Clinic-Prompt: Few-Shot Discrete Clinical Prompt Optimization

Rochana Prih Hastuti[1], **Rian Adam Rajagede**[1, 2], **Mengxin Zheng**[1], **Qian Lou**[1]

[1]University of Central Florida, Orlando, FL, USA
[2]Universitas Islam Indonesia, Sleman, Indonesia
{rochana, rian, mengxin.zheng, qian.lou}@ucf.edu

Abstract

Language models have achieved significant success in demonstrating intelligent capabilities. Incorporating these models into clinical healthcare can greatly benefit society. However, they face challenges in clinical tasks due to the need for domain-specific knowledge and expertise and the limited availability of relevant data samples for fine-tuning. Prompt tuning and optimization with fixed language model weights have emerged as highly effective strategies to address this. These approaches adapt pre-trained language models for diverse downstream tasks, particularly in data-scarce (few-shot) settings. In clinical healthcare, natural-language-level discrete prompt optimization is preferred for its superior interpretability and reliability compared to continuous, differentiable prompt vectors. However, the few-shot discrete clinical prompt optimization is unexplored. To tackle this challenge, in this paper, we introduce a novel scheme, *Clinic-Prompt*, that models the non-differentiable discrete prompt optimization as a reinforcement learning problem and incorporates clinical knowledge into the optimization to enhance the performance in two clinical applications: multi-label International Classification of Diseases (ICD) code classification and mortality prediction. Furthermore, we demonstrate the applicability of Clinic-Prompt in a large language model (GPT-4o-mini) setting for the Medication Status Extraction task. Experimental results demonstrate the effectiveness of Clinic-Prompt, improving the performance and applicability of pre-trained models for clinical tasks, with a 2.17% increase in F1-micro and 2.32% increase in accuracy, respectively.

Introduction

Language models have demonstrated remarkable intelligence capabilities to process, understand, and respond to complex queries, which makes them promising candidates for incorporation into clinical healthcare applications. Also, the rapid growth of electronic health records has led to an increased demand for automated clinical decision support systems. However, many clinical tasks are challenging due to the large number of classes, complex medical terminology, and the need for domain expertise. For example, the multi-label classification task of assigning International Classification of Diseases (ICD) codes from the MIMIC-III dataset (Johnson et al. 2016) involves 8,692 unique ICD-9 codes. The subsequent MIMIC-IV dataset (Johnson et al. 2023) increases this number to 14,092 unique ICD-9 and ICD-10 codes and updates the class format. The rapid changes in the dataset format add to the complexity and challenge of adapting new updates. Currently, most proposed methods rely on fine-tuning models to enhance Language Model (LM) performance in this domain, often requiring external domain knowledge and expertise for each specific task as in KEPT (Yang et al. 2022) or MSMN (Yuan, Tan, and Huang 2022).

However, in the medical domain, large amounts of data to fine-tune the model are not always available. In this situation, few-shot prompt training is a promising approach. This approach can solve a wide range of problems using large pre-trained language models (LMs), including left-to-right models such as GPTs (Radford et al. 2019; Brown et al. 2020) and masked LMs such as BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019), etc. Compared to conventional fine-tuning that expensively updates the massive LM parameters for each downstream task, prompting concatenates the inputs with an additional piece of text that steers the LM to produce the desired outputs, where it can be made automatically. For example, using AutoPrompt (Shin et al. 2020) or RLPrompt (Deng et al. 2022).

Clinical knowledge-enriched language models, for example, Clinical-Longformer (Li et al. 2023) or large model like BioMedLM (Bolton et al. 2024), have been used for fine-tuning on downstream clinical tasks. However, the exploration of leveraging these models for automatic prompt approaches remains relatively underexplored. Previous works focus on prompt engineering by exploiting manual templates as well as external medical knowledge as in (Taylor et al. 2023; Lu, Zhao, and Wang 2023) make it less applicable for different tasks. This gap presents an opportunity to investigate how prompt optimization can improve the performance and applicability of pre-trained models in clinical settings.

Additionally, in clinical healthcare, natural-language-level discrete prompt optimization is preferred for its superior interpretability and reliability compared to continuous, differentiable prompt vectors (Deng et al. 2022; Xue et al. 2024). However, the non-differentiable discrete prompt optimization can not use the gradient-based optimization methods. In this paper, we model the discrete clinical prompt optimization into a reinforcement learning (RL) problem and redesign the policy model, rewards, and incorporates clinical knowledge. First, a Policy LM to generate the discrete prompt automatically. Secondly, a single-layer MLP is inserted in the middle of the frozen Policy LM. This layer is trained using Reinforcement Learning (RL) by adjusting the reward function of the downstream task. Lastly, a Task LM with a simple verbalizer setting is modified to ensure the framework's applicability across different downstream tasks. Our proposed framework, Clinic-Prompt, improves RL-based optimization results by leveraging the clinical model in the Policy LM to enable prompt searching in clinical space and further feeding input-aware Policy LM. In the few-shot setting, Clinic-Prompt also consistently improves upon the baseline and achieves results similar to the full-set setting.

