ size
  for each $\mathcal{K}_i \in \mathcal{M}$ **do**
    use p^i generate $\nabla\mathcal{F}$
  end for
  choose an α_k
  $\alpha_k = \alpha_{k-1} \times \gamma^{iteration}$
  $p_{k+1} = p_k - \alpha_k \nabla\mathcal{F}$
end while
return Outputs

Algorithm 2 momentum algorithm

Input: p,x,y

Output: p^*

while end condition **do**
  choose a $\mathcal{K}$ size
  for each $\mathcal{K}_i \in \mathcal{M}$ **do**
    use p^i generate $\nabla_p J(p)$
  end for
  calculate $v_t = \beta v_{t-1} + \eta \nabla_p J(p_i)$
  $p_t = p_{t-1} - v_t$
end while
return Outputs

6.2 Results

In this section, the paper present the performance of our method on different datasets following the algorithm optimization described in the previous section. First, we demonstrate the performance of the CMBclin dataset. Our work is manually divided into four distinct tasks: cure, check, diagnose, and else. After 5 iterations, the newly generated prompts showed significant improvements in Precision, Recall, and F1-score compared to the previous prompts, as evaluated using Bertscore on the test set.

Themes	Precision	Recall	F1
cureold	0.602049	0.671040	0.633757
cure	**0.644055**	**0.696207**	**0.668422**
checkold	0.633994	0.701150	0.664763
check	**0.660891**	**0.712418**	**0.685074**
diagnoseold	0.654446	0.694288	0.672382
diagnose	**0.673877**	**0.707131**	**0.689273**
elseold	0.590821	0.687988	0.633317
else	**0.631141**	**0.704557**	**0.664556**

Table 1: 4 tasks results on CMBclin data

This paper conducts comparative experiments between the optimized prompts generated in this study and several other sets of prompts derived from different cognitive chains. Table 6.2 presents several examples of cognitive chain prompts used in this study. These prompts were selected based on previous research in the medical domain.

COT	content
COT#1	let's think step by step
COT#2	Let' s think step by step like a medical expert
COT#3	Let' s follow a Bayesian step by step approach

Table 2: some COT used in the paper

Next, the paper shows the performance of the MedQA dataset. The experiments revealed that using conventional COT alone yielded poor results. However, text comparison experiments showed that the performance improved by 50% when using the new long prompt compared to conventional COT. Moreover, after iterations, the accuracy of the new prompt on the MedQA dataset increased by 4%. Additionally, research indicates that prompts generated with more complex methods tend to have longer average content.

Meanwhile, the paper tests the effect of Bayesian cue words. However, in testing, it is found that sometimes the generated content does not appear as given by the prompt word. The paper has counted the following table:

prompt	acc	average length
COT#1	40%	170
COT#2	47%	183
COT#3	33%	193
old	60%	300
new	**64%**	**450**

Table 3: Performance on MedQA dataset

problem	frequency
No display conditions	50%
Incorrect information on display conditions	50%
Problems with the reasoning process	20%
Appearance of gibberish	10%
Derivation problems due to faulty knowledge base	80%

Table 4: Problems occur in Bayes prompt

Because this is a subset of data selected from the entire MedQA dataset, some questions may not fully comply with the task requirements set by the prompt words. As a result, the accuracy of answering on the entire dataset is not satisfactory. However, by selecting specific tasks, Bayesian prompt words can address some of these issues.

Limitations. This study focuses on optimizing long prompt words and employing prompt engineering techniques to develop the performance of base models in the vertical domain of healthcare within Large Language Models (LLM). As a method that does not require GPU or other computational resources, solely optimizing the capabilities of base models through LLM is a time-saving and low-cost approach.

Although this method is time and cost-effective, it has only been tested on a small portion of the dataset. Due to the time-consuming nature of calling API interfaces and iterations, there are insufficient resources to conduct extensive testing on other datasets. Therefore, whether this method can be transferred to other downstream tasks of LLM is worthy of further research.

