Novel Model Proposal of Hierarchy-of-Thoughts using LLM for Academic Audit

Dennis Jose[1*], Durgansh Sharma[2], Smriti Mathur[3]

[1,2,3]School of Business and Management, CHRIST(Deemed to be University), Bangalore

E-mail:dennisjose.work@gmail.com

Abstract

This paper proposes a novel framework called Hierarchy-of-Thoughts (HoT) for academic auditing using Large Language Models (LLMs). HoT utilizes different prompting approaches tailored to the specific roles and responsibilities of various agents involved in the auditing process. The framework encompasses Chain-of-Thoughts (CoT), Self-Consistency Chain-of-Thoughts (SC-CoT), Tree-of-Thoughts (ToT), and Graph-of-Thoughts (GoT) prompts, catering to the diverse needs of different stakeholders. The proposed methodology employs a data orchestration approach, where prompts from various agents are consolidated into a single thought cloud. This unified representation is then fed into the LLM, generating comprehensive audit outputs. The HoT framework demonstrates promising potential for enhanced privacy, accuracy, efficiency, and adaptability in academic auditing tasks.

Keywords: *Hierarchy-of-Thoughts, Prompting, Large Language Models, Academic Audit, Data Orchestration*

1. Introduction

Artificial intelligence (AI) is the imitation of human intelligence in machines designed to carry out activities requiring human-like problem-solving, reasoning, and decision-making. The emergence of AI has completely changed the world, reshaping sectors and creating new opportunities. In a sense, artificial intelligence (AI) has become the norm. AI is capable of carrying out tasks that are typically done by humans, including as interacting with people, learning, and solving problems [1]. The idea of large language models (LLMs) has been around since the 1960s, when the world's first chatbot, Eliza, was created. Eliza was a simple program that could simulate conversation with humans, but it marked the beginning of research into Natural Language Processing (NLP) and the development of more sophisticated LLMs.

In the 1990s, the advent of Long Short-Term Memory (LSTM) networks made it possible to create LLMs that could handle more complex tasks, such as machine translation and text generation. Transformers are a type of neural network architecture that has become popular in natural language processing (NLP) [2]. Transformers are based on the self-attention mechanism, which allows them to learn long-range dependencies in sequential data. Transformers have revolutionized the field of NLP and have enabled significant progress on a wide range of tasks. [3]

In recent years, there has been a rapid acceleration in the development of LLMs, with the release of models like BERT and GPT-3 that can produce human-quality text and code.
As per [4], prompt engineering has the potential to revolutionize the property valuation industry. The competent use of generative artificial intelligence (AI) models depends fundamentally on the artful crafting of prompts to ensure their outputs align with specified standards and regulations. By delivering precise directives and contextual guidance, property valuers and researchers can effectively steer AI models to generate outputs that are both accurate and aligned with the unique demands of the property industry.

The most recent milestone in the history of LLMs is the release of ChatGPT in 2022. ChatGPT is a chatbot that can carry on conversations with humans in a way that is indistinguishable from a real person. This has led to a growing public interest in LLMs and their potential applications in a wide range of fields [5].

LLMs are still a relatively new technology, but they have the potential to revolutionize the way we interact with computers [6]. They are already being used in a variety of applications, such as machine translation, text generation, and question answering. As LLMs continue to develop, we can expect to see even more innovative and groundbreaking applications in the years to come. [7]

2. Literature Review

LLMs still struggle with complex reasoning and problem-solving tasks that require multiple steps of inference and combining separate pieces of information.
One way to improve the reasoning capabilities of LLMs is to use prompts that guide the model's thought process. [7]

[8] and [9] recently proposed Tree of Thoughts (ToT), a framework that generalizes chain-of-thought prompting and encourages exploration of thoughts as intermediate steps for general problem solving with language models. In other words, ToT is a new way of prompting language models to solve problems that requires more than just a linear sequence of steps. It allows the language model to explore different possible solutions and to make strategic decisions about which path to take.