Related Work

The existing prompt-based approaches for clinical tasks are not yet prominent, often relying on static templates to improve fine-tuning strategies (Yang et al. 2022; Yuan, Tan, and Huang 2022; Wang, Xiao, and Sun 2023; Shoham and Rappoport 2023). On the other hand, few-shot prompt learning studies adapting GPT-3 to clinical tasks have observed a decrease in performance compared to general domain tasks (Moradi et al. 2021; Gutiérrez et al. 2022). This suggests that even large PLMs may not yield optimal results in specialized domains, highlighting the need for domain-specific PLMs. Other studies used frozen PLMs for zero-shot classification (Sivarajkumar and Wang 2022) and generative tasks (Boyle et al. 2023), the other (Taylor et al. 2023) combined various handcrafted discrete prompt templates with soft prompt learning strategies, finding that prompt learning outperformed traditional classification head training on frozen PLMs. They also found that smaller, domain-specific PLMs can be more effective than large, general PLMs.

The prompt learning strategy itself often resorts to tuning soft prompts (e.g., embeddings) (Li and Liang 2021; Qian et al. 2022; An et al. 2022) which fall short of interpretability and applicability when gradients of PLMs are not accessible and often expensive to compute. It is thus often desirable to use discrete prompts which consist of concrete tokens from a vocabulary. In the other hand, previous work have found that the choice of prompt format, training examples, and prompt order can cause the performance to vary quite significantly (Zhao et al. 2021). Thus, effort have been made to explore discrete prompt formation strategy by selecting from multiple paraphrased prompts (Prasad et al. 2023; Hao et al. 2022), using gradient information to edit the prompt tokens (Shin et al. 2020), and modeling searching of vocabulary (Deng et al. 2022) and editing space (Zhang et al. 2023) with RL-based framework.

Our work builds upon these finding by introducing discrete prompt optimization for clinical tasks. Furthermore, we also explore how exploiting the vocabulary space of clinical PLM affecting the prompt generation process. In addition, we investigate the impact of few-shot settings on prompt learning.

Methods	Few	Auto	Opt	LLM
1) Healthprompt	✗	✗	✗	✗
2) Clinical Manual & CoT	✓	✗	✗	✓
3) RLPrompt	✓	✓	✓	✗
Clinic-Prompt **(Ours)**	✓	✓	✓	✓

Table 1: Comparisons between Clinic-Prompt and related works of prompt learning in clinical and non-clinical setting. *Few*: few-shot, *Auto*: automatic, *Opt*: optimize. Works of 1) Sivarajkumar and Wang (2022), 2) Taylor et al. (2023); Sivarajkumar et al. (2024), and 3) Deng et al. (2022)

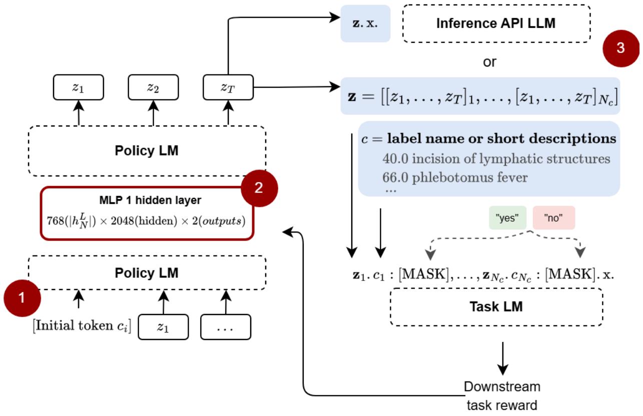
$\mathbf{z}.\,\mathrm{x}.$
Inference API LLM
3
or
z_1
z_2
z_T
$\mathbf{z} = [[z_1, \ldots, z_T]_1, \ldots, [z_1, \ldots, z_T]_{N_c}]$
Policy LM
$c =$ **label name or short descriptions**
40.0 incision of lymphatic structures
66.0 phlebotomus fever
...
2
MLP 1 hidden layer
$768(|h_N^L|) \times 2048(\mathrm{hidden}) \times 2(outputs)$
"yes"
"no"
Policy LM
$\mathbf{z}_1.\,c_1$: [MASK], ..., $\mathbf{z}_{N_c}.\,c_{N_c}$: [MASK]. x.
1
[Initial token c_i]
z_1
...
Task LM
Downstream task reward

Figure 1: Overview of Clinic-Prompt: RL-based discrete prompt optimization using input-aware knowledge for clinical tasks. Step 1: input-aware prompt generation via Policy LM. Step 2: Reinforcement Learning (RL) optimization using a single-layer MLP. Step 3 : Task LM: masked LM modeling for classification tasks. All of LM are kept frozen, with only the task-specific MLP (in red) being trained.

Methods

In this section, we will describe the prompt and reward design in the clinical tasks, specifically in three tasks: Mortality prediction, ICD Code Classification, and Medication Status Extraction.

