A FRAMEWORK FOR PROMPTOPS IN GENAI APPLICATION DEVELOPMENT LIFECYCLE

Anonymous authors
Paper under double-blind review

ABSTRACT

The use of ”prompts” in the creation process of Generative Artificial Intelligence (GenAI) systems is receiving increasing interest. The significance of these prompts throughout the development cycle, however, is not properly used by current software development lifecycle approaches. This study proposes a unique methodology for integrating timely engineering and management into the creation of GenAI applications.Organizations may benefit from using “PromptOps” to create GenAI applications more quickly, effectively, and securely. It offers a technique to lower the danger of bias, increase the accuracy and dependability of GenAI systems, and decrease the cost of development and implementation.Our platform facilitates the seamless integration of several automated technologies in software development by performing prompt operations (PromptOps). These include Continuous Integration/Continuous Deployment (CI/CD) pipelines, workflows, APIs, and more. Our approach enables developers to easily include automated technologies, leading to a more simplified and efficient process. Furthermore, this study indicates that the framework may enable all stakeholders, including non-engineering units, to convert prompts into services, expanding their use in the building of applications. This study emphasizes the critical significance of prompts in GenAI and shows how their incorporation may improve AI application development, eventually stimulating creativity and driving the adoption of Generative AI technology.

1 INTRODUCTION

The advent of Large Language Models (LLMs) represents a monumental milestone in the field of Natural Language Processing (NLP). Attention-based architectures have revolutionized the field by replacing complex recurrent or convolutional neural networks with a network architecture solely based on attention mechanisms, resulting in models that excel in both quality and efficiency, significantly influencing subsequent LLM advancements (Kaplan et al., 2020; Vaswani et al., 2017). These powerful LLMs, such as GPT-3 (Generative Pre-trained Transformer 3), Palm2 (Pathways Language Model 2) and their successors, have demonstrated remarkable capabilities as versatile computational engines.Their ability to process and generate natural language text has found applications across a wide range of domains, from language translation to content generation. However, harnessing the full potential of LLMs and effectively utilizing them in real-world applications necessitates a nuanced understanding of the critical role that prompts play in steering these models.

Emergence of Large Language Models (LLMs) The emergence of LLMs has reshaped the landscape of AI-driven applications. These models have the capacity to understand, interpret, and generate human language at an unprecedented scale and complexity. They have demonstrated the capability to perform tasks that range from text completion to question-answering and text summarization. This transformative technology has laid the foundation for a new era of AI-driven applications, where human-machine interactions are facilitated through natural language.

The Trend Towards LLM-Based Application Development In recent years, there has been a noticeable trend towards building applications that leverage LLMs as the underlying intelligence. This trend can be attributed to the versatility and adaptability of LLMs, which allow developers to create a wide array of applications without the need for custom-built machine learning models. From simple text-based chatbots to sophisticated content generators, LLMs have become the go-to choice for developers seeking to integrate natural language understanding and generation capabilities into their applications. This shift is evident in domains like customer support, content creation, and data analysis.

The Crucial Role of Prompts in LLM Communication To harness the power of LLMs effectively, communication with these models is achieved primarily through prompts. A prompt is a natural language input or instruction that is provided to the LLM to elicit a desired response or behavior. It acts as the bridge between human intent and machine execution. For example, a prompt might instruct an LLM to translate a sentence from English to French or generate a summary of a news article. The quality, clarity, and specificity of prompts are paramount in obtaining the desired output from the LLM.

The Impact of Prompts and Model Variants on Output However, it is essential to recognize that prompts alone are not the sole determinants of LLM behavior. The choice of prompt and the specific LLM variant being used can significantly influence the output generated. Different prompts may yield varying results, even when targeting the same task. Moreover, LLMs often exist in multiple versions or variants, each with its own characteristics and performance nuances. The interaction between prompts and model versions is complex and requires careful consideration. The significance of prompts and their management has become increasingly evident as more organizations and developers integrate LLMs into their workflows and applications. The quality of prompts directly impacts the utility, reliability, and accuracy of LLM-based systems. Hence, the effective control and optimization of prompts have emerged as critical areas of research and development in the field of NLP.

