Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421p-ISSN: 2958-0544

Prompt Engineering: Unleashing the Power of Large Language Models to Defend Against Social Engineering Attacks

Ahmed I. Nezer[1], Prof. Bashar M. Nema[1*], Wisam Makki Salim[2]

[1]Department of Computer, College of Science, Mustansiriyah University, Baghdad, 14022, Iraq.
[2]College of Dentistry, Al-Iraqia University, Baghdad, 10011, Iraq.

*Corresponding Author: Prof. Bashar M. Nema

DOI: https://doi.org/10.52866/ijcsm.2024.05.03.024
Received February 2024; Accepted May 2024; Available online August 2024

ABSTRACT: Prompt Engineering is an emerging area of study that pertains to the act of conceptualizing, perfecting, and executing prompts that guide an AI model to an intended purpose. The AI model is an LLM, which they are the "hit" of our time and probably the controversial type of AI. They are capable of executing several tasks using natural language processing algorithms. Due to their ease of use and fast development, they are becoming highly dependent. We found that to interact correctly with these models and gain the best performance, several techniques should be taken into consideration. Moreover, there are additional methods or tips to write a good prompt.

Keywords: Large Language Models, ChatGPT, Prompt Engineering, Prompting, Natural Language Processing, Human-Computer Interaction.

1. INTRODUCTION

In recent years, artificial intelligence (AI) technologies exemplified by models similar to OpenAI's GPT (Generative Pre-trained Transformer), have demonstrated remarkable proficiency in understanding and generating human-like text [1,2], it has become a central theme in the progression of language models. Prompt Engineering is an emerging area of study that pertains to the act of conceptualizing, perfecting, and executing prompts (a prompt is simply a text that is given to a Large Language Model) or directives that steer the outcome of Large Language Models (LLMs) with the intention to help in various tasks [3]. In other words, this approach is concerned with strategically crafting input prompts to guide the model's responses in the desired directions.

Large Language Models are the peak of AI of our time. They have some ahead-of-time capabilities (a result of using deep learning techniques [4]) such as human-like text generation [5], contextual awareness, and robust problem-solving skills, making them invaluable in various domains [6]. An AI model should possess four features to be called an LLM: deep comprehension of natural language for tasks such as translation; ability to create human-like text; contextual awareness, especially in knowledge-intensive domains; and strong problem-solving and decision-making using text-based information for tasks [7]. They offer a wide range of applications and services across various domains such as healthcare [8], customer support [9], code generation and evaluation [10], finance [11], and education [10, 11]. They are categorized into two categories, Encoder-Decoder or Encoder-Only (BERT-style LLMs) and Decoder-Only (GPT-style LLMs). BERT-style LLMs predict the masked words in a sentence, considering the surrounding context simultaneously. This capability allows the model to enhance a deeper understanding of the relationships between words and the context in which they are used. They began to fade away after the emergence of decoder-only models. Notable examples of these LLMs are BERT [14] and T5 [15]. GPT-style LLMs generate the next word in a sequence given the previous words, this is the best paradigm for improving zero-shot and few-shot performance. These models are widely used for text generation and question answering. Notable examples of this type are GPT-3 [16] and BLOOM [17]. LLMs appeared and evolved rapidly in recent years, the following figure shows LLMs escalated through the years:

*Corresponding author: bmn774@uomustansiriyah.edu.iq

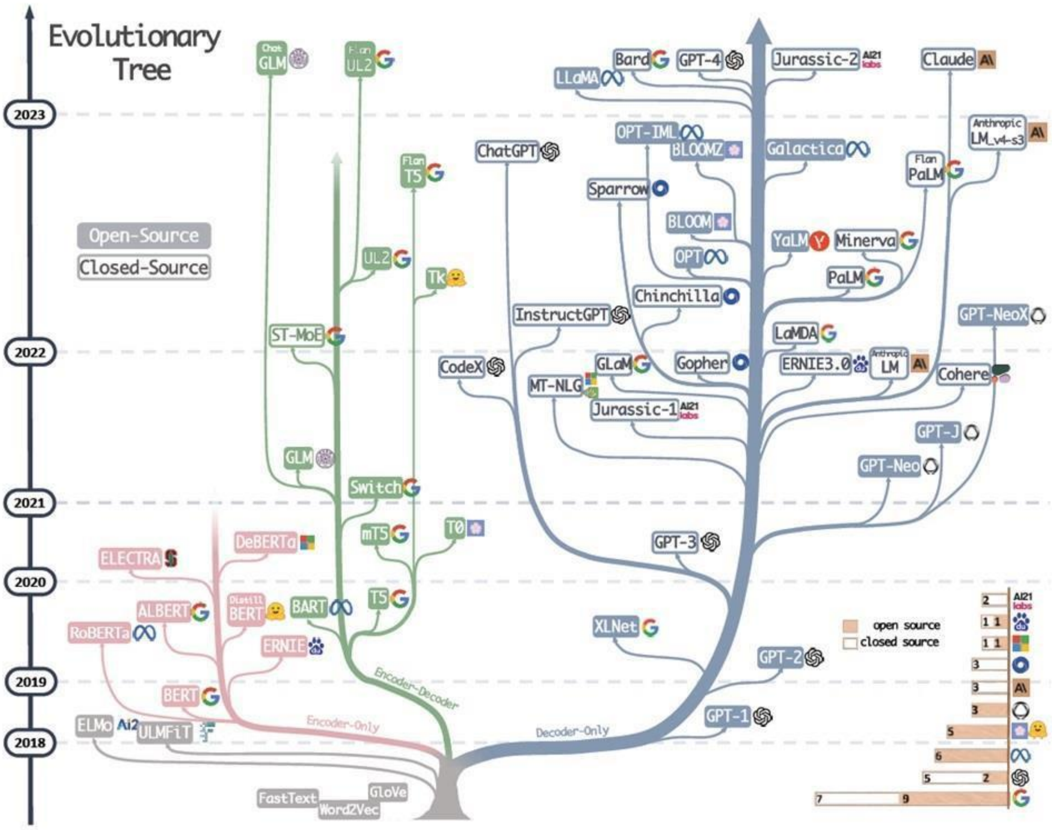

Figure1. - LLMs evolution [7].

Despite all of these abilities, types, and fast evolution, LLMs are still primitive and prone to make errors. Many of these errors are a result of a wrong prompt. This research explores the nuanced relationship between prompt design and model output, delving into the intricacies of linguistic cues, contextual nuances, and the influence of specific instructions on the AI generation.

2. PROMPT ENGINEERING

Communication with AI is substantial and understanding how to communicate with it effectively is useful. The communication process revolves around writing commands which are referred to as "prompts". With that said, prompt engineering can be defined as the systematic process of creating inputs that determine the output to be generated by an AI language model [18]. High-quality inputs will lead to better output. Similarly, poorly defined prompts will produce inaccurate responses or responses that might negatively impact the user. This is why the quality of outputs depends on how much information is provided to an LLM and how well-crafted it is [19].

Prompts can be categorized into several types based on their shape. These categories can be used to structure and understand a robust prompt. In general, prompts can be cloze prompts [20], which are fill-in-the-blank prompts (i.e., Marlon Brando was born in ______) or prefix prompts [21], which continue a string prefix (i.e., ________ is the father of quantum physics). They can be information-seeking prompts, which are the prompts that are used to gather information and mostly start with "what", "where", …, (i.e., what is the largest star of our galaxy?), instruction-based prompts, which start with an order (i.e., generate a short romantic story), context-providing prompts, which contains details to help the LLM to understand the prompt better, opinion-seeking prompts, which asks the LLM about its opinion, or reflective prompts, which can be used to get a deeper understanding of a certain belief, action or to get the help of making decisions [12], [22 – 24].

It seems obvious that LLMs will play a crucial role in the future. This technology will continue to enhance daily routines at work, home, or school. To benefit from this, an individual must be able to interact effectively with these systems. That is where prompt engineering emphasizes. By understanding how to craft a robust prompt, the interaction between humans and machines can be improved.