Adapting RL-based Discrete Prompt Optimization for Clinical Tasks

Combination of discrete text prompt $\mathbf{z}$ with input $\mathbf{x}$ is feasible to execute various NLP tasks directly by using a pre-trained LM's generative distribution $P_{\text{LM}}(\mathbf{y}|\mathbf{z}, \mathbf{x})$, without needing to fine-tune the model (Brown et al. 2020; Gao, Fisch, and Chen 2021). The LM here is a masked language model (MLM) such as BERT (Devlin et al. 2018) or a similar model for longer input like Longformer (Beltagy, Peters, and Cohan 2020) where it is the case for many clinical documents, and $\mathbf{y}$ is the class-label token or verbalizer in the mask position. See Figure 1 (right) for illustrative examples. We use $\mathbf{y}_{\text{LM}}(\mathbf{z}, \mathbf{x})$ to denote the LM output on $\mathbf{x}$ prompted by $\mathbf{z}$.

Our goal is to find the optimal discrete prompt $\mathbf{z}^*$ from vocabulary $\mathcal{V}$ to maximize some downstream performance measure R of $\mathbf{y}_{\text{LM}}(\mathbf{z}^*, \mathbf{x})$. The metric $R(\mathbf{y})$ can be as simple as a match with gold label $\mathbf{y}^*$. Assuming the prompts have fixed length T, we write the task of discrete prompt optimization in the general format below:

$$\max_{\mathbf{z}\in\mathcal{V}^T} R(\mathbf{y}_{\text{LM}}(\mathbf{z}, \mathbf{x})) \tag{1}$$

Automatically searching for discrete prompt $\mathbf{z}$ can be formulated as a Reinforcement Learning (RL) problem as similarly has been done by Deng et al. (2022) and Zhang et al. (2023). An agent selects prompt tokens $[z_1, ..., z_T]$ one by one to maximize the reward $R(\mathbf{y}_{\text{LM}}(\mathbf{z}, \mathbf{x}))$. At time step t, the agent receives previous prompt tokens $\mathbf{z}_{<t}$ and generates the next prompt token z_t according to a policy $\pi(z_t|\mathbf{z}_{<t})$. After the agent finishes the entire prompt $\hat{\mathbf{z}}$, it receives the task reward $R(\mathbf{y}_{\text{LM}}(\hat{\mathbf{z}}, \mathbf{x}))$. Parameterizing the policy network with θ, we can rewrite the problem as

$$\max_{\theta} R(\mathbf{y}_{\text{LM}}(\hat{\mathbf{z}}, \mathbf{x})), \hat{\mathbf{z}} \sim \prod_{t=1}^{T} \pi_\theta(z_t|\mathbf{z}_{<t}). \tag{2}$$

The RL formulation above has the key advantage of not needing gradient access to the LM, treating it instead as a black-box function. This enables us to optimize prompts for LMs whose gradients are too expensive to compute, or LMs that are solely available as inference APIs (e.g., GPT-4).

The policy network π_θ is parameterized to adapt a frozen pre-trained LM (i.e., policy LM) with a simple MLP layer that contains all the parameters θ to be trained. During training, we compute the MLP gradients by back-propagating through the policy LM. After training, we discard the MLP and simply use the learned discrete text prompt for inference. Figure 1 (left) illustrates the policy LM architecture. Specifically, we use the LM to extract contextual embeddings of partial prompt $\hat{\mathbf{z}}_{<\mathbf{t}}$, apply the added task-specific MLP layer to compute the adapted embeddings, and pass the output into the model's original LM head to obtain the next prompt token probabilities.

Task LM Design for In-hospital Mortality Prediction In-hospital mortality is a task that describes whether a patient died during the current admission and is a binary classification task (Van Aken et al. 2021). We adopt the typical prompting setting (Brown et al. 2020; Schick and Schütze 2021b), which solves classification by token infilling for an MLM like BERT by selecting tokens that correspond to a set of predetermined class labels, a.k.a., verbalizers.

However, in the clinical-domain, selecting vocabulary for verbalizers presents a challenge due to the potential specificity of certain terms in the LM vocabulary. To address this, we present the mortality binary classification task to assign a binary label $y_i \in \{0, 1\}$ which is represented by vocabulary token `yes` (or `no`) in the incorporated `[MASK]` token corresponds with the label `deceased`. This approach determines whether an instance is positive (or negative) for the given class, which gives an input format `z.deceased.[MASK].x`. Utilizing this format facilitates its application to other tasks with unique label names, enhancing adaptability and consistency across different classification tasks. In the next section, we use a similar format for the multi-label classification task.