As LLMs continue to evolve and grow in power, the importance of prompt engineering becomes a prominent concern. There has been some prior research of identifying suitable prompts (Sanh et al., 2021;Mishra et al., 2021) along with the proposal of relevant tools (Bach et al., 2022; Zhou et al., 2023). However, there is a lack of research concerning prompt management and operation. To address this gap, we propose the ”*GenFlow*” framework, aimed at streamlining prompt management and operations. **Our main contributions are:**

- We introduce the concept of incorporating prompt management into the DevOps (Development and Operations) flow. By aligning PromptOps with established software development practices, we facilitate the seamless integration of prompt operations into development pipelines, workflows, and APIs. This integration ensures that prompts are managed consistently, enhancing the reliability and efficiency of GenAI application development.

- Our proposed method, ”GenFlow”, empowers both developers and non-coders. This tool democratizes Prompt usage by providing an accessible interface for creating, modifying, and optimizing prompts. Its user-friendly design allows stakeholders from diverse backgrounds to harness the potential of prompts, thereby broadening their utility in application development.

- Within our framework, we introduce the concept of Prompt as a Service (PaaS). This extends the reach of prompts beyond development teams, enabling various stakeholders to utilize prompts as integral components in application building. This extension aligns with the growing recognition that prompt engineering will be a pivotal focus in the future of GenAI technology.

2 RELATED WORK

Prompt Engineering The utilization of prompts provides an intuitive and natural way for human interaction with generative models, such as text-to-image model and LLMs. (Brown et al., 2020; Schick & Schütze, 2021; Sanh et al., 2022; Rombach et al., 2022). However, the effectiveness of LLMs requires accurate prompt design, either through manual intervention Laria & Kyle (2021) or automated methods (Gao et al., 2021; Shin et al., 2020). It is primarily since LLMs do not interpret prompts in the same manner as humans Albert & Ellie (2021) Yao et al. (2021). While numerous successful prompt tuning methods have leveraged gradient-based optimization over continuous

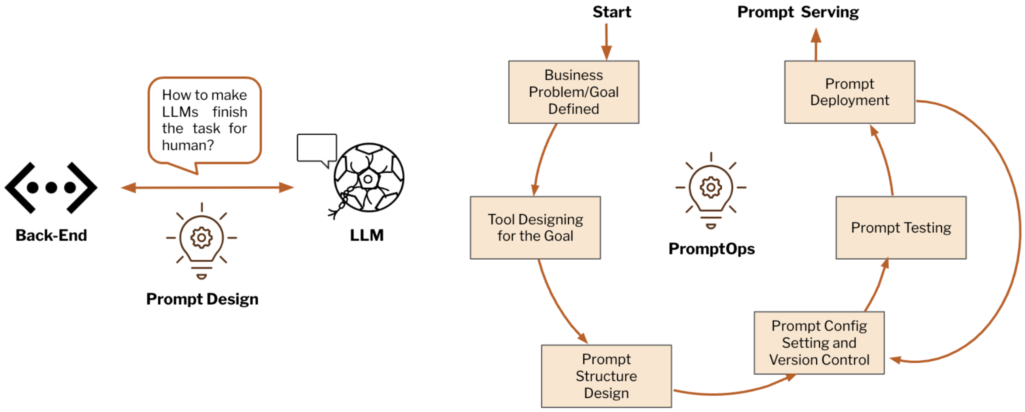

How to make LLMs finish the task for human?
Back-End
LLM
Prompt Design
Start
Prompt Serving
Business Problem/Goal Defined
Prompt Deployment
PromptOps
Tool Designing for the Goal
Prompt Testing
Prompt Config Setting and Version Control
Prompt Structure Design

Figure 1: (a) Prompt is the bridge between application and LLM (b) The diagram which describes each step in the workflow of prompt management

spaces (Liu et al., 2021; Qin & Eisner, 2021; Lester et al., 2021), the computational cost of computing gradients escalates, especially when access to models shifts to APIs that may not offer gradient access. This raises a practical challenge: How can we empower users to create, design, and refine prompts effectively? This process, known as prompt engineering, is crucial for successful deployment due to the significant impact downstream predictions caused by prompt choices, especially in zero-shot settings (Perez et al., 2021; Zhao et al., 2021; Albert & Ellie (2021).