3. SOCIAL ENGINEERING

"The fact that you could persuade someone to plug a computer in and switch it on means even that powered down, computers are vulnerable." This is a common hacker moto that led to social engineering. Social Engineering is the act of acquiring information via technical and non-technical means. It can be achieved using manipulation and psychological tricks to make the victim assist the attacker in their attack [25]. Although social engineering is still in its early stages, it has proven to be very dangerous, sometimes more than other types of cyberattacks [26]. There a several types of social engineering attacks, including [26], [27], [28], [29], [30]:

• **Phishing:** This type of attack occurs when an attacker scams their victim through socials (usually emails) to hunt for information, using fake messages that look identical to ones that come from trusted people or organizations. For example, an attacker might send an email claiming to be from the victim's bank, stating that their account's password has been compromised. Because the email looks authentic and the message feels urgent, they will comply willingly without a second thought.

• **Pretexting:** Occurs when an attacker creates a fake persona, or fake scenarios, or misuses their actual role to steal a victim's information. It is what often happens with data breaches from the inside. The attack is performed via phone calls, emails, or physical media. This is what Edward Snowden did with his coworkers.

• **Tailgating or Piggybacking:** In this type of social engineering attack, an authorized person grants an unauthorized person access to restricted data. For example, following an employee to enter a certain organization, disguising as a delivery driver, or pretending that they are new and forgot their ID.

• **Quo Pro Quo:** "A favor for a favor." This attack occurs when an attacker offers a free service to seduce the victim in exchange for their personal credentials. For example, an attacker may call a victim pretending to be an employee of his ISP (Internet Service Provider), offering to speed up the connection.

LLMs are capable of preventing social engineering attacks. They can be used in the field of digital forensics to detect anomalies, fake emails, and accounts. Since social engineering attacks occur in social networks, people can harness their abilities using prompt engineering techniques to steer them into defending against cyberattacks, even asking LLMs for advice can be beneficial [31].

4. RELATED WORK

Several works can be referenced, including:

In 2023, Li et al. [32] created "Communicative Agents for "Mind" Exploration of Large Language Model Society" (CAMEL), which is a communicative agent to study the "mind" of LLMs. Their approach involves using inception prompting that allows agents to provoke each other to solve tasks by role playing, to guide LLMs toward task completion while maintaining consistency with human intentions. However, the study showed how role playing can provide information for investigating LLMs.

In 2023, Zhou et al. [33] proposed an "Automated Prompt Engineer" (APE) for prompt generation and selection, inspired by program synthesis and the human approach to prompt engineering. The researchers showed that APE prompts are able to control and manifest models toward "truthfulness and informativeness" and improve few-shot learning performance.

Another study in 2023 by Mialon et al. [34], showed that LLMs augment their context through additional relevant information, received from prompts. It can help the LLM to perform missing token prediction. This is a resilient ability that accelerates expanding the LLM's knowledge and generating more accurate statements. Furthermore, White et al. [35] provided a pattern of prompts when communicating with LLMs, leveraging the ability of a particular LLM to produce a reusable solution to a common problem of a certain context (i.e., creating a login page using C#). These prompt patterns can help users to interact with LLMs more effectively and efficiently. This was mentioned earlier by Scao and Rush [36], in 2021. Their systematic study showed that prompting indeed improves the training process and that it is robust to patterns.

Bach et al. [37], 2022, created "PromptSource". It is an open-source system for generating and sharing prompts. It enhances the model's flexibility by training it using new collaborative tools (allows the user to refine the generated prompts), in addition to setting a template for declaring data-linked prompts.

Emergent Abilities were discussed by Wei et al. [38] in 2022. They are present only in large models. The researchers discovered that these abilities are an outcome of scaling up LLMs with prompts.

Prompt engineering has been extended to computer vision through the creation of CLIP (Contrastive Language-Image Pretraining), by Radford et al. [39], 2021. The researchers trained CLIP to minimize the loss, enabling it to classify images based on natural language instruction.

5. PROMPTING TECHNIQUES

As shown before, the main purpose of studying prompts is to craft inputs for deep generative models [40]. To achieve such intent, several techniques can be used:

5.1 Zero-Shot Prompting

The most straightforward method of prompting. It is how most people interact with LLMs by providing them "orders" without any previous explanation or examples (As shown in Figure 2), in short, they deal with LLMs as chatbots. Despite the fact that this technique has been used for a long time now, LLMs' performance in zero-shot learning is poor [41]. By employing zero-shot prompting, the efficiency and zero-shot learning capabilities of an LLM can be significantly enhanced, enabling users to achieve their desired outcomes more effectively and efficiently [42].