Additionally, in the design of Bayesian prompt words, this paper lacks rigorous formulas and data for calculating the given probabilities, leading to some deviations in the final results. This aspect also requires further research and supplementation in the future.

7 Conclusion

This paper demonstrates the excellent answering capability of Large Language Models (LLM) in the medical domain by utilizing a method based on the intrinsic capabilities of the base model itself. This method eliminates the need for fine-tuning the base model or prompt tuning.

Through the study of prompt engineering in LLM, the paper finds that although there are many methods for optimizing prompt words, they either involve semantic rewriting of long prompt words or rely on text gradient-based optimization of short prompt words. Experimental results reveal that long prompt words often guide LLM in logical deduction or completing specific downstream tasks. Therefore, the paper proposes a method for optimizing long prompt words based on text space gradient descent, which can be specifically targeted for optimizing prompt words in medical tasks. Furthermore, leveraging the methods of medprompt, the paper enhances the capabilities of LLM by generating templates and utilizing few-shot and RAG methods. Additionally, the paper designs Bayesian prompt words, which mimic the thought process of doctors in handling problems under real-life circumstances.

Although this research focuses on utilizing the capabilities of base models without extensive GPU resources, it is believed that there will be more fine-tuning methods or updated base models emerging. The paper believes that in the medical field, LLM will demonstrate infinite possibilities.

Acknowledgement

Appreciation for my parents, gratitude towards my friends, thanks to the school. I am especially grateful to Prof. Gan Wensheng, who has guided me with great care, and without his help, I would not have been able to write this article.

A Appendix

A.1 Used prompts

Meta-prompt

> Your task is to point out the problems with the current prompt based on the wrong examples. The current prompt is: current But this prompt gets the following examples wrong.You should analyze the differences between wrong predictions and ground truth answers, and carefully consider why this prompt led to incorrect predictions. Below are the task examples with Queston, Wrong prediction, and Ground truth answer. error analyze

combine prompt

> Your task is to integrate the problems in the previous prompt and the current prompt. Below are the problems that arose from the previous prompts.previous problem Carefully analyze the previous prompts and write a new improved prompt to replace old prompt You are allowed to change up to step words in the current prompt. Wrap the new prompt with START and END.

A.2 original prompts

prompt type	original prompt
cure	拿到病历文本信息后，根据主要症状（现病史，包括症状和检查结果）分析可能的疾病，尽量用一种疾病解释所有症状。如果有不止一种疾病能满足所有症状，结合其他风险因素（年龄，既往史，家族史等）找出最有可能的。优先考虑常见病。根据最有可能的诊断，简述治疗原则。
diagnose	拿到病历文本信息后，根据主要症状：现病史，包括症状和检查结果，分析可能的疾病，尽量用一种疾病解释所有症状。如果有不止一种疾病能满足所有症状，结合其他风险因素（年龄，既往史，家族史等）找出最有可能的。优先考虑常见病，再通过分析病人的病情，结合最有效的信息，简述 3 个该病人的鉴别诊断。
check	让我们一步一步分析文本内容，分析患者体格检查的异常之处，包括一般情况、生命体征、特殊体征（如皮肤、头部、颈部、胸部、心脏、腹部、四肢等）的检查结果。同时分析实验室及影像检查的异常之处，如患者进行的各类实验室检查（如血液检查、尿液检查等）及影像学检查（如 X 光、CT、MRI 等）结果。
else	你现在是一名医生，具备丰富的医学知识和临床经验。你擅长诊断和治疗各种疾病，能为病人提供专业的医疗建议。在你思考这个问题的时候，首先要梳理提供的文本内容分析提供的信息，并且根据提出的问题对应到相关的领域，如果是选择题，先根据之前整理出来的信息，对于每一个选项逐步思考能否选择该选项及其理由。最后告诉我答案。如果不是选择题，那么需要你根据整理出来的信息逐步分析，最后回答问题。
bayes	从文本中提取患者的信息作为条件，形如条件 1,2,3,4，条件 1 是患者的主诉，剩下的为其他条件，并列出问题以及选项，不要给出正确答案第一步，你现在只有条件 1，2,3，不允许考虑其他任何条件，给出一个 0 到 0.5 之间的数字，表示只在条件 1 的信息下选择每个选项的概率，对每个选项都解释给出该数字的理由，如果概率很小就给一个比较接近 0 的数，不要给 0，如果概率比较大就给一个比较接近 0.5 的数。记住要把严重的疾病优先考虑第二步，结合你拥有的知识，以一个专业医生的视角，告诉我每一个条件可能的原因，不要漏掉任何一个条件以及数据。给出一个 0 到 1 之间数字，表示在患者的所有条件下选择每个选项的概率，对每个选项都解释给出该数字的理由，尽可能的详细分析每一个条件，如果概率很小就给一个比较接近 0 的数，不要给 0，如果概率很大就给一个比较接近 1 的数最后将每个选项得到的这两个概率相乘，并选出概率最大的选项