Prompt Management Prompt management is a vital aspect of the emerging field of Natural Language Processing (NLP) where prompts serve as a bridge for interaction with Large Language Models (LLMs). Researchers have developed various techniques to search suitable prompts, such as prompt generation Tianyu et al. (2021) Eyal et al. (2021), prompt scoring Joe et al. (2019), and prompt paraphrasing Jiang et al. (2020) Weizhe et al. (2021). However, the journey doesn't end with well-engineered prompts, there arises a need for prompt management. Prompt management encompasses the organization, storage, and retrieval of these well-designed prompts to facilitate their efficient use in different contexts. The most notable tools in prompt management are OpenPrompt Ning et al. (2022) and PromptSource (Bach et al., 2022).

OpenPrompt simplifies prompt management within the world of pre-trained language models (PLMs). With the rise of prompt learning in natural language processing, there emerged a pressing need for a standardized framework. OpenPrompt fills this void by offering a user-friendly, research-focused toolkit. It's designed to cater to diverse PLMs, task formats, and prompting modules, providing a unified platform for effective prompt management.

Alongside OpenPrompt, PromptSource steps in as an essential tool for prompt development and sharing within the NLP landscape. As a prompt repository, its templating language for crafting data-linked prompts, an intuitive iterative development interface, and a vibrant community that contributes and collaborates, PromptSource empowers users with a wealth of over 2,000 prompts spanning around 170 datasets. This invaluable resource fosters seamless prompt management and utilization in language model training and querying, ultimately enhancing your NLP endeavors.

Although these tools have been introduced, the increasing prevalence of incorporating large language models (LLMs) into diverse applications highlights the imperative requirement for establishing **operational workflows for prompts** within this context. In pursuit of enhanced operational capabilities for prompts, encompassing Continuous Delivery (CD), version control, and more, we introduce a novel framework known as ”*GenFlow*.”

3 SYSTEM DESIGN AND WORKFLOW

The conventional approach to leveraging Large Language Models (LLMs) often involves the use of numerous prompts and processes to construct an application, particularly when each prompt requires fine-tuning. This method, while effective in achieving specific results, can lead to complications in maintenance, deployment, and management.

3.1 GenFlow FRAMEWORK

Through the abstraction of prompts into a configurable format, we simplify the development ”process” significantly. Once prompts are transformed into configurations, they consist of parameters (variables) and prompt text. The prompt's parameters can be set through a user-friendly front-end interface. Prompt configurations become a resource introduced to the backend, allowing the backend or core application to focus on communication with the generative AI model, reducing its coupling with backend or core applications. A simplified representation of a PromptConfig is as follows:

PromptConfig:

```
{
  ”param1”: ”value1”,
  ”param2”: ”value2”,
  ...
}
```

PromptConfig can be engineered by AI consumers from various domains and can be configured accordingly. In other words, the same backend/core application can generate different applications simply by employing different PromptConfigs.

With PromptConfigs in place, we can establish independent version control processes for prompts. This includes version tracking, diff comparisons, and version control. Prompts become independently manageable, effectively transforming prompts into PromptOps that seamlessly integrate into DevOps practices. (Figure 2c)

However, this architecture has its limitations, as it does not facilitate a sequence of operations, even though many applications are composed of a series of prompts. To address this limitation, we elevate the concept of promptConfig to that of a web API. Web APIs are constructed using URLs (Endpoints) and parameters, where each Endpoint can represent a functional module. This implies that every promptConfig can become an individual API. The structure of a PromptConfig as a web API is represented as follows:

PromptConfig:

```
{
  ”param1”: ”value1”,
  ”param2”: ”value2”,
  ...
}
```

PromptAPI (GET): https://domain.name/{promptName}?param1=value1&param2=value2...

PromptAPI (POST):

https://domain.name/{promptName}

POST data:

```
{
  ”param1”: ”value1”,
  ”param2”: ”value2”,
  ...
}
```

When each prompt becomes an API, it becomes straightforward to manage and integrate through workflow management tools. Consequently, diverse prompts can be seamlessly combined to create extensive applications, and even the composition of prompts can be exposed as an API.

promptCombosAPI = (prompt 1 API -¿ prompt 2 API -¿ prompt 3 API)

This approach allows us to regard Prompts as a Service. We term this entire process a ”FLOW,” where each PromptConfig/PromptAPI serves as a ”NODE.” This sequence of operations forms a ”FLOW”, as Figure 2b, where PromptConfigs are used as ”NODES” that can be orchestrated into a coherent process.