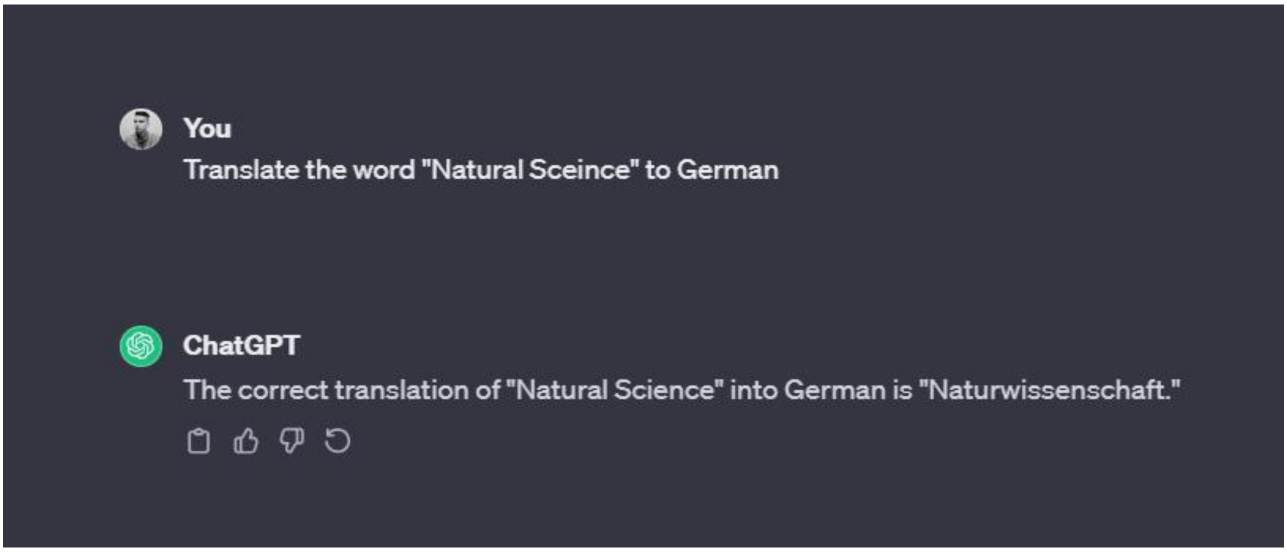

Figure2. - Example of Zero-Shot Prompting [43].

5.2 Few-Shot Prompting

It is an effective strategy that can lead the model to generate accurate and appropriately structured responses. By providing multiple examples (as shown in Figure 3), few-shot prompting enables the model to understand the desired output format and respond accordingly, making it a preferred method over zero-shot in most scenarios [44].