Chain-of-thought (CoT) prompting is a technique that was introduced in the paper "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models" by [10]. CoT prompting encourages large language models (LLMs) to generate intermediate reasoning steps when answering questions or completing tasks. This can help LLMs to perform better on complex tasks that require reasoning, such as arithmetic, commonsense, and symbolic reasoning.

Tree of thoughts (ToT) prompts represents the problem-solving process as a tree, where each node represents a step in the process and the edges represent the dependencies between the steps. This approach is simple to understand and implement, but it can be limiting for problems that require non-linear reasoning [11] . The main thought is at the top of the tree, and the branches represent the different ways that thought can be expanded or developed.

Self-Consistency Chain of Thoughts (SCCoT) is a decoding strategy proposed by to improve the performance of chain-of-thought prompting in language models. The idea is to sample multiple, diverse reasoning paths through few-shot CoT and use the generations to select the most consistent answer. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer [12].

Graph of thoughts (GoT) prompts represent the problem-solving process as an arbitrary graph, where each node represents a piece of information and the edges represent the dependencies between the pieces of information. [13]This approach is the most flexible of the three, and it allows the model to reason in a non-linear way.

AutoGen is an open-source framework for creating next-generation large language model (LLM) applications that use several agents who may talk with one another to complete tasks. It offers a set of functional systems spanning a wide range of applications from various fields and difficulties, as well as a high-level abstraction for constructing LLM workflows.

3. Proposed Methodology

The genesis of prompting started with Natural Language Process (NLP) [14], which was the supported by Recurrent Neutral Network(RNN). RNN (Recurrent Neural Networks) are a type of neural network that are well-suited for processing sequential data, such as text. RNNs are able to capture long-range dependencies in language, making them an important component of modern NLP models. LSTM (Long Short-Term Memory) [15] is a type of RNN that is specifically designed to address the vanishing gradient problem, which can hinder the training of RNNs. LSTMs have become the dominant RNN architecture for NLP research.

NLU (Natural Language Understanding) is the task of extracting meaning from natural language. It is a crucial component of NLP [16], as it enables computers to understand the intent of prompts and generate meaningful responses.

Transformers are a type of neural network architecture that has revolutionized NLP. They are able to process entire sentences or paragraphs at once, which makes them more efficient than RNNs. They have also been shown to be more accurate than RNNs on a variety of NLP tasks. Dictionaries serve as a fundamental resource for NLU by providing a repository of words, their definitions, and their relationships. They help NLU systems identify and interpret words accurately, enabling them to understand the nuances of language. Vector databases store and manage data in the form of vectors, which are mathematical representations of words or phrases. These vectors capture the semantic and syntactic relationships between words, allowing NLU systems to perform tasks like word similarity analysis and semantic search. Tokens are the basic units of text, such as individual words, punctuation marks, or symbols. They are the raw input for NLU systems, and the accuracy of tokenization significantly impacts the overall performance of NLU tasks. Dictionaries are used by NLU systems to grasp the meaning of words and their relationships. Vector databases enable the representation and storage of words in a form that is compatible with NLU algorithms. Tokens are the input data for NLU systems, and identifying them correctly is critical for reliable NLU outcomes. Transformers serve as the foundational architecture for many LLMs, allowing them to properly process and understand plain language. LLMs use transformer power to generate human-quality writing, translate languages, and execute other NLP tasks [17]). Graph DL approaches can be used to improve LLMs by adding knowledge graphs and extracting insights from entity relationships. Graph DL APIs provide the tools and frameworks required to construct and integrate Graph DL models into LLM applications [18].