Task LM Design for ICD Coding Multi-Label Classification Automatic International Classification of Diseases (ICD) coding aims to assign multiple ICD codes to a medical note input with an average length of 3,000+ tokens, making it a multi-label classification task. Similar to the previous task, we assign a binary label $y_i \in \{0, 1\}$ for each ICD code in the label space $\mathcal{Y}$ in the incorporated `[MASK]` token, where 1 means that input is positive for an ICD disease or procedure and $i \in [1, N_c]$. We reformulate multi-label classification task input format as $\mathbf{z}_1.c_1$: `[MASK]`, $\ldots, \mathbf{z}_{N_c}.c_{N_c}$: `[MASK]`$.\mathbf{x}$. following Yang et al. (2022). For each candidate code, c_i is a short code description phrase in a free text. For instance, code 40.0 has description *incision of lymphatic structures*. Code descriptions c is the set of all N_c numbers of c_i.

Task LM Design for Medication Status Extraction The medication status extraction task extracts a list of medications from medical notes and labels each with a status modifier: *active, discontinued,* or *neither*. We will use this task for LLM evaluation using OpenAI API. This task can be split into two subtasks: extraction and multiclass classification (Agrawal et al. 2022). We only evaluate the classification task and use Structured Output features from OpenAI API for the extraction part. We use the template of the prior work (Sivarajkumar et al. 2024) for the task input format.

Reward Function Design Both in-hospital mortality prediction and ICD coding tasks are formulated in a similar fashion, which is a binary classification for each possible class, making it easy to design a general reward function. The classification task aims to correctly assign input text $\mathbf{x}$ to its ground truth label y from a set of classes $\mathcal{Y}$. Given prompt $\mathbf{z}$ and training example $(\mathbf{x}, y)$, we compute the reward similarly to hinge loss as the gap between the label probability and the highest probability from other classes. Using the short hand $P_{\mathbf{z}}(y) := P_{\mathbf{LM}}(y|\mathbf{z}, \mathbf{x})$ to denote the probability of label y, we can write the gap as $\text{Gap}_{\mathbf{z}}(y) := P_{\mathbf{z}}(y) - max_{y' \neq y} P_{\mathbf{z}}(y')$. The gap value is positive when the prediction is correct and negative otherwise. We denote $\text{Correct} := \mathbf{1}[\text{Gap}_{\mathbf{z}}(y) > 0]$. For a correct prediction, we multiply the positive reward by a large number to signal its desirability. The resulting reward function in general for binary classification is defined as:

$$R(\mathbf{x}, y) = \lambda_1^{1-\text{Correct}} \lambda_2^{\text{Correct}} \text{Gap}_{\mathbf{z}}(y) \tag{3}$$

Adjusting Equation 3 to ICD coding multi-label classification task is straightforward as follows:

$$R(\mathbf{x}, y) = \frac{1}{|\mathcal{Y}|} \Sigma^{i\in|\mathcal{Y}|} \lambda_1^{1-\text{Correct}} \lambda_2^{\text{Correct}} \text{Gap}_{\mathbf{z}}(y_i) \tag{4}$$

For the medication status extraction task, we set up the predicted class to have a probability of 1 and all other classes to have a probability of 0, similar to the work of Xue et al. (2024). For instance, when GPT-4 extracts and classifies the status medication *insulin* as *active*, we assign a probability of 1 to the active class and 0 to the other two classes. The designated probabilities are then used to compute the Gap, where subsequent steps remain the same as in Equation 4.

Replacing General Policy LM with Domain-Specific LM

The RL approach explores the prompt in the vocabulary $\mathcal{V}$ space guided by the reward signals. This is important as prompt generation for clinical application needs the flexibility of the policy network to adjust it to more suitable LM's vocabulary. Considering previous works (Schick and Schütze 2021a; Gao, Fisch, and Chen 2021), simple words used to define template such as *It was* ______.[Input] or [Input]. *In summary* ______., where we can observe came from general, not specific-domain vocabulary. However, for clinical application (Sivarajkumar and Wang 2022; Taylor et al. 2023) showed that hand-crafted templates with clinical-domain influenced prompt may be more suitable, such as [Input]. *Disease:* ______. and [Input]. ______ *disorder*.

This motivates us to explore the prompt space by changing the policy network between general LM and clinical LM, and examine its impact on the classification performance. After training the policy, we select tokens greedily during inference to produce a deterministic prompt. Later, from the experiment, we observed that changing the general LM used in the policy network to clinical LM does not suffice to get a clinical-domain influenced prompt.

We summarised the average sequence of length as in Gee et al. (2022) on prompts generated from clinical LM, using both general and clinical domain-specific tokenizers. Let $\mathcal{T}$ be a tokenizer associated with vocabulary $\mathcal{V}$, given string w, the mapping function is $\mathcal{T} : w \rightarrow (s_1, \ldots, s_n), s_i \in \mathcal{V}$. Different tokenizers will have different mapping functions which result in different sequences of token, e.g., clinical domain-specific terms represented as a single token in BioMedLM (Bolton et al. 2024) and are not broken down into multiple tokens compared as in GPT2. However, the observation shows that there is no difference in the average length of the sequence, which indicates knowledge from the clinical LM vocabulary was not transferred. Later, we propose to improve this problem in the next section.