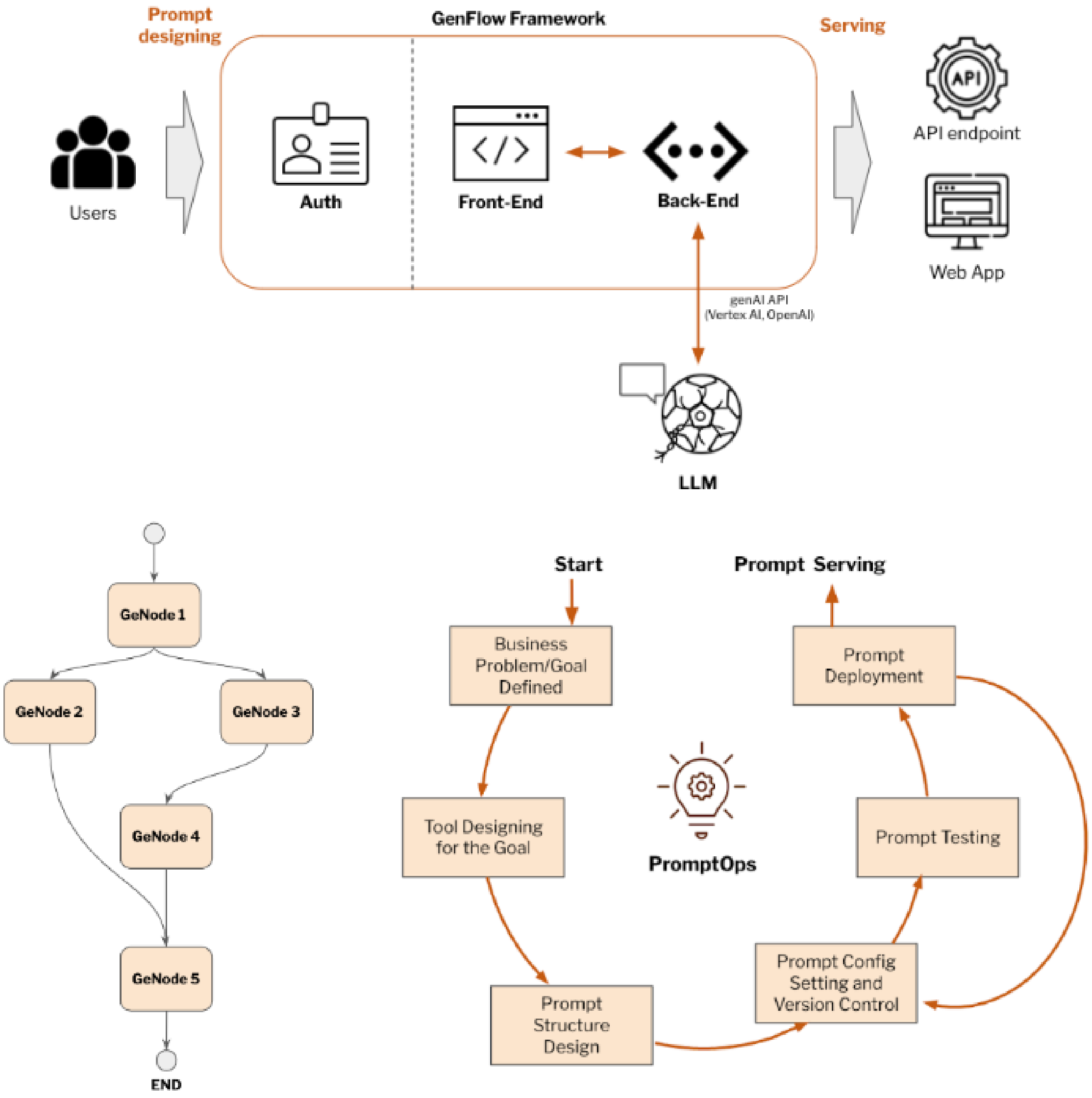


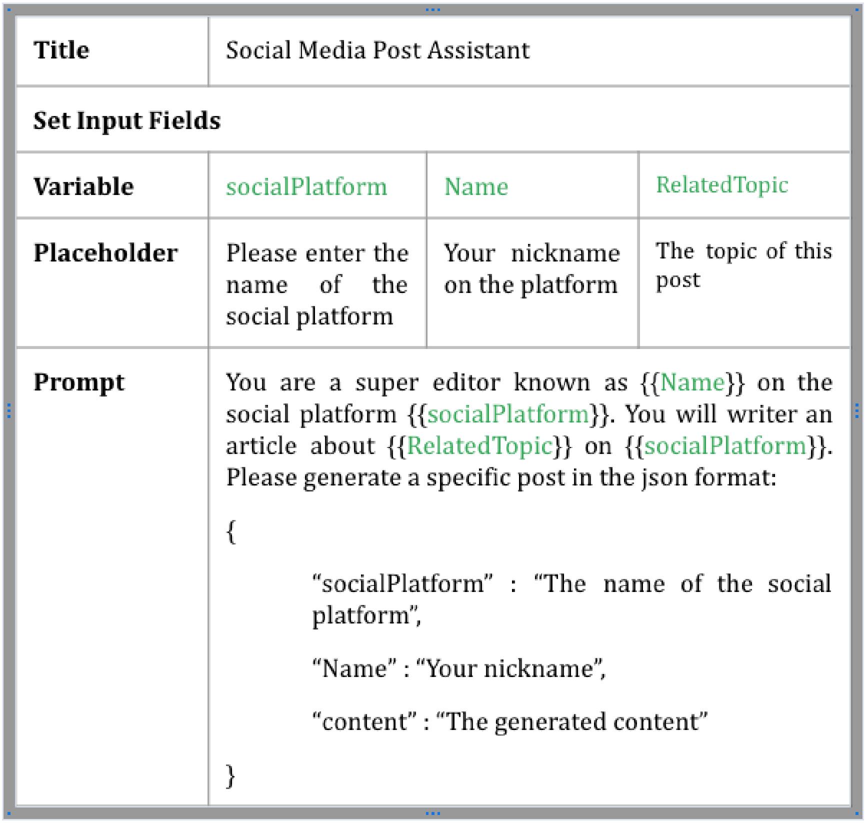
Figure 2: (a) This diagram illustrates how users utilize *GenFlow* framework (upper) (b) The diagram of a “FLOW” (down left) (c) The PromptOps workflow (down right)

3.2 GeNode

Within the *GenFlow* framework, there exists a fundamental component known as ”*GeNode*,” which assumes the role of facilitating prompt editing and serves as a fundamental unit for prompt configuration. Within the user interface of *GeNode*, users are provided with the flexibility to tailor various facets of *GeNode* in accordance with their specific requirements. This entails the ability to designate a user-defined title, specify personalized variables, and construct the prompt content according to their preferences. For instance, users have the autonomy to create a *GeNode* themselves, as exemplified in Figure 3. Once these configurations have been meticulously defined and subsequently saved, the *GeNode* is preserved as a prompt template, effectively culminating in the creation of a *GeNode* aptly named ”Social Media Post Generator.”

Through the innovative power of *GeNode*, users gain the remarkable ability to effortlessly create a multitude of applications, all without the need for advanced coding skills. By simply configuring various parameters within the *GeNod*e interface, users can easily craft a diverse range of utilities. These encompass a broad spectrum of functions, including translation services, computational tools, code generation aids, and much more.

By wholeheartedly embracing *GeNode*’s capabilities, users not only unlock the untapped potential of prompts for comprehending natural language but also effectively harness this resource to build an impressive array of real-world applications. Each of these applications is meticulously tailored to meet their unique requirements. This adaptability and flexibility undeniably position *GeNode* as an indispensable and cardinal component within the continually evolving landscape of prompt-driven tool development.



