International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 08 | Aug 2024 www.irjet.net p-ISSN: 2395-0072

Efficient Prompt Design Automation for Large Language Models by Parts of Speech tagging leveraging Viterbi Algorithm

Sayan Guha[1]

Associate Director & Principal Architect, AI & Analytics Practice, Cognizant Technology Solutions, West Bengal, India

---

Abstract – *Maintaining a standard in writing prompts to interact with Language models requires a disciplined learning and approach on the acceptable precision, conciseness and brevity of the prompts which effectively produces results better in cases where such standards are maintained as compared to the situations where it is not maintained. The prompt engineering practice guidelines states to have only actions which relevant and which can produce precise results as compared to incidents where standards are not maintained. The manual process of prompt design could be accelerated by a system which includes Viterbi algorithm-based Path Pruning which would accept raw English language input and would prune the path which is most relevant and would discard the paths which have lower probabilities. In this process, the Viterbi algorithm would identify the key elements in Parts of Speech tagging and will align and assemble the natural language to fit a prompt template in the order expected by the Large Language model and in the right practice of prompt design.*

Key Words: Prompt Engineering, Viterbi Algorithm, Path Pruning, Prompt design, Few Shot prompting, Natural Language, Parts of Speech (POS) Tagging, RTF (Role Task Format), Generative AI, Large Language Model (LLM), Artificial Neural Network (ANN), Hidden Markov Model (HMM)

1.INTRODUCTION

In today's rapidly evolving technological landscape around prompt engineering practice evolved as a major component for Generative-AI driven solutions the ability to design effective prompts remains a critical challenge, especially when dealing with raw, unstructured English inputs. To address this, I propose leveraging the Viterbi algorithm to identify the optimal prompt while discarding low effective options through path pruning. This approach aims to automate and streamline the prompt design process, making it more accessible and time-efficient for individuals who may lack expertise in standard prompt writing techniques. In this paper, I will explore how this method can facilitate the creation of high-quality prompts, thereby enhancing productivity and ensuring consistency in various applications.

1.1 Motivation

My motivation for writing this paper is to accelerate and simplify the process of prompt engineering, making it accessible to a broader audience, including those who may not be skilled in writing prompts according to standard practices. By leveraging the Viterbi algorithm, I aim to transform raw, unstructured inputs into effective and suitable prompts. This approach not only streamlines the prompt design process but also ensures that high-quality prompts can be generated efficiently. The system I propose will take raw prompts as input, apply the Viterbi algorithm to identify the best possible prompt, and discard less-effective options through path pruning. This method will facilitate an automated, time-efficient solution, empowering individuals to create effective prompts without needing extensive expertise in prompt engineering. Ultimately, this research seeks to scale the practice of prompt engineering, making it more inclusive and efficient.

1.2 Aim of this paper

The aim of this paper is to develop a novel approach to efficient crafting of prompts that accelerates and simplifies the process, making it accessible to a broader audience, including those without specialized skills in writing prompts according to standard practices. By leveraging the Viterbi algorithm, this research seeks to transform raw, unstructured inputs into effective and suitable prompts. The proposed system will take raw prompts as input, apply the Viterbi algorithm to identify the best possible prompt, and discard less-effective options through path pruning. This method aims to streamline the prompt design process, ensuring the efficient generation of high-quality prompts. Ultimately, this research endeavors to scale the practice of prompt engineering, making it more inclusive and efficient, thereby empowering individuals to create effective prompts without needing extensive expertise in the field.

2. LITERATURE REVIEW

A literature review was undertaken encompassing few academic and industry papers. This section reviews Prompt engineering best practices as undertaken by organizations & the key aspect of prompt design can be achieved using Viterbi algorithm.

In [1], the authors highlight the progression from simple probabilistic frameworks to sophisticated neural networks like the Generative Pre-trained Transformer (GPT) series. These advancements have significantly enhanced machine understanding and the generation of human-like language. The authors also observed Prompt engineering customizes language models for specific tasks, improving human-AI interaction and hence prompt engineering as the de-facto approach to interact with language models in the fundamental premise of my research. We also observe that the authors considered the basic understanding that ability to understand contextual Relationships in case of Modern language models, such as GPT, learn contextual relationships between words from extensive text corpora

In [2], the authors have discussed **self-consistency** as a significant concept on effective prompt design. For complex reasoning tasks with multiple valid paths, self-consistency generates diverse reasoning chains by sampling from the language model's decoder. It then identifies the most consistent final answer by marginalizing these sampled chains. The authors also observed the ability to perform logical reasoning is critical for LLMs to solve complex, multi-step problems across diverse domains.