Input-aware RL-based Discrete Prompt Optimization

Querying LM with category names can generate additional cues and knowledge of disease as used in Liu et al. (2023). Inspired by that, we inserted an input-aware template as an initial token replacing the default $< |\texttt{endoftext}| >$ token to trigger the policy network in generating clinical prompts. Our template is defined as characterize c_i $\mathbf{z_i}$ where $i \in [1, N_c]$ and c is a knowledge variable that could be label name in general or any knowledge related to each class. The policy network generates input-aware prompt repetitively $[\mathbf{z_1}, \ldots, \mathbf{z}_{N_c}]$ so that each class has a different prompt. We call our proposed scheme Clinic-Prompt scheme which is illustrated in Figure 1.

Experiments

The proposed framework, Clinic-Prompt, was applied to both PLM and LLM to ensure it works in both settings. It was tested on three clinical tasks: mortality prediction (binary classification), ICD Coding (multilabel classification) and medication status extraction. The first two tasks utilized the MIMIC dataset and publicly available PLM. The third task used the CASI-based dataset and LLM to ensure compliance with the MIMIC dataset privacy rules regarding public API.

Dataset

We use MIMIC-III-50 dataset for the multi-label classification task. MIMIC-III-50 is a subset of the original MIMIC-III data that contains the 50 most frequent codes. Data is generated and split following the prior works (Mullenbach et al. 2018; Yang et al. 2022). For task In-hospital mortality prediction, we generate the Mortality Prediction (MORT) subset from MIMIC-III using steps used in Van Aken et al. (2021). For the Medication Status Extraction task, we took the subset from (Agrawal et al. 2022), which is built on top of the CASI dataset (Moon et al. 2014).

Few-shot settings include the creation of subset MIMIC-50-few and MORT-few, generated by down-sampling from the original MIMIC-III-50 and MORT datasets, respectively. In the K-shot setting, we do uniform random sampling from the original set so that each class has at least K samples, preserving the class ratio. It is worth noting that our MIMIC-50-few is different from the MIMIC-III-few used in Yang et al. (2023) as they sample from the bigger MIMIC-III dataset, therefore producing larger ICD codes. We chose 5-shot and 16-shot to align with the prior works (Yang et al. 2022; Taylor et al. 2023).

Models

For ICD Code and Mortality Prediction tasks, we use KEPT-Longformer[1] (Yang et al. 2022) models as the Task LM, then apply templates shown in the previous section during inference. We chose longformer-based LMs because the MIMIC-III dataset has long input text with an average of 3000+ tokens, beyond some common LMs capability (Devlin et al. 2018; Liu et al. 2019) and KEPTLongformer is one of the state-of-the-art Longformer-based for ICD Code classification tasks. For Medication Status Extraction, we use OpenAI GPT-4o mini model[2], specifically "gpt-4o-mini-2024-07-18" accessed via API. All evaluations are performed with Policy and Task LMs maintained in a frozen state, solely used for inference. This entails inputting tokens and retrieving logits without accessing their internal architecture or parameters.

Pre-Processing and Knowledge Variable

For the ICD task we remove characters of the input data as in Yang et al. (2022)[3]. We also chose processed code descrip-

[1] https://huggingface.co/whaleloops/keptlongformer

[2] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[3] https://github.com/whaleloops/KEPT

Scheme	Prompt
Manual Prompt	”is”, ”it is”, ”is a”
Healthprompt (Taylor et al. 2023; Sivarajkumar and Wang 2022)	”disease”, ”patient is on the path to”
Prefix-A (Agrawal et al. 2022)	”Which medications are mentioned and whether they are active, discontinued, or neither”
CoT-S (0-shot) (Sivarajkumar et al. 2024)	”Label any medications in the clinical note as active, discontinued, or neither. Think step by step.”
CoT-S (1-shot) (Sivarajkumar et al. 2024)	”EXAMPLE: Label any medications in the clinical note as active, discontinued, or neither. Text: 2̈. Prior major leg infection. 3. Penicillin allergy. PLANS: Add Fortaz to vancomycin and DC clindamycin. Will also give one dose of tobramycin. If we do another I&D or dressing change, it would be of value to do swab culture of the wound to see what is colonizing it superficially.” \n ANSWER: 1. Penicillin: Neither (doesn't specifically mention active or discontinued, if it's an allergy it's labeled as neither), 2. Fortaz: Active (states it was added to vancomycin, adding indicates it's an active medication), 3. Vancomycin: Active (text states Fortaz was added to Vancomycin, adding indicates it's an active medication), 4. Clindamycin: Discontinued (DC stands for discontinued). \n QUESTION: Using the stored example, label any medications in the text as active, discontinued, or neither. Text:”
Clinic Prompt	”{GENERATED_PROMPT} In the clinical note, extract the medication and its current status as active, discontinued, or neither. Clinical note:”

Table 2: Handcrafted prompts used in all experiments

tions available from the KEPT repository as the knowledge variable. These code descriptions are shorter than the ICD 9 descriptions in Mullenbach et al. (2018)[4]. For the Mortality prediction, we apply all pre-processing steps in Van Aken et al. (2021) to the input and use `deceased` as the knowledge variable. For medication status extraction we did not do any pre-processing and use `medication` as the knowledge variable.