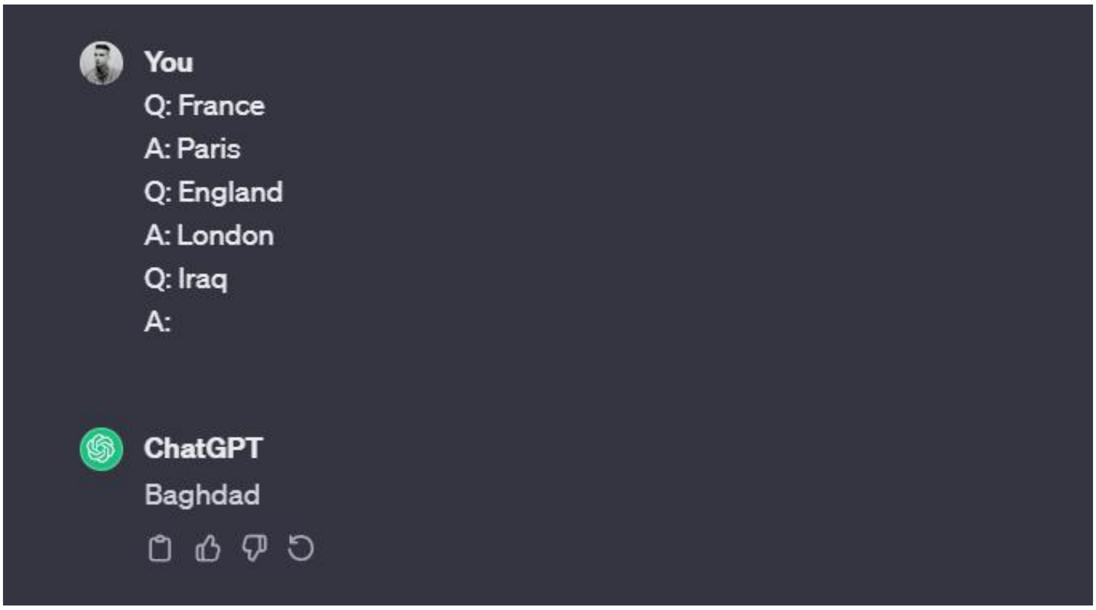

Figure3. - Example of Fero-Shot Prompting [43].

5.3 Chain-of-Thought Prompting

A chain of thought is a series of reasoning steps that enhances the LLMs' abilities of complex reasoning, because are not trained to solve these types of problems (Figure 4). Each step is provided with demonstrations as an example of prompting (as shown in Figure 5). This method was suggested by Wei et al [45]. The researchers found that chain-of-thought prompting improves effectiveness on a range of arithmetic, commonsense, and symbolic reasoning tasks.

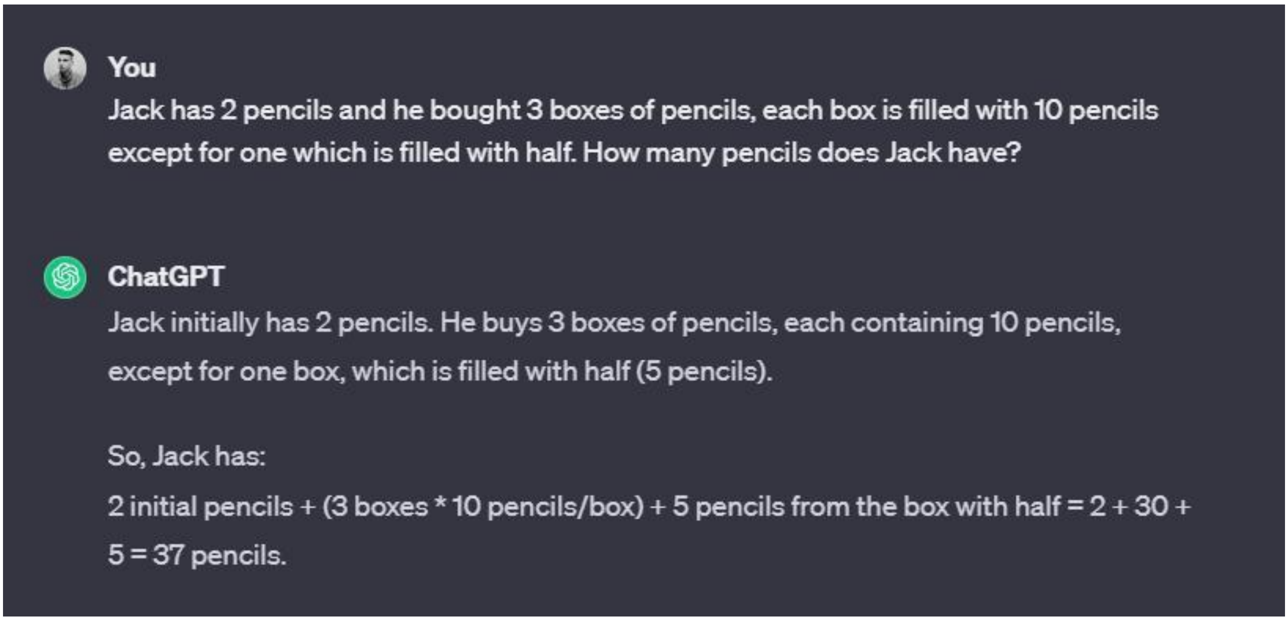

Figure 4. - Example of a simple reasoning problem. ChatGPT could not solve it correctly [43].