Further to this as studied in [3], the author has elaborately talked about in-depth analysis of various prompt engineering techniques, including the use of parts of speech in Chain-of-Thought prompting. Referring the study done by the author we can build the following table to summarize the Parts of Speech and their need to maintain the consistency and coherence of the Prompts

Table -1: Parts of Speech Construct in Prompt Design

Part of Speech	Construct	Purpose
Nouns and Pronouns	Identify entities	Maintain coherence in reasoning
Verbs	Describing actions	Sequential Reasoning
Adjectives and Adverbs	Provides additional context to entities	Specificity to Prompts
Prepositions & Conjunctions	Connect parts of reasoning	Logical flow and coherence

The above table of Parts of Speech construct and purpose has been derived from the work in [3] and the remarkable deduction on the coherence has been discussed in detail in her work.

In the literature to have even a new perspective in [4] the author illuminates us on the way parts of speech in prompt design can significantly enhance the effectiveness of educational prompts. It emphasizes the importance of understanding grammatical structures to create prompts that are clear, engaging, and tailored to specific learning objectives and hence there are ample studies and literature I have studied and adhered to my own industry experience that prompts structuring, coherence and part of speech construct in a consistent manner lays the foundation for effective way of interaction to exploit the best from LLMs.

In the next set of literatures to follow, I have studied the existing work of the researchers on POS (Parts of Speech tagging) leveraging Viterbi algorithms from the viewpoint of its advantages over other auto tagging algorithms. The studies made by the authors in [5] talks about various approaches on auto tagging. Considering rule-based approach, Artificial Neural Network and Hidden Markov model-based approach leading to Viterbi algorithm approach for the later, the below table provides the findings from all the three different approaches which makes me pursue Viterbi algorithm for POS tagging for my scenario where raw English input is structured with POS tagging and further constructed as per the expectations of prompt design.

Table -2: Comparison of algorithms on POS tagging

Algorithm	Approach	Observations obtained from literature
Rule Based	Linguistic feature classification	Time consuming and expert help required on linguistics classification
Artificial Neural Network (ANN)	An assortment of an enormous number of interconnected handling neurons	Pre-processing activity before working on the actual ANN-based tagger
Hidden Markov Model (HMM) & Viterbi algorithm	Markov model-based tagger framework -Viterbi algorithm is used utilizing the Hidden Markov Model	Comparatively more accurate, addressing stochasticity and less time consuming

In [6], the author has taken a detailed view of Viterbi's performance and considered Viterbi with the combination of partial probabilities (δ), back pointers (ψ), and backtracking, which finds the globally best path, provides the highest accuracy gain compared to that of Unigram model with probabilities of single words contribute the least to accuracy, as they don't account for dependencies between words.

Furthermore, the authors in [7], concluded through their experiment that Viterbi method uses dynamic programming

to compute the likelihood of a word in every conceivable POS and choose the best one as the final POS tagger.

3. BUSINESS SCENARIOS

There could be possibilities of many business scenarios where the solution fits in and adds value. For the sake of the experiment, I have limited my experiment for two standard use cases scenarios, evaluation of which to be discussed in Section-5 to follow.

I have not considered very common scenarios like AI-Driven Customer Support system bot or Product Recommendation engines to keep the originality of this work and for the reason, Viterbi algorithm and various experiments on using the same or by means of Hidden Markov Models have been already accomplished by many of the researchers in the past.

I have chosen the subjects of business scenario as below:

1. Stock Enquiry (Inventory Availability)
2. Warranty Enquiry auto support

3.1 Stock Enquiry / Inventory Availability System

Business Scenario: Ensuring about the availability of a particular product in the stock through chat window in any order