Experiment Setup

Baseline for PLM setting The first evaluation goal is to observe how Clinical LM and input-aware strategy affect classification performance. As baseline, we show the performance of non-prompt masked LM, whose design is mentioned in section and , excluding the optimized prompt $\mathbf{z}$ part. We compare three schemes: the manually crafted prompt, the RL-base, and Clinic-Prompt. We use both general and clinical manually crafted prompts from (Taylor et al. 2023), detailed template in Table 2. For RL-base and Clinic-Prompt we compare the use of DistilGPT2 and BioMedLM for Policy LM. DistilGPT2 (Sanh 2019) is chosen as a general LM that is also used in the prior approach, RLPrompt (Deng et al. 2022), while BioMedLM (Bolton et al. 2024) is chosen as one of the recent generative LM that trained on biomedical data.

In the second evaluation, we explore the robustness of RL-based discrete optimization schemes in the few-shot setting by training the RL model using MIMIC-50-few and MORT-few datasets. In this evaluation, we evaluate schemes using clinical LM in both their Policy LM (BioMedLM) and Task LM (KEPTLongformer), where it shows in the previous evaluation achieved the best performance.

We evaluate both evaluation schemes by calculating the F1-micro result on each task. We use F1-micro as it is commonly used in multi-label classification tasks and to align with prior works (Mullenbach et al. 2018; Yang et al. 2022). Similar to Yang et al. (2022), we use a threshold from the validation set when calculating the F1-score.

Baseline for LLM setting In our third evaluation, we aim to demonstrate that our scheme can be effectively applied to more recent large language model (LLM). Specifically, we use the Medication Status Extraction task and the GPT-4o-mini OpenAI API to replace the Task LM part of the Clinic-Prompt framework. We compare our approach with the prefix and Chain-of-Thought schemes presented by Sivarajkumar et al. (2024), and additionally, we incorporate a prefix method from Agrawal et al. (2022).The evaluation is conducted in both zero-shot and one-shot settings similar to (Sivarajkumar et al. 2024) with Conditional Accuracy metric as in Agrawal et al. (2022). Each evaluation is run twice (baselines) and three times (Clinic-Prompt) to account for the non-deterministic nature of the GPT-4o-mini API.

Results

The first evaluation result can be seen in Table 3. We can see that RL-based discrete prompt optimization approaches, in general, can be used to improve LM performance in the clinical domain. Our proposed scheme, Clinic-Prompt, achieved the highest F1-micro. This improvement occurs in both ICD Code classification and Mortality prediction tasks.

For the second evaluation, the results can be seen in Table 4. Consistent with the results in Table 3, our proposed scheme improves the original KEPTLongformer model even using a small amount of data. The Clinic-Prompt scheme trained in 5-shot or 16-shot settings maintains a higher F1 score than the baseline in both tasks.

The third evaluation is shown in Table 5. The Clinic-Prompt scheme demonstrates significant improvements in Medication Status Extraction tasks under few-shot settings.

Adapting RL-based Discrete Prompt Optimization in Clinical Tasks

Our first attempt to adapt RL-based discrete prompt optimization is by changing the input representation and the

[4] https://github.com/jamesmullenbach/caml-mimic

Scheme	**Policy LM**	**Task**	
		ICD Code	**Mortality**
Baseline	-	22.52	18.79
Manual	-	21.99	16.09
Healthprompt	-	23.19	18.60
RL-based	DistilGPT2	21.86 (0.69)	18.88 (0.23)
w/ *Clinical LM*	BioMedLM	22.04 (0.30)	19.00 (0.12)
w/ *Input-Aware*	DistilGPT2	22.10 (0.69)	19.07 (0.23)
Clinic-Prompt	BiomedLM	**24.27** (0.57)	**19.22** (0.42)

Table 3: Results of non-prompt masked LM (baseline), manual prompt, RL-based prompt optimization, and Clinic-Prompt on full-set data. The values represent the average F1-micro score (%) with the standard deviation shown in parentheses. Bold text represents the best.

Scheme		**Task**	
		ICD Code	**Mortality**
Baseline		22.52	18.79
5-shot	RL-based	22.10 (0.2)	18.89 (0.0)
	Clinic-Prompt	24.31 (0.5)	19.00 (0.1)
16-shot	RL-based	23.18 (1.0)	18.84 (0.1)
	Clinic-Prompt	**24.02** (0.6)	**19.14** (0.2)
Full train set	Clinic-Prompt	24.27 (0.5)	19.22 (0.4)

Table 4: Comparison results of all schemes using clinical Policy LM (BioMedLM) and Clinical Task LM (KEPT-Longformer) in few-shot settings. Values shown are the average of the F1-micro score (%) with the value in the parentheses indicating its standard deviation.