Title	Social Media Post Assistant		
Set Input Fields			
Variable	socialPlatform	Name	RelatedTopic
Placeholder	Please enter the name of the social platform	Your nickname on the platform	The topic of this post

Prompt	You are a super editor known as {{Name}} on the social platform {{socialPlatform}}. You will writer an article about {{RelatedTopic}} on {{socialPlatform}}. Please generate a specific post in the json format:
	{
	“socialPlatform” : “The name of the social platform”,
	“Name” : “Your nickname”,
	“content” : “The generated content”
	}		

Figure 3: An example of *GeNode* design

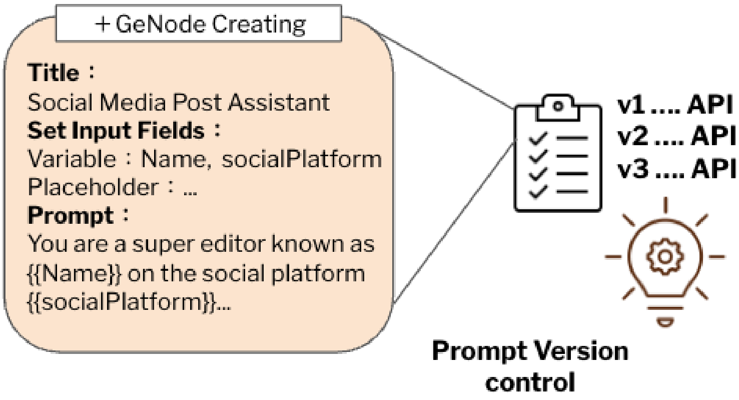
3.3 GeNode VERSION LIST

Within the *GenFlow* framework, we have incorporated the concept of version control. When creating a *GeNode*, users have the ability to establish new versions for the same *GeNode*. These new versions can be directly modified from the original *GeNode* prompt. Consequently, a single application (*GeNode*) can have multiple distinct versions, with the option to select one as the published version. Users can access this published version through a user-friendly URL link that leads to the application's web user interface. Alternatively, they can serve the application directly by copying its API URL link.

Through this meticulously designed system, we have seamlessly integrated prompt configuration, version control, and serving as API endpoints and web UIs within our framework. This integration greatly simplifies the processes of application development and deployment, rendering them more accessible, manageable, and user-friendly.

3.4 GeNode List

Once a *GeNode* has been successfully created, it will be listed in the *GeNode* List page. Each *GeNode* will be accompanied by essential details such as its title, published version, version Create time, and actions. Specifically, under the "Actions" column, users will find convenient links to perform actions such as creating a new version, accessing the UI via URL link, and serving the application using the API URL link. In the *GenFlow* UI section, visual representations will provide further insights and guidance on these features.



Title：
Social Media Post Assistant
Set Input Fields：
Variable：Name, socialPlatform
Placeholder：...
Prompt：
You are a super editor known as {{Name}} on the social platform {{socialPlatform}}...

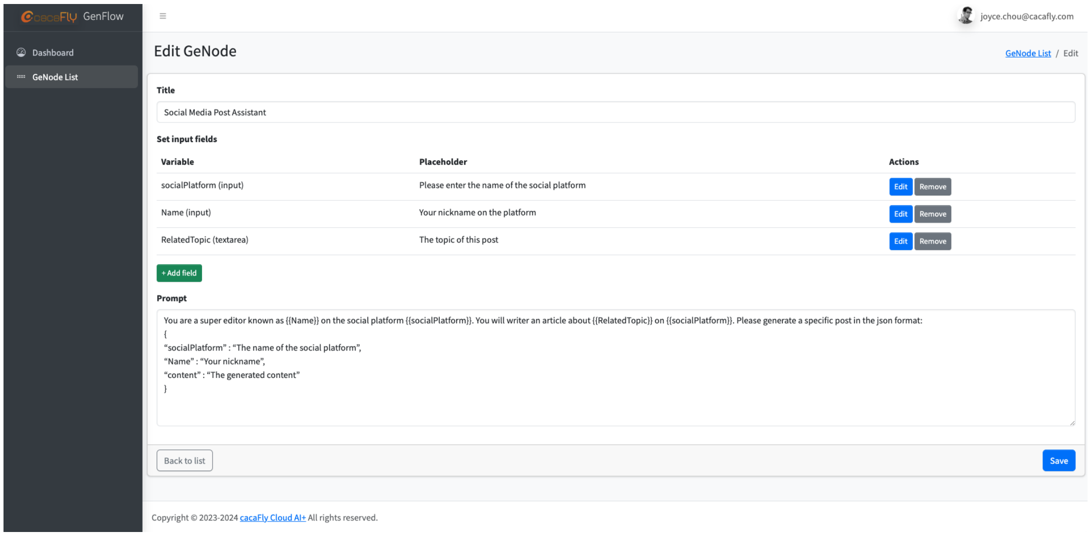
Figure 4: The diagram of *GenFlow* version control