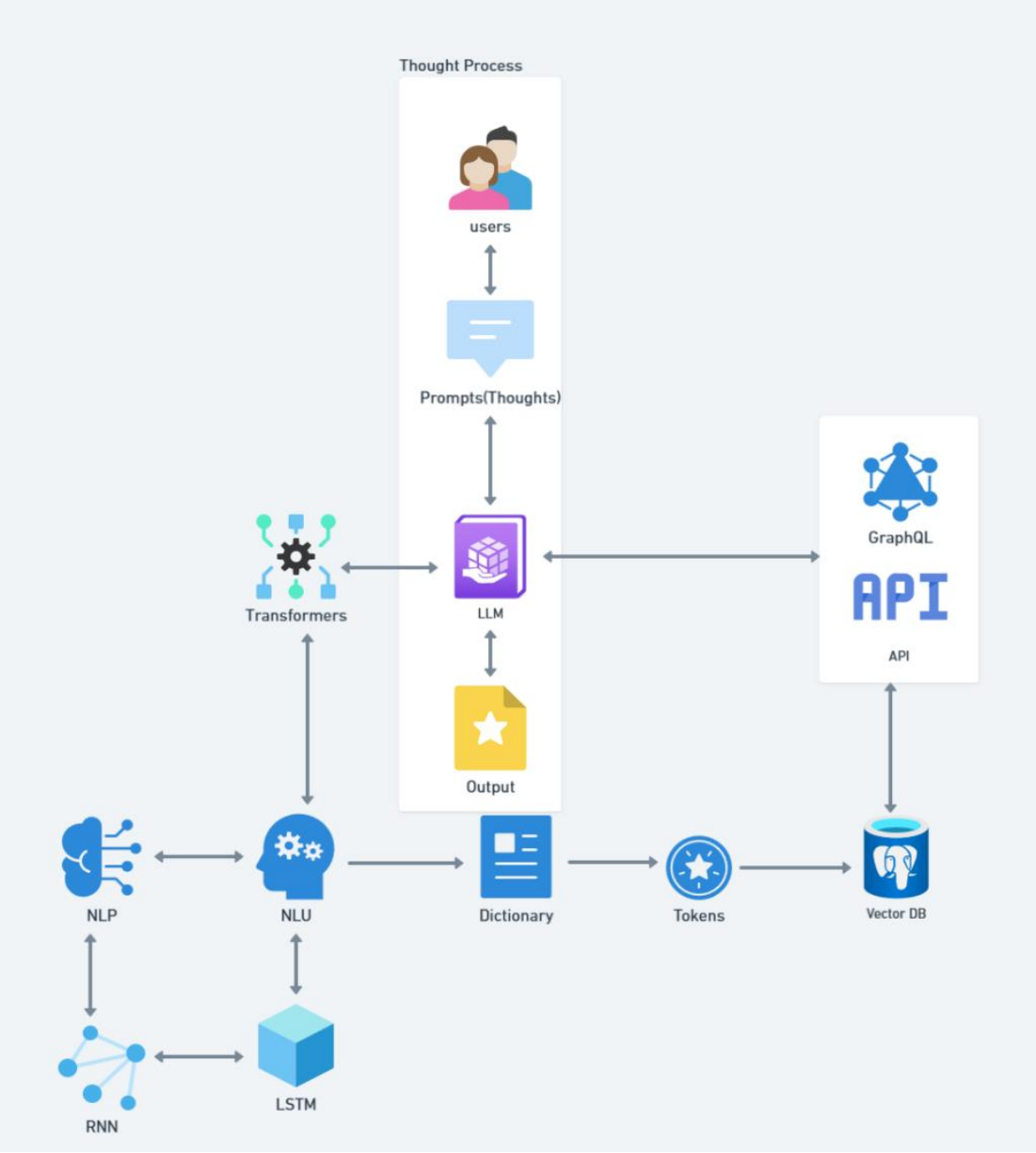

Figure 1: Genesis of LLM

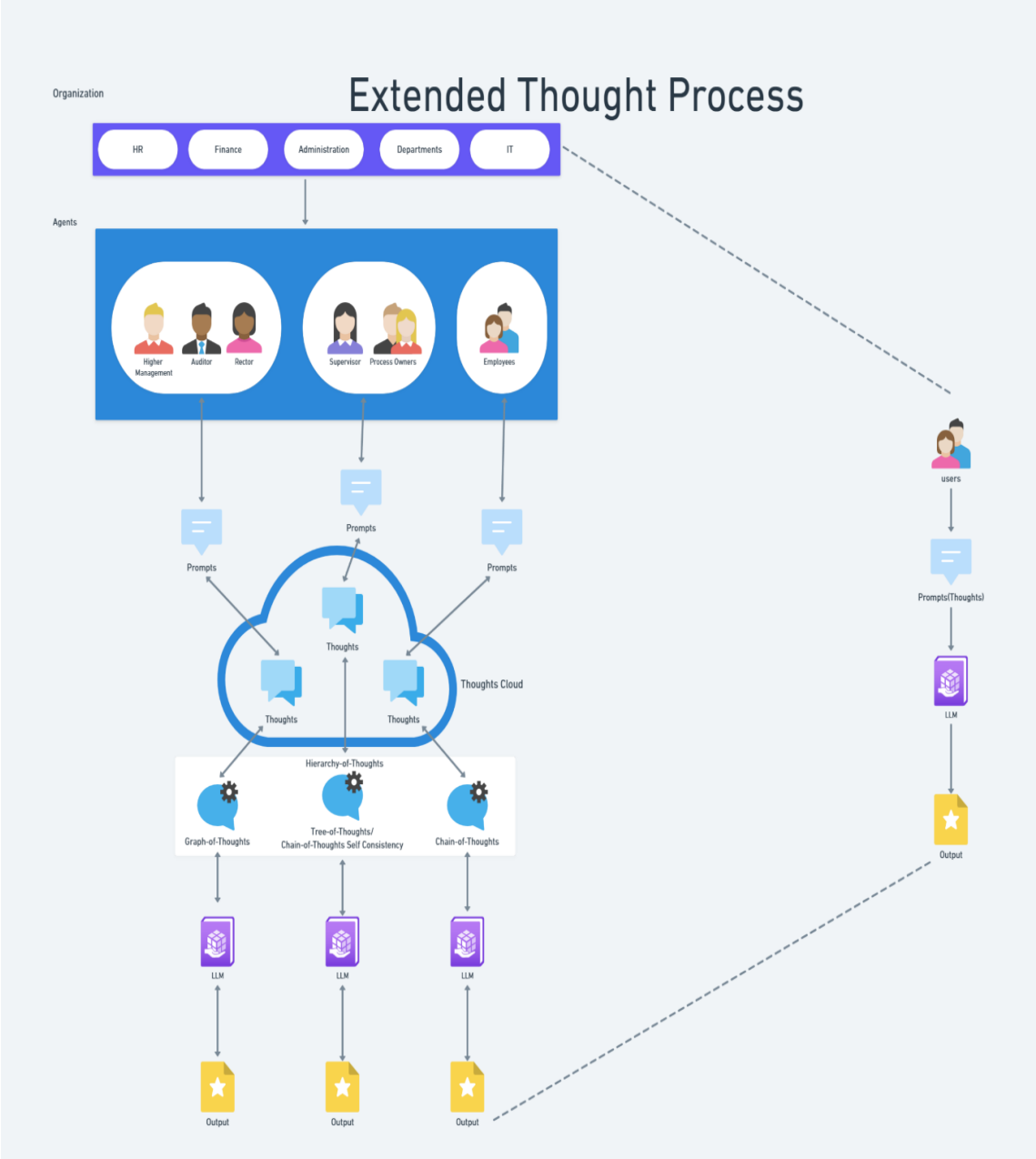

Figure 2: Extended Thought process of Hierarchy-of-Thoughts

3.1 Proposed Methodology Use Case

In the academic auditing, there are different departments in the college or university, which comprises of Human Resource, Finance, Administration, IT, Different Schools/departments of course. Each department have different group of personnel with different authority, roles and responsibilities. Here we have taken 3 types of groups of users as 3 different agents i.e. Employees as agent one, supervisors and process owners, who owns the documentation, as agent two and Top management, Auditor and Rector, as agent three. All these agents access the knowledge base for different purposes. Agent one will access the knowledge base for information retrieval or to know the existing process from the documentation.

Agent two will access the for-documentation creation, revision and self-evaluation process which is done by the supervisor. Agent three will access the knowledge base for audit purposes.

All the three agents will have different approaches for writing the prompt. Agent one will use chain-of-thoughts as they would require retrieving information based on the series of prompts. Agent two will usually be using the tree-of-thought/self-consistency chain-of-thought approach as they have to go through a regress process of approaching of creating or revising a process document and they might have to look for multiple ways. Agent three will be using graph-of-thought approach as they have to audit the documents in every possible way.