Scheme		**Conditional Accuracy**
0-shot	CoT-S	78.48 (0.17)
	Prefix-A	79.43 (0.25)
	Clinic-Prompt	**80.24** (0.31)
1-shot	CoT-S	76.70 (0.33)
	Prefix-A	80.01 (0.14)
	Clinic-Prompt	**82.33** (0.37)

Table 5: Medication Status Extraction results in few-shot setting. All schemes used GPT-4o-mini as Task LM. Values in the parentheses indicate standard deviation.

reward function without changing the Policy LM (DistilGPT2). We can see in Table 3 in the ICD Code task this approach does not improve the baseline. In the mortality prediction, the improvement is insignificant.

After modifying to use clinical LM as Policy LM we can gain a small improvement compared to general LM. However, this improvement is still small considering we use domain-specific task LM. We believe it occurred because of the limitation of the RL strategy, which is only guided by a single reward value from the downstream task to greedily search prompts in the LM space. With this small amount of information, it seems that BioMedLM can not effectively leverage the exploration of the clinical space. It is also worth mentioning that the RL model must consider 50 codes in the ICD Code classification task, which is more challenging than in the Mortality prediction task. Therefore, we can see in Table 3 that searching in clinical space by using BioMedLM has more effect on the Mortality prediction task, which is only a binary classification task. It also can be seen in Table 3, by applying the input-aware scheme alone to general LM, DistilGPT2, only gives a small improvement. The overall performance improvement is not as high as using clinical LM, because the general Policy LM can not explore clinical space to respond to the input. Finally, by combining the input-aware scheme and clinical LM usage, in the Clinic-Prompt scheme we can gain significant improvement. The input-aware scheme introduces clinical information of the dataset to the clinical Policy LM, leading to an improved prompt generated by the RL model. We can see in all tasks and different Task LM, our proposed scheme Clinic-Prompt gains the highest performance.

Clinic-Prompt in Few-shot Setting

In Table 4 we run the Clinic-Prompt scheme in the few-shot settings. Our proposed scheme consistently improves the baseline. Compared to RL-based without input-aware, Clinic-Prompt achieve more significant improvement. In the Clinic-Prompt scheme, the result when trained with small data is also quite close to when trained with the full data. In the ICD Code task, in the 16-shot setting, by using only 536 of 8,066 samples, Clinic-Prompt gains 1.5% higher than the baseline. These results highlight the Clinic-Prompt robustness in conditions with only a small amount of data.

Clinic-Prompt in LLM

LLM setting result in Table 5 shows that in 0-shot scenario, initially Prefix-A (Agrawal et al. 2022) design was better than Chain of Thought (Sivarajkumar et al. 2024). Clinic-Prompt, which use part of CoT-S (not the best result) as prefix, eventually still achieved the best result with 80.24% accuracy. Similar performance is also shown in the 1-shot setting where Clinic-Prompt with improved Verbalized Reward surpasses all carefully designed baselines for this task, with 82.33% accuracy. These results highlight the effectiveness of Clinic-Prompt, not only in low-data scenarios but also in enhancing other prompt optimization techniques.

Conclusion

This paper introduces *Clinic-Prompt*, a novel scheme that frames non-differentiable discrete prompt optimization as a reinforcement learning problem. Clinic-Prompt integrates clinical knowledge into the optimization process to enhance performance of PLM or LLM in three clinical applications: multi-label International Classification of Diseases (ICD) code classification, mortality prediction, and medication status extraction under few-shot scenarios. Experimental results illustrate the effectiveness of this approach, showing enhancements in both the performance and applicability of pre-trained models for clinical tasks.

References

Agrawal, M.; Hegselmann, S.; Lang, H.; Kim, Y.; and Sontag, D. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1998–2022.

An, S.; Li, Y.; Lin, Z.; Liu, Q.; Chen, B.; Fu, Q.; Chen, W.; Zheng, N.; and Lou, J.-G. 2022. Input-Tuning: Adapting Unfamiliar Inputs to Frozen Pretrained Models. *arXiv e-prints*, arXiv–2203.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150*.

Bolton, E.; Venigalla, A.; Yasunaga, M.; Hall, D.; Xiong, B.; Lee, T.; Daneshjou, R.; Frankle, J.; Liang, P.; Carbin, M.; et al. 2024. Biomedlm: A 2.7 b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*.

Boyle, J. S.; Kascenas, A.; Lok, P.; Liakata, M.; and O'Neil, A. Q. 2023. Automated clinical coding using off-the-shelf large language models. In *Deep Generative Models for Health Workshop NeurIPS 2023*.

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.

Deng, M.; Wang, J.; Hsieh, C.-P.; Wang, Y.; Guo, H.; Shu, T.; Song, M.; Xing, E.; and Hu, Z. 2022. RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3369–3391.

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*.

Gao, T.; Fisch, A.; and Chen, D. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3816–3830.

Gee, L.; Zugarini, A.; Rigutini, L.; and Torroni, P. 2022. Fast Vocabulary Transfer for Language Model Compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 409–416.