4 IMPLEMENT BASED ON OUR FRAMEWORK

To facilitate user interaction and streamline the process further, we have developed the *GenFlow* web interface, as illustrated in Figure 5. Within this interface, users can seamlessly engage in prompt design and editing on the *GeNode* edit page. This flexibility allows a single *GeNode* to be edited into multiple versions, with users having the prerogative to select which version to publish.

Figure 7 showcases a user-friendly version list UI, which enhances the management of these different *GeNode* versions. This framework empowers users to rapidly create customized applications. Through the utilization of UI links and API links, these applications can be served effortlessly. Notably, this entire workflow is devoid of coding complexities, as users can solely focus on prompt editing. This concept aligns harmoniously with our previously introduced notion of ”Prompt as a service,” revolutionizing the application development paradigm.

By using nodes (*GeNode*) strung together as a flow can be achieved through various methods. Currently, there are similar services available. For instance, GCP offers Workflow (as shown in Figure 6), and AWS provides Step Functions.



Edit GeNode

Title

Social Media Post Assistant

Set input fields

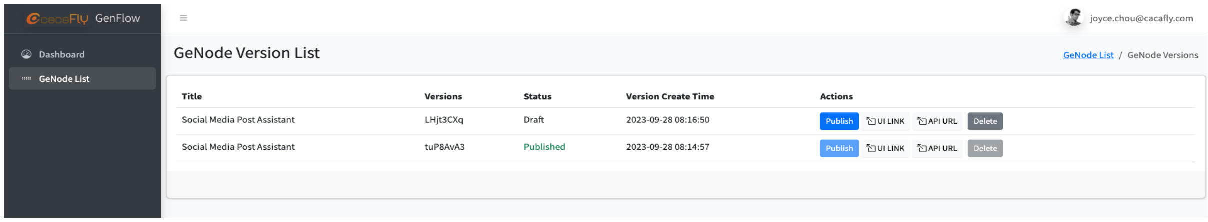
Variable	Placeholder	Actions
socialPlatform (input)	Please enter the name of the social platform	Edit Remove
Name (input)	Your nickname on the platform	Edit Remove
RelatedTopic (textarea)	The topic of this post	Edit Remove

Prompt

```
You are a super editor known as {{Name}} on the social platform {{socialPlatform}}. You will writer an article about {{RelatedTopic}} on {{socialPlatform}}. Please generate a specific post in the json format:
{
“socialPlatform” : “The name of the social platform”,
“Name” : “Your nickname”,
“content” : “The generated content”
}
```

Copyright © 2023-2024 cacaFly Cloud AI+ All rights reserved.

Figure 5: With the GeNode view, users can build their own application by designing the prompt



GeNode Version List

Title	Versions	Status	Version Create Time	Actions
Social Media Post Assistant	LHjt3CXq	Draft	2023-09-28 08:16:50	Publish UI LINK API URL Delete
Social Media Post Assistant	tuP8AvA3	Published	2023-09-28 08:14:57	Publish UI LINK API URL Delete

Figure 6: *GeNode Version List* view

5 CONCLUSION

In this study, we have introduced a comprehensive framework, *GenFlow*, to streamline and enhance the development of applications powered by Large Language Models (LLMs) through prompt management and operation. As the utilization of LLMs becomes increasingly prevalent in diverse applications, the role of prompts in shaping the behavior of these models has become paramount.

Our framework, *GenFlow*, addresses the critical need for prompt management within the context of LLM-driven application development. By seamlessly integrating prompt operations (PromptOps) into established DevOps practices, we have demonstrated the potential to enhance the reliability and efficiency of GenAI application development. This integration extends to Continuous Integration/Continuous Deployment (CI/CD) pipelines, workflows, APIs, and more, ensuring that prompts are consistently managed.