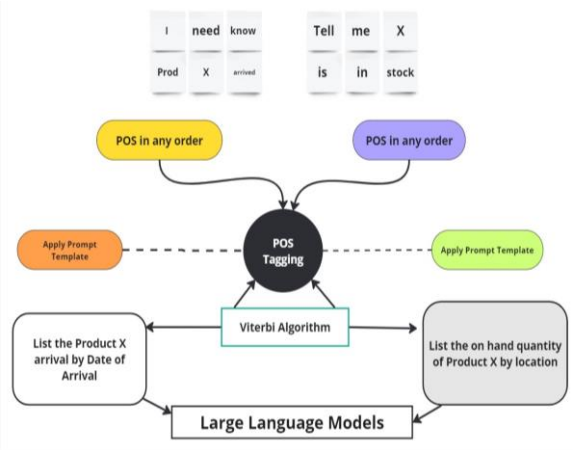

Fig. 1: Sequence flow diagram (Stock Enquiry)

Solution Construct:

- POS tagging of the chat using Viterbi algorithm
- Apply pre-configured prompt template
- Populate the prompt template
- Generate LLM response in user interactive contextual manner
- Deliver the response back to the user

Business Value: Ensuring the accurate information about stock in hand and inventory management leading to better sales over period

3.2 Warranty Enquiry auto support

Business Scenario: Ensuring about the availability of a particular product in the stock through chat window in any order

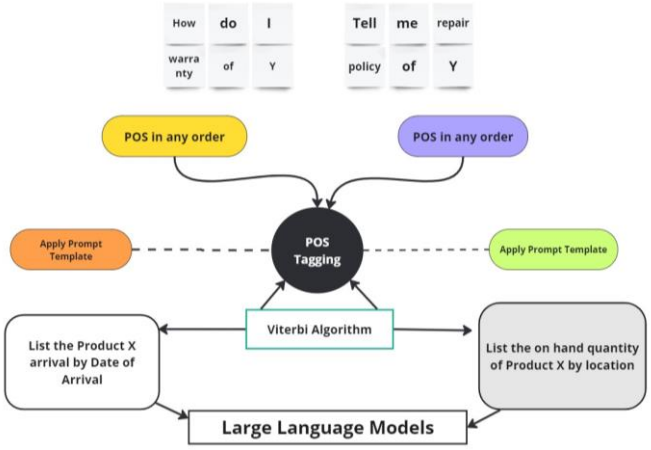

Fig. 1: Sequence flow diagram (Warranty Enquiry)

Solution Construct:

- POS tagging of the chat using Viterbi algorithm
- Apply pre-configured prompt template
- Populate the prompt template
- Generate LLM response in user interactive contextual manner
- Deliver the response back to the user

Business Value: Ensuring the accurate information about stock in hand and inventory management leading to better sales over period.

3.3 Further Study

To interact with the large language model, we need to have very concise prompts with certain predefined structure, regardless of the input we are getting from user text. We can refer to [8] and understand the core concepts surrounding efficient prompt design pertaining to the interaction with LLM. The author describes the fundamental principles of prompt design, including the importance of clear and concise instructions to guide the model's output.

Further to this it is worth mentioning the work of the authors in the research paper [9] where Chain-of-Thought Prompting in LLMs generating a series of intermediate reasoning steps to improve the performance of large language models on complex tasks however aligning to precise steps, hence a general recommendation would be structuring the prompt using Prompt templates.

4. SOLUTION DESIGN

The key design considerations for the using the Hidden Markov model-based Viterbi algorithm in the use case of structuring the prompt in the expected format are:

- The prompt engineering standards of Role-Task Format which I have discussed in the subsequent section 4.2 is not known to the greater community and hence there is a flexibility to enter data in any form which may not be syntactically correct.
- The user query is also taken as a separate input which is subject to produce a response from LLM using API call
- Hidden Markov Model based Viterbi algorithm can automatically structure the Parts of Speech by tagging into multiple allowable possibilities and would consider the provisioned prompt template to come up with the structured prompt using path pruning technique (discussed further in section 4.3).
- The produced structured prompt is now automatically triggered using an event driven serverless architecture to interact with the LLM and produce response on the user query.

4.1 Architecture Solution

I have gone through [10], in which a group of authors have collaboratively presented their work on which they have discussed the intersection of large generative AI models and cloud native architectures and they have deeply investigated the concepts of "containerization" and "orchestration" are major building solution tenets for architectural solution involving Generative AI solution and the authors finds that it is imperative that we stay to the standards of cloud native solution.