Table 5: original prompts the paper used

A.3 optimized prompt

As a highly skilled and experienced doctor, you possess a wealth of medical knowledge and clinical expertise. Your proficiency lies in diagnosing and treating various diseases, enabling you to provide professional medical advice to patients. When approaching a problem, it is crucial to carefully analyze the information provided in the text and correlate it with the relevant field based on the questions posed.To ensure clarity and accuracy in your responses, it is important to address the following issues that have been identified in previous prompts:1. Clearly state the task or question that needs to be answered. Specify the objective of the analysis and the specific questions that require a response. This will provide a clear direction for your evaluation and prevent confusion or incorrect answers.2. Provide sufficient context and information for the examples. Include relevant details about the patient's medical history, physical examination findings, and any diagnostic tests or results. This additional information will enable you to make accurate diagnoses and formulate appropriate treatment plans.3. Specify the format of the expected answers. Clearly indicate whether the responses should be in the form of multiple-choice options or written explanations. Provide guidelines on the necessary components to include in the answers, such as relevant symptoms, diagnostic criteria, and treatment options.By addressing these concerns, you can enhance the prompt and provide clearer instructions and context for the model. This will result in more accurate and informative answers, enabling you to effectively utilize your medical expertise and provide valuable insights to patients.Now, with these improvements in mind, carefully analyze the provided text and answer the questions accordingly. Remember to consider the relevant information, evaluate each option (if applicable), and provide a well-reasoned response.

A.4 Mathematical Deduction of Momentum

A.4.1 Part 1

From a mathematical perspective, momentum method is first understood by introducing the exponentially weighted moving average. For a given hyperparameter $0 \leq \gamma < 1$, the variable y_t at the current time step t is a linear combination of the variable y_{t-1} from the previous step $t-1$ and another variable x_t at the current time step.

$$y_t = \gamma y_{t-1} + (1-\gamma)x_t, \tag{15}$$

Expanding y_t:

$$\begin{aligned} y_t &= (1-\gamma)x_t + \gamma y_{t-1} && (16)\\ &= (1-\gamma)x_t + (1-\gamma)\cdot\gamma x_{t-1} + \gamma^2 y_{t-2} && (17)\\ &= (1-\gamma)x_t + (1-\gamma)\cdot\gamma x_{t-1} + (1-\gamma)\cdot\gamma^2 x_{t-2} + \gamma^3 y_{t-3} && (18)\\ &... \end{aligned}$$

let $n = 1/(1-\gamma)$. Then $(1-1/n)^n = \gamma^{1/(1-\gamma)}$. Because,

$$\lim_{n\to+\infty} = \left(1-\frac{1}{n}\right)^n = \exp(-1) \approx 0.3679, \tag{19}$$