Gutiérrez, B. J.; McNeal, N.; Washington, C.; Chen, Y.; Li, L.; Sun, H.; and Su, Y. 2022. Thinking about GPT-3 In-Context Learning for Biomedical IE? Think Again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 4497–4512.

Hao, S.; Tan, B.; Tang, K.; Ni, B.; Zhang, H.; Xing, E. P.; and Hu, Z. 2022. BertNet: Harvesting Knowledge Graphs from Pretrained Language Models. *arXiv preprint arXiv:2206.14268*.

Johnson, A. E.; Bulgarelli, L.; Shen, L.; Gayles, A.; Shammout, A.; Horng, S.; Pollard, T. J.; Hao, S.; Moody, B.; Gow, B.; et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, 10(1): 1.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1): 1–9.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4582–4597.

Li, Y.; Wehbe, R. M.; Ahmad, F. S.; Wang, H.; and Luo, Y. 2023. A comparative study of pretrained language models for long clinical text. *Journal of the American Medical Informatics Association*, 30(2): 340–347.

Liu, J.; Hu, T.; Zhang, Y.; Gai, X.; FENG, Y.; and Liu, Z. 2023. A ChatGPT Aided Explainable Framework for Zero-Shot Medical Image Diagnosis. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*.

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

Lu, Y.; Zhao, X.; and Wang, J. 2023. Medical knowledge-enhanced prompt learning for diagnosis classification from clinical text. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 278–288.

Moon, S.; Pakhomov, S.; Liu, N.; Ryan, J. O.; and Melton, G. B. 2014. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of the American Medical Informatics Association*, 21(2): 299–307.

Moradi, M.; Blagec, K.; Haberl, F.; and Samwald, M. 2021. Gpt-3 models are poor few-shot learners in the biomedical domain. *arXiv preprint arXiv:2109.02555*.

Mullenbach, J.; Wiegreffe, S.; Duke, J.; Sun, J.; and Eisenstein, J. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1101–1111.

Prasad, A.; Hase, P.; Zhou, X.; and Bansal, M. 2023. GrIPS: Gradient-free, Edit-based Instruction Search for Prompting Large Language Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 3845–3864.

Qian, J.; Dong, L.; Shen, Y.; Wei, F.; and Chen, W. 2022. Controllable Natural Language Generation with Contrastive Prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2912–2924.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Sanh, V. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*.

Schick, T.; and Schütze, H. 2021a. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 255–269.

Schick, T.; and Schütze, H. 2021b. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2339–2352.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4222–4235.

Shoham, O. B.; and Rappoport, N. 2023. Cpllm: Clinical prediction with large language models. *arXiv preprint arXiv:2309.11295*.

Sivarajkumar, S.; Kelley, M.; Samolyk-Mazzanti, A.; Visweswaran, S.; Wang, Y.; et al. 2024. An Empirical Evaluation of Prompting Strategies for Large Language Models in Zero-Shot Clinical Natural Language Processing: Algorithm Development and Validation Study. *JMIR Medical Informatics*, 12(1): e55318.

Sivarajkumar, S.; and Wang, Y. 2022. Healthprompt: A zero-shot learning paradigm for clinical natural language processing. In *AMIA Annual Symposium Proceedings*, volume 2022, 972. American Medical Informatics Association.

Taylor, N.; Zhang, Y.; Joyce, D. W.; Gao, Z.; Kormilitzin, A.; and Nevado-Holgado, A. 2023. Clinical prompt learning with frozen language models. *IEEE Transactions on Neural Networks and Learning Systems*.

Van Aken, B.; Papaioannou, J.-M.; Mayrdorfer, M.; Budde, K.; Gers, F.; and Loeser, A. 2021. Clinical Outcome Prediction from Admission Notes using Self-Supervised Knowledge Integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 881–893.

Wang, Z.; Xiao, C.; and Sun, J. 2023. AutoTrial: Prompting Language Models for Clinical Trial Design. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12461–12472.

Xue, J.; Zheng, M.; Hua, T.; Shen, Y.; Liu, Y.; Bölöni, L.; and Lou, Q. 2024. Trojllm: A black-box trojan prompt attack on large language models. *Advances in Neural Information Processing Systems*, 36.

Yang, Z.; Kwon, S.; Yao, Z.; and Yu, H. 2023. Multi-label few-shot ICD coding as autoregressive generation with prompt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5366–5374.

Yang, Z.; Wang, S.; Rawat, B. P. S.; Mitra, A.; and Yu, H. 2022. Knowledge Injected Prompt Based Fine-tuning for Multi-label Few-shot ICD Coding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 1767–1781.

Yuan, Z.; Tan, C.; and Huang, S. 2022. Code Synonyms Do Matter: Multiple Synonyms Matching Network for Automatic ICD Coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 808–814.

Zhang, T.; Wang, X.; Zhou, D.; Schuurmans, D.; and Gonzalez, J. E. 2023. TEMPERA: Test-Time Prompt Editing via Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, 12697–12706. PMLR.