The solution I have considered shall therefore align to the following:

- It should be able to consider raw English input and apply POS tagging and automated path pruning to generate the structured prompt in Role-Task-Format specification.
- It should at the same time consider user input which will pass the user data and prompt generated above to the large language model which will generate the response back to the user.
- Cloud native serverless solution is our go to approach to have cost effectivity of the LLMs using event driven trigger of the LLMs only when needed. The Viterbi algorithms will be trained outside this solution and deployed in the solution on Cloud platform.

The solutions show the generic Cloud native services providing the ability of reusability of solution across major public cloud solutions.

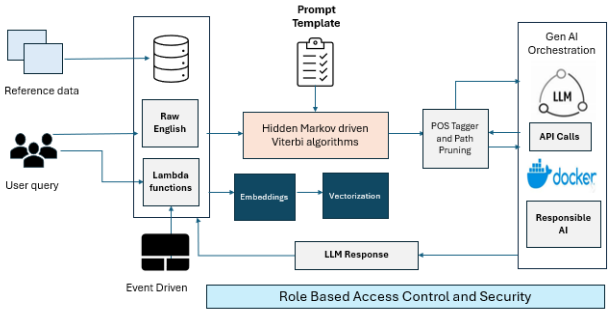

Fig. 3: Solution Architecture (Cloud native but platform agnostic)

The above architecture contains the below building blocks:

- **Data Sources:** The point of origin from where user enters the data into the systems as user queries. For now, considered only textual data (If other modes of data are considered e.g. audio, images etc. then the architecture need to modify for requisite image processing or audio transcripts layers).
- **Event Driving layer:** The data entered as user query; user may also include the data for raw English as input for the prompt design. This will shift the control of the prompt engineering from the AI engineer to the end user.
- **Viterbi layer:** Raw English is considered Hidden Markov state transition system, consideration of the transition and emission probabilities in Markov process followed by possible types of POS tagged statements using the POS tagger. The prompt template will help to realign the POS tagged output into the expected prompt required for LLM interaction. The whole solution will be event driven and hence is automated.
- **LLM Orchestration**: LLM response generated using API calls, LLM orchestration is deployed using docker containers ensures the service based loosely coupled architectural adherence.

The whole solution is governed by cloud platform-based content filtering, data security and role-based access control principles.

With this solution I ensure a digital cloud native yet cloud platform agnostic well governed solution, which is loosely coupled, service based and at the same time I have applied Viterbi algorithm-based POS tagger to generate multiple semantically and syntactically correct prompts which are further tailored using Prompt templates and thereby ensuring the entire process is event driven giving importance to serverless services and automated prompt engineering for LLM interactions.

4.2 Role Task Format

The standard prompt expected should be inclusive a Role-Task – Format standard connotations. In this piece of work, I have explored how a raw English can be POS tagged into multiple possible input and using path pruning shortlisted into the right POS tagging with respect to the context.

Few of the language models, with which I have interacted with are GPT 4 (Open AI), Bloom (Hugging Face), Llama 3.1 and have found a standard RTF format works as expected. Hence the POS tagged English prompt I found from the output of my Viterbi layer when contextualized with standard prompt templates produced the desired Role Task Format specified prompt suitable for the purpose.

Studying the work in [11], the authors have standardized the approach of foundational principles of prompt engineering, such as role-prompting, one-shot, and few-shot prompting, as well as more advanced methodologies such as the chain-of-thought and tree-of-thoughts prompting.

In the context of the 2 business scenarios 1) Stock Enquiry and 2) Warranty Support as discussed earlier in my article in Section 3.1 and Section 3.2. below are my findings of accuracy of responses in terms of the expected response for the 3 LLMs I have considered.

Table -3: LLM Accuracy on Structured RTF prompt

LLM	Accuracy on Stock Enquiry queries	Accuracy on Warranty Support queries
GPT 4 (Open AI)	High accuracy	High Accuracy
Bloom (Hugging Face)	Medium accuracy	High Accuracy
Llama 3.1 (Meta)	High Accuracy	Medium Accuracy

Here are few examples of prompts which were structured in Role Task Format from the output of Viterbi layer and re-casted using standard prompt templates