So as $\gamma \to 1$, $\gamma^{1/(1-\gamma)} = \exp(-1)$, such as $0.95^{20} \approx \exp(-1)$. If treat $\exp(-1)$ as a relatively small number, the formula can neglect all terms containing $\gamma^{1/(1-\gamma)}$ and coefficients of higher orders than $\gamma^{1/(1-\gamma)}$. For example, when $\gamma = 0.95$

$$y_t \approx 0.05\sum_{i=0}^{19} 0.95^i x_{t-i}, \tag{20}$$

Therefore, in practice, y_t is often regarded as the weighted average of the recent $1/(1-\gamma)$ time steps of x_t. For example, when $\gamma = 0.95$,y_t can be seen as the weighted average of the last 20 time steps of x_t; When $\gamma = 0.9$, $y_t$can be seen as the weighted average of the last 10 time steps of x_t. Moreover, the closer the x_t values are to the current time step t, the greater the weight they receive.

In momentum methods, we transform the velocity variable as follows:

$$v_t \to \beta v_{t-1} + (1-\beta)\left(\frac{1}{1-\beta}\alpha\nabla_\theta J(\theta)\right), \tag{21}$$

From the form of exponentially weighted moving average, deducing that the velocity variable v_t is effectively a sequence of:

$$\left\{\frac{\alpha\nabla_\theta^{t-i}(\theta)}{1-\beta}, i = 0, 1, ..., \frac{1}{1-\beta} - 1\right\}, \tag{22}$$

By performing exponential weighted moving averages, in other words, compared to stochastic gradient descent with mini-batches, the update of the independent variable at each time step in momentum methods is approximately equivalent to taking the exponentially weighted moving average of the updates from the most recent $1/(1-\beta)$ time steps of the former, then dividing by $1-\beta$. Therefore, in momentum methods, the magnitude of the movement of the independent variable in each direction depends not only on the current gradient but also on whether the past gradients in each direction have been consistent.

A.4.2 Part 2

Introducing a quadratic convex function:

$$h(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{x}^\top \mathbf{c} + b. \tag{23}$$

This is a standard quadratic function. For a positive definite matrix $\mathbf{Q} \succ 0$, meaning a matrix with positive eigenvalues, the minimum is $\boldsymbol{x}^* = -\mathbf{Q}^{-1}\boldsymbol{c}$, and the minimum value is $b - \frac{1}{2}\mathbf{c}^\top \mathbf{Q}^{-1}\mathbf{c}$. Therefore, rewriting h as:

$$h(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{Q}^{-1}\mathbf{c})^\top \mathbf{Q}(\mathbf{x} - \mathbf{Q}^{-1}\mathbf{c}) + b - \frac{1}{2}\mathbf{c}^\top \mathbf{Q}^{-1}\mathbf{c}. \tag{24}$$

The gradient is given by $\partial_{\mathbf{x}} f(\mathbf{x}) = \mathbf{Q}(\mathbf{x} + \mathbf{Q}^{-1}\mathbf{c})$. In other words, it is obtained by multiplying the distance between $\mathbf{x}$ and the minimum by $\mathbf{Q}$. Therefore, momentum methods involve a linear combination of $\mathbf{Q}(\mathbf{x}_t + \mathbf{Q}^{-1}\mathbf{c})$.

Because $\mathbf{Q}$ is positive definite, it can be decomposed as $\mathbf{Q} = \mathbf{O}^\top \mathbf{\Lambda} \mathbf{O}$ into an orthogonal matrix $\mathbf{O}$ and a diagonal matrix of positive eigenvalues $\mathbf{\Lambda}$. This allows variables to be changed from $\mathbf{x}$ to $\mathbf{z} := \mathbf{O}(\mathbf{x} + \mathbf{Q}^{-1}\mathbf{c})$, resulting in a highly simplified expression:

$$h(\mathbf{z}) = \frac{1}{2}\mathbf{z}^\top \mathbf{\Lambda}\mathbf{z} + b'. \tag{25}$$