One of the key contributions of our approach is the democratization of prompt usage. *Genflow* empowers not only developers but also non-engineering units to create, modify, and optimize prompts through its accessible interface. This democratization broadens the utility of prompts in application development, facilitating collaboration and creativity across diverse teams.

Furthermore, our framework introduces the concept of Prompt as a Service (PaaS), allowing various stakeholders to leverage prompts as integral components in application building. This extension

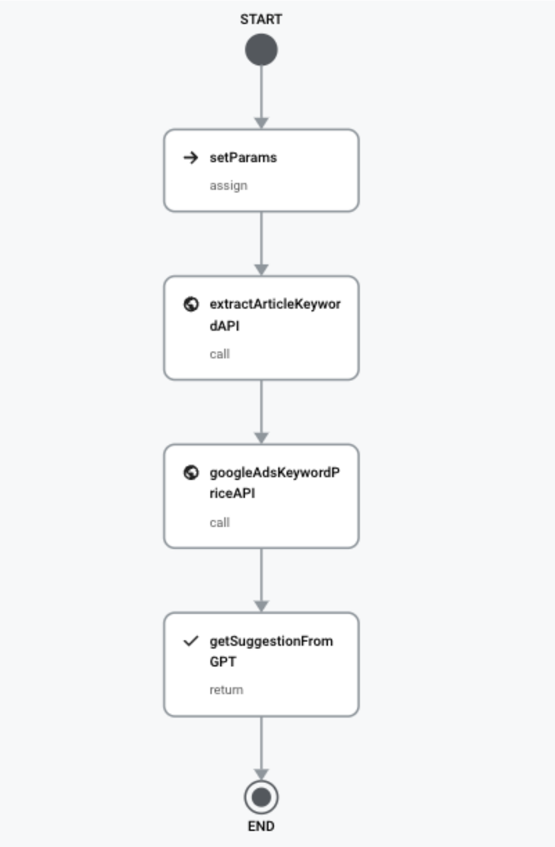


Figure 7: An example of GCP *Workflow* service

aligns with the growing recognition that prompt engineering will play a pivotal role in the future of GenAI technology.

Overall, the *GenFlow* framework offers a holistic solution to the challenges of prompt management and operation in GenAI application development. It empowers users to harness the full potential of LLMs while simplifying the process and promoting collaboration across disciplines. As the field of GenAI continues to evolve, prompt management will remain a critical aspect, and *GenFlow* is poised to play a pivotal role in driving innovation and efficiency in this domain.

In conclusion, the effective control and optimization of prompts through *GenFlow* hold the promise of transforming the GenAI landscape, ultimately fostering creativity, reliability, and widespread adoption of Generative AI technology. We look forward to further research and development in this exciting field, with the aim of advancing the state of the art in GenAI application development.

REFERENCES

Webson Albert and Pavlick Ellie. Do prompt-based models really understand the meaning of their prompts? 2021.

Ben-David Eyal, Oved Nadav, and Reichart Roi. Pada: A prompt-based autoregressive approach for adaptation unseen domains. 2021.

Zheng Bao Jiang, Frank F, Xu, Junaraki, Howcanweknowwhatlanguage, and Transactionsoftheassociationforcomputationallinguistics. 8:423–438, 2020.

Davison Joe, Feldman Joshua, and Rush Alexander M. Common sense knowledge mining from pretrained models. pp. 1173–1178, 2019.

Reynolds Laria and Mcdonell Kyle. Prompt programming for large language models: Beyond the few-shot paradigm. pp. 1–7, 2021.

Ding Ning, Hu Shengding, Zhao Weilin, Chen Yulin, Liu Zhiyuan, Zheng Haitao, and Sun Maosong. Openprompt: An open-source framework for prompt-learning. pp. 105–113, 2022.

Gao Tianyu, Fisch Adam, and Chen Danqi. Making pre-trained language models better few-shot learners. pp. 3816–3830, 2021.

Yuan Weizhe, Neubig Graham, and Liu Pengfei. Bartscore: Evaluating generated text as text generation. 34:27263–27277, 2021.

Lu Yao, Bartolo Max, Moore Alastair, Riedel Sebastian, and Stenetorp Pontus. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. 2021.