Table -4: Prompt examples in RTF

Prompt	Stock Enquiry queries	Warranty Support queries
One Shot	**Role**: Inventory Manager **Task**: Provide the current stock levels for all items in the warehouse. **Prompt:** *As an Inventory Manager, your task is to provide the current stock levels for all items in the warehouse for location X. Please list each item along with its quantity*	**Role**: Warranty Support Specialist **Task**: Provide the warranty status for a specific product. **Prompt:** *As a Warranty Support Specialist, your task is to provide the warranty status for a specific product. Please include the warranty period, expiration date, and any coverage details.*
Few Shot	*As an Inventory Manager, your task is to provide the current stock levels for specific items and suggest restocking if necessary. Here are a few examples:* ***Example 1:*** *-Item: X* *- Current Stock: 15* *- Restock Suggestion: Yes, reorder 10 units.* ***Example 2:*** *Item: Y* *Current Stock: 50* *- Restock Suggestion: No, sufficient stock* *Now, provide the stock levels and restocking suggestions for the following items:* *- Item: X* *- Item: Y*	*As a Warranty Support Specialist, your task is to provide the warranty status for multiple products and suggest next steps if the warranty has expired. Here are a few examples:* ***Example 1:*** *- Product: X* *-Warranty Period: 2 years* *- Expiration Date: 2024-08-15* *- Coverage Details: Parts and labor* ***Example 2:*** *- Product: Y* *-Warranty Period: 3 years* *- Expiration Date: 2024-08-31* *- Coverage Details: Parts* *Now, provide the warranty status and coverage for the following products:* *- Product: A* *- Product: B*

4.3. Workflow

With the business scenario examples taken in Section 3.2 with the example below are my step-by-step approach to the workflow

Step 1: Raw Input Processing:

The initial step involves capturing the raw natural language input from the user. For example:

Tell me the repairing policy of Product Y.

Step 2: POS Tagging with Viterbi Algorithm:

The Viterbi algorithm is employed to perform POS tagging on the input, identifying the grammatical categories of each word. The output for the example input is:

Tell/VB me/PRP the/DT repairing/VBG policy/NN of/IN Product/NN Y/NN./.

Step 3: Path Pruning:

Path pruning simplifies the POS tagging output by focusing on the most relevant parts of the sentence for the task at hand. The pruned output for the example input is

Tell/VB repairing/VBG policy/NN Product/NN Y/NN

Step 4: Mapping to Prompt Template:

The pruned output is then mapped to a predefined prompt template. The template is designed to structure the input into a Role Task Format (RTF) prompt. The template used is:

As a Warranty Support Specialist, your task is to provide the repairing policy for specific products. Please include the repair coverage, process, and any associated costs.

Provide the repairing policy for the following product:
- Product: [Product1]

Step 5: Automatic Prompt Generation:

The pruned output is used to fill in the placeholders in the prompt template, resulting in the following structured prompt:

As a Warranty Support Specialist, your task is to provide the repairing policy for specific products. Please include the repair coverage, process, and any associated costs.
Provide the repairing policy for the following product:
- Product: Product Y

5. EVALUATION EXPERIMENT

In the experiment, I have performed the following

- Validated the accuracy of POS tagging & grammatical ambiguity using Viterbi algorithms for a bulk of 100 + raw English input
- With the prompt generated from the previous step, validate the accuracy of the response from the LLM comparing the same with accuracy of manual prompt engineering

As I observed, the way I have defined before in section 4.3 and considering various research study, the POS tagging outcome for 100 prompts comes with less than 1 % margin of error and less 2% of the cases where grammatical ambiguities were observed making Viterbi algorithm a perfect candidate for the purpose

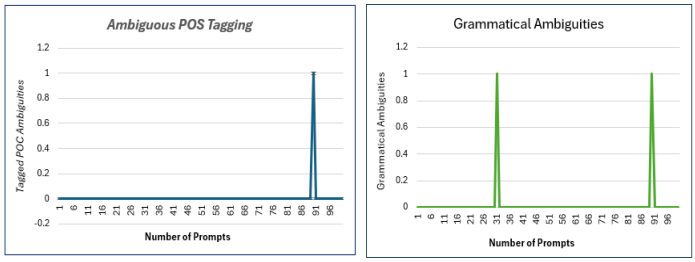

Fig. 4: Margin of error on Generated prompts

(POS Tagging ambiguities and Grammatical ambiguities)