Here, $b' = b - \frac{1}{2}\mathbf{c}^\top \mathbf{Q}^{-1}\mathbf{c}$. Since $\mathbf{O}$ is just an orthogonal matrix, it does not significantly perturb the matrix gradient. The gradient descent expressed in terms of $\mathbf{z}$ becomes:

$$\mathbf{z}_t = \mathbf{z}_{t-1} - \mathbf{\Lambda}\mathbf{z}_{t-1} = (\mathbf{I} - \mathbf{\Lambda})\mathbf{z}_{t-1}, \tag{26}$$

The key fact in this expression is that gradient descent does not mix between different feature spaces. In other words, if represented in the eigenvalues of $\mathbf{Q}$, the optimization problem proceeds in a coordinate-wise manner. In the context of momentum methods, we express it as:

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + \mathbf{\Lambda}\mathbf{z}_{t-1} \tag{27}$$

$$\mathbf{z}_t = \mathbf{z}_{t-1} - \eta\left(\beta \mathbf{v}_{t-1} + \mathbf{\Lambda}\mathbf{z}_{t-1}\right) \tag{28}$$

$$= (\mathbf{I} - \eta\mathbf{\Lambda})\mathbf{z}_{t-1} - \eta\beta\mathbf{v}_{t-1}. \tag{29}$$

The above process demonstrates that gradient descent with momentum for quadratic functions can be decomposed into optimization along the coordinates of the eigenvectors of the quadratic matrix in order of eigenvalues. Next, let's minimize the function $f(x) = \frac{\lambda}{2}x^2$ to illustrate this specifically.

For gradient descent, we have:

$$x_{t+1} = x_t - \eta\lambda x_t = (1 - \eta\lambda)x_t. \tag{30}$$

For optimization with $|1 - \eta\lambda| < 1$, convergence occurs at an exponential rate, as $x_t = (1 - \eta\lambda)^t x_0$ after t steps. This illustrates the improvement in convergence rate as the learning rate η is increased until $\eta\lambda = 1$. For $\eta\lambda > 2$, the optimization problem will diverge.

To analyze the convergence of momentum, let's first rewrite the update equations using two scalars: one for x and another for momentum v. This yields:

$$\begin{bmatrix} v_{t+1} \\ x_{t+1} \end{bmatrix} = \begin{bmatrix} \beta & \lambda \\ -\eta\beta & (1-\eta\lambda) \end{bmatrix} \begin{bmatrix} v_t \\ x_t \end{bmatrix} = \mathbf{R}(\beta, \eta, \lambda) \begin{bmatrix} v_t \\ x_t \end{bmatrix}. \tag{31}$$

Using $\mathbf{R}$ to represent the convergence behavior managed by a 2×2 matrix, after t steps, the initial value $[v_0, x_0]$ transforms into $\mathbf{R}(\beta, \eta, \lambda)^t[v_0, x_0]$. Therefore, the convergence rate is determined by the eigenvalues of $\mathbf{R}$. In short, momentum converges when $0 < \eta\lambda < 2 + 2\beta$. Compared to the $0 < \eta\lambda < 2$ range discussed in the analysis of gradient descent earlier, momentum method offers a larger feasible range for parameters.

References

[Achiam et al., 2023] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

[Ahn et al., 2024] Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., and Yin, W. (2024). Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Bsharat et al., 2023] Bsharat, S. M., Myrzakhan, A., and Shen, Z. (2023). Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4. *arXiv preprint arXiv:2312.16171*.