All these prompts from the agents are converged into a single thought cloud called data orchestration. Now based on their prompts, it will go to LLM and provide the output as shown in figure 2.

There is an exception that there will be a change in the agent roles and responsibilities due to absence of agent, etc. If the agents are given additional authority, their prompting approaches will also change, this is why the thought are collected in a single thought cloud after which it can be used accordingly based on prompts. Thus, creating a Hierarchy-of-Thoughts approach for the agents to access the knowledge base.

The Hierarchy-of-Thoughts (HoT) is the umbrella term for the thoughts i.e. Chain-of-Thoughts, Self-Consistency Chain-of-Thoughts, Tree-of-Thoughts, and Graph-of-Thoughts for different agents based on their hierarchy in the organization. The word "Hierarchy" is used for the hierarchical approach used in the organization for productivity, privacy, security and compliance purposes.

3.2. Logical Explanation of Hierarchy-of-Thoughts

Let Thought generated from prompts be $T_{n,}$ where n represents the number of thoughts generated.
Thoughts generated from prompt will be { $T_{1,}$ $T_{2,..}$ T_n }

3.2.1. Chain-Of-Thoughts (CoT) Prompts

CoT prompts can be represented using a simple linear equation:

$$CoT(A) = \int f(A, t)\, dt \quad (1)$$

where:

CoT(A) is the output of the CoT prompt for agent A
f(A, t) is the intermediate reasoning step at time t for agent A

3.2.2. Self-Consistency Chain-Of-Thoughts (SCCOT) Prompts

SCCoT prompts can be represented using an extended linear equation that incorporates self-consistency checks:

$$SCCoT(A) = \int [f(A, t) + g(A, t)]\, dt \quad (2)$$

where:

SC-CoT(A) is the output of the SC-CoT prompt for agent A

f(A, t) is the intermediate reasoning step at time t for agent A

g(A, t) is a self-consistency check function that evaluates the consistency of the reasoning steps at time t for agent A

3.2.3. Tree-Of-Thoughts (ToT) Prompts

ToT prompts can be represented using a recursive equation that captures the hierarchical structure of the reasoning process:

$$ToT(A) = \Sigma\, ToT(A_i) + h(A) \quad (3)$$

where:

ToT(A) is the output of the ToT prompt for agent A

$ToT(A_i)$ is the output of the ToT prompt for child node A_i of agent A

h(A) is a function that combines the outputs of the child nodes and adds any additional information from agent A

3.2.4. Graph-Of-Thoughts (GoT) Prompts

GoT prompts can be represented using a system of differential equations that capture the dynamic relationships between information fragments:

$$dF(t)/dt = g(F(t), A) \quad (4)$$

where:

F(t) is the vector of information fragments at time t

g(F(t), A) is a function that updates the vector of information fragments based on its current state and the input from agent A

dF(t)/dt represents the derivative of F(t) with respect to time

3.2.5. Hierarchy-Of-Thoughts (HoT) Prompts

HoT prompts can be represented using a combination of the above equations, depending on the specific prompt type and the role of the agent:

$$HoT(A) = \int_{level=lowest}^{level=highest} (\mathrm{CoT(A)} + \mathrm{ToT(A)} + \mathrm{SCCoT(A)} + \mathrm{GoT(A)}) \quad (5)$$

where:

HoT(A) is the output of the HoT prompt for agent A
CoT(A) is the output of the CoT prompt for agent A
ToT(A) is the output of the ToT prompt for agent A
GoT(A) is the output of the GoT prompt for agent A
SCCoT(A) is the output of the SCCoT prompt for agent A

These equations provide a mathematical framework for understanding the different HoT prompts and their application for auditing tasks with different agents.