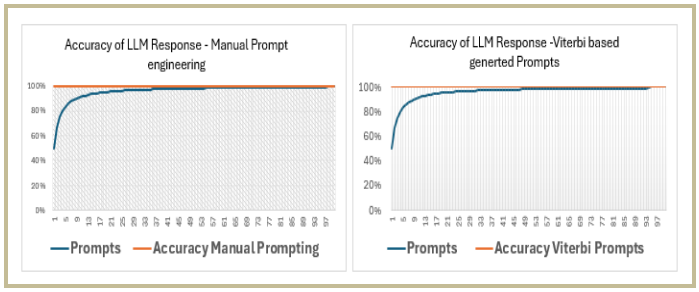

Fig. 5: Accuracy of LLM response

(Manual Prompt and Viterbi generated prompt)

- The accuracy of prompts was measured by comparing the response generated comparing with accuracy generated by manual prompt engineering practice for gamut of 100 prompt cases and 95 cases out of 100 in our experiment yielded the same accuracy as 97 out of 100 when we resort to manual prompt engineering practices. Hence, I conclude that the experiment provides enough evidence to infer that this methodology could be leveraged for greater number of similar use cases in future.

6. CONCLUSIONS

I expect that my work to augment to the existing research work in the field of exploration of automated generation of prompt utilizing the power of Viterbi algorithm for tagging Part of Speech from a raw English statement and associating the prompt templates to generate prompts which is suited to the expected format and structure of prompts suited for interaction with Large Language models

I have observed the prompt design automation in couple of business scenarios obtained from a real-life experience for Stock availability enquiry system and Warranty support system and the response behaviors on three leading large language models in terms of the accuracy generated. I acknowledge this piece of work to yield accuracy as manual generated prompt would need more intelligence into the system but within the scope of my work observed the accuracy and margin of error for both one shot and few shot prompt minimal as compared to the manual time-consuming approach of prompt design. My work will fit in scenarios where analysis of the data is accomplished prior and the Viterbi algorithm will fit into the POS tagging of raw English input, deeper complexity of prompt design can be explored from here on

I, trust this work will motivate upcoming avenues of future research where data input of the request can come from people who are not experts in prompt engineering and the practices and would alleviate their pain with working with Large Language models since the restructuring and reformatting can be taken care of by the solution. The solution can act as an alternate thought process of prompt design where design is taken care of with automated algorithm-based approach to get fruitful response from Large Language models.

REFERENCES

[1] Izzul Fatawi, Muhammad Asy'ari, H. Hunaepi, Taufik Samsuri, Muhammad Roil Bilad "Empowering Language Models Through Advanced Prompt Engineering: A Comprehensive Bibliometric Review", pp-442-444

[2] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal and Aman Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications", pp 2-3

[3] Aarushi Kansal ," Prompt Engineering Techniques " pp 24-28

[4] William Cain, "Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education"

[5] Alebachew Chiche, Betselot Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches", pp-5-6

[6] Jana Diesner, Part of Speech Tagging for English Text Data", pp 3-4

[7] N. Jahnavi, P.V. Parimala, Venkata Vardhan, K. Uday Kishore, L. Ravinandan and G. Rajendra Kumar, "Parts of Speech Tagging Using Viterbi algorithm, pp-69-70, May 2022

[8] Xavier Amatriain, "Prompt Design and Engineering: Introduction and Advanced Methods" ,2024, pp-2-3

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, Denny Zhou: "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models", Jan 2022, pp-3.

[10] Yao Lu, Song Bian, Lequn Chen, Yongjun He, Yulong Hui, Matthew Lentz, Beibin Li, Fei Liu, Jialin Li, Qi Liu, Rui Liu, Xiaoxuan Liu, Lin Ma, Kexin Rong, Jianguo Wang, Yingjun Wu, Yongji Wu, Huanchen Zhang, Minjia Zhang, Qizhen Zhang, Tianyi Zhou, Danyang Zhuo: "Computing in the Era of Large Generative Models: From Cloud-Native to AI-Native**",** Jan 2024, pp- 3

[11] Banghao Chen, Zhaofeng Zhang, Nicolas Langren´e , Shengxin Zhu, "Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review" Jun 2024, pp 4-7

BIOGRAPHY

Sayan Guha completed his Bachelors in Electronics & Communication Engineering in 2006. He is currently pursuing his Masters in Artificial Intelligence. He has been serving Information Technology industry supporting Artificial Intelligence, Data & Analytics Space and Data Architecture across business domains of Retail, Banking, and Insurance & Telecom. He is currently working with leading IT service provider and his area of interests are in the fields of Artificial Intelligence, Cloud Solution architecture, Big Data Integration, Data Modeling, and Information Architecture.