[Bubeck et al., 2023] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

[Chung et al., 2022] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Gu et al., 2021] Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

[Guo et al., 2023] Guo, Q., Wang, R., Guo, J., Li, B., Song, K., Tan, X., Liu, G., Bian, J., and Yang, Y. (2023). Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

[Hu et al., 2021] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[Jin et al., 2020] Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., and Szolovits, P. (2020). What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

[Jin et al., 2019] Jin, Q., Dhingra, B., Liu, Z., Cohen, W. W., and Lu, X. (2019). Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

[Lewis et al., 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

[Li and Liang, 2021] Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

[Liévin et al., 2023] Liévin, V., Hother, C. E., Motzfeldt, A. G., and Winther, O. (2023). Can large language models reason about medical questions? *Patterns*.

[Liu et al., 2023] Liu, J., Zhou, P., Hua, Y., Chong, D., Tian, Z., Liu, A., Wang, H., You, C., Guo, Z., Zhu, L., et al. (2023). Benchmarking large language models on cmexam–a comprehensive chinese medical exam dataset. *arXiv preprint arXiv:2306.03030*.

[Luo et al., 2022] Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., and Liu, T.-Y. (2022). Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

[Nori et al., 2023a] Nori, H., King, N., McKinney, S. M., Carignan, D., and Horvitz, E. (2023a). Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

[Nori et al., 2023b] Nori, H., Lee, Y. T., Zhang, S., Carignan, D., Edgar, R., Fusi, N., King, N., Larson, J., Li, Y., Liu, W., et al. (2023b). Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[Pryzant et al., 2023] Pryzant, R., Iter, D., Li, J., Lee, Y. T., Zhu, C., and Zeng, M. (2023). Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

[Schick and Schütze, 2020] Schick, T. and Schütze, H. (2020). Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

[Singhal et al., 2023a] Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. (2023a). Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

[Singhal et al., 2023b] Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., et al. (2023b). Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*.

[Tang et al., 2024] Tang, X., Wang, X., Zhao, W. X., Lu, S., Li, Y., and Wen, J.-R. (2024). Unleashing the potential of large language models as prompt optimizers: An analogical analysis with gradient-based model optimizers. *arXiv preprint arXiv:2402.17564*.

[Touvron et al., 2023] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

[van Zandvoort et al., 2023] van Zandvoort, D., Wiersema, L., Huibers, T., van Dulmen, S., and Brinkkemper, S. (2023). Enhancing summarization performance through transformer-based prompt engineering in automated medical reporting. *arXiv preprint arXiv:2311.13274*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Wang et al., 2023] Wang, X., Chen, G. H., Song, D., Zhang, Z., Chen, Z., Xiao, Q., Jiang, F., Li, J., Wan, X., Wang, B., et al. (2023). Cmb: A comprehensive medical benchmark in chinese. *arXiv preprint arXiv:2308.08833*.

[Wei et al., 2022a] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022a). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

[Wei et al., 2022b] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

[Yang et al., 2023] Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. (2023). Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

[Yang and Li, 2023] Yang, H. and Li, K. (2023). Instoptima: Evolutionary multi-objective instruction optimization via large language model-based instruction operators.

[Yao et al., 2024] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. (2024). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

[Yao et al., 2023] Yao, Z., Jaafar, A., Wang, B., Zhu, Y., Yang, Z., and Yu, H. (2023). Do physicians know how to prompt? the need for automatic prompt optimization help in clinical note generation. *arXiv preprint arXiv:2311.09684*.

[Yasunaga et al., 2022a] Yasunaga, M., Bosselut, A., Ren, H., Zhang, X., Manning, C. D., Liang, P. S., and Leskovec, J. (2022a). Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

[Yasunaga et al., 2022b] Yasunaga, M., Leskovec, J., and Liang, P. (2022b). Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

[Ye et al., 2023] Ye, Q., Axmed, M., Pryzant, R., and Khani, F. (2023). Prompt engineering a prompt engineer. *arXiv preprint arXiv:2311.05661*.

[Zhang et al., 2019] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

[Zhou et al., 2022] Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. (2022). Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

[Zhu et al., 2023] Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Gong, N. Z., Zhang, Y., et al. (2023). Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

[Ziegler et al., 2019] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.