4. Conclusion and Future Work

This is a conceptual framework of hierarchy-of-thoughts, where different prompting approaches can be used by different agent(s). This can be implemented further in Thread Intelligence Model for healthcare [19] where the stakeholders can access the patient's data based on their authority or role in the organisation. Similarly, medical professionals who can accessing the patient's medical records via MIST based cyber physical systems [20]. They can further use it for the enhancement of treatment as well as securing the medical records patients as per HIPAA(Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation) standards.

5. References

[1] S. S. Diware, S. R. Dahaput, L. C. Karale, M. R. Bhide and H. M. Raghuwanshi, "A Review on Can AI Replace Humans?," *International Journal For Reseach In Applied Science and Engineering Technology,* pp. 3420-3422, 2023.

[2] K. Guu, K. Lee, Z. Tung, P. Pasupat and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, 2020.

[3] S. Doddapaneni, G. Ramesh , M. M. Khapra, A. Kunchukuttan and P. Kumar, "A Primer on Pretrained Multilingual Language Models," *arXiv.org,* 2021.

[4] K. S. Cheung, "Unleashing the potential of ChatGPT in property valuation reports: the "Red Book" compliance Chain-of-thought (CoT) prompt engineering," *Journal of Property Investment & Finance,* 2023.

[5] A. Aghajanyan, D. Okhonko, M. Lewis, M. Joshi, H. Xu, G. Ghosh and L. Zettlemoyer, "HTLM: Hyper-Text Pre-Training and Prompting of Language Models," *arxiv.org,* 2021.

[6] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards , Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray , N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever and W. Zaremba, "Evaluating Large Language Models Trained on Code," *arvix.org,* 2021.

[7] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan and J. Ba, "LARGE LANGUAGE MODELS ARE HUMAN-LEVEL: PROMPT ENGINEERS," *arxiv.org/pdf/2211.01910,* 2023.

[8] J. Long, "Large Language Model Guided Tree-of-Thought," *arxiv.org/pdf/2305.08291,* 2023.

[9] Y. S., L. M. and Zhao, H , "Tree of thoughts: Deliberate problem solving with large language models.," *arXiv preprint arXiv:2305.10601.,* 2023.

[10] J. Wei, . X. Wang, D. Schuurmans, M. Bosma, B. Ichter, . F. Xia, E. Chi, Q. Le and D. Zhou, "Chain of thoughts: A simple and effective prompt for large language models on reasoning tasks," *arXiv:2201.11903 ,* 2023.

[11] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee and E.-P. Lim, "Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought," *https://arxiv.org/pdf/2305.04091,* 2023.

[12] P. Wang, . Z. Wang, Z. Li, Y. Gao, B. Yin and X. Ren, "SCOTT: Self-Consistent Chain-of-Thought Distillation," *https://arxiv.org/abs/2305.01879,* 2023.

[13] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, L. Gianinazzi, J. Gajda, T. Lehmann, M. Podstawski, H. Niewiadomski, P. Nyczyk and T. Hoefler, "Graph of thoughts: Solving elaborate problems with large language models.," *https://arxiv.org/pdf/2308.09687,* 2023.

[14] S. Singh and A. Mahmood, "The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures," *IEEE Access,* vol. 9, pp. 68675-68702, 2021.

[15] E. Chemali, P. J. Kollmeyer , M. Preindl, R. Ahmed and A. Emadi, "Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 8, pp. 6730-6739, 2018.

[16] C. Dupuy, R. Arava, R. Gupta and A. Rumshisky, "An Efficient DP-SGD Mechanism for Large Scale NLU Models," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 4118-4122, 2022.

[17] S. Kang, J. Yoon and S. Yoo, "Large Language Models are Few-shot Testers: Exploring LLM-based General Bug Reproduction," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023.

[18] DAIR.AI. [Online]. Available: https://www.promptingguide.ai/techniques/graph.

[19] D. Sharma, T. K. Singhal and D. Singh, "Threat Intelligence Model to Secure Iot Based Body Area Network and Prosthetic Sensors," *ECS Transactions,* vol. 107, no. 1, pp. 15417-15425, 2022.

[20] D. Sharma, T. K. Singhal, D. Singh and A. Qadir, "MIST-based Tuning of Cyber-Physical Systems Towards Holistic Healthcare Informatics," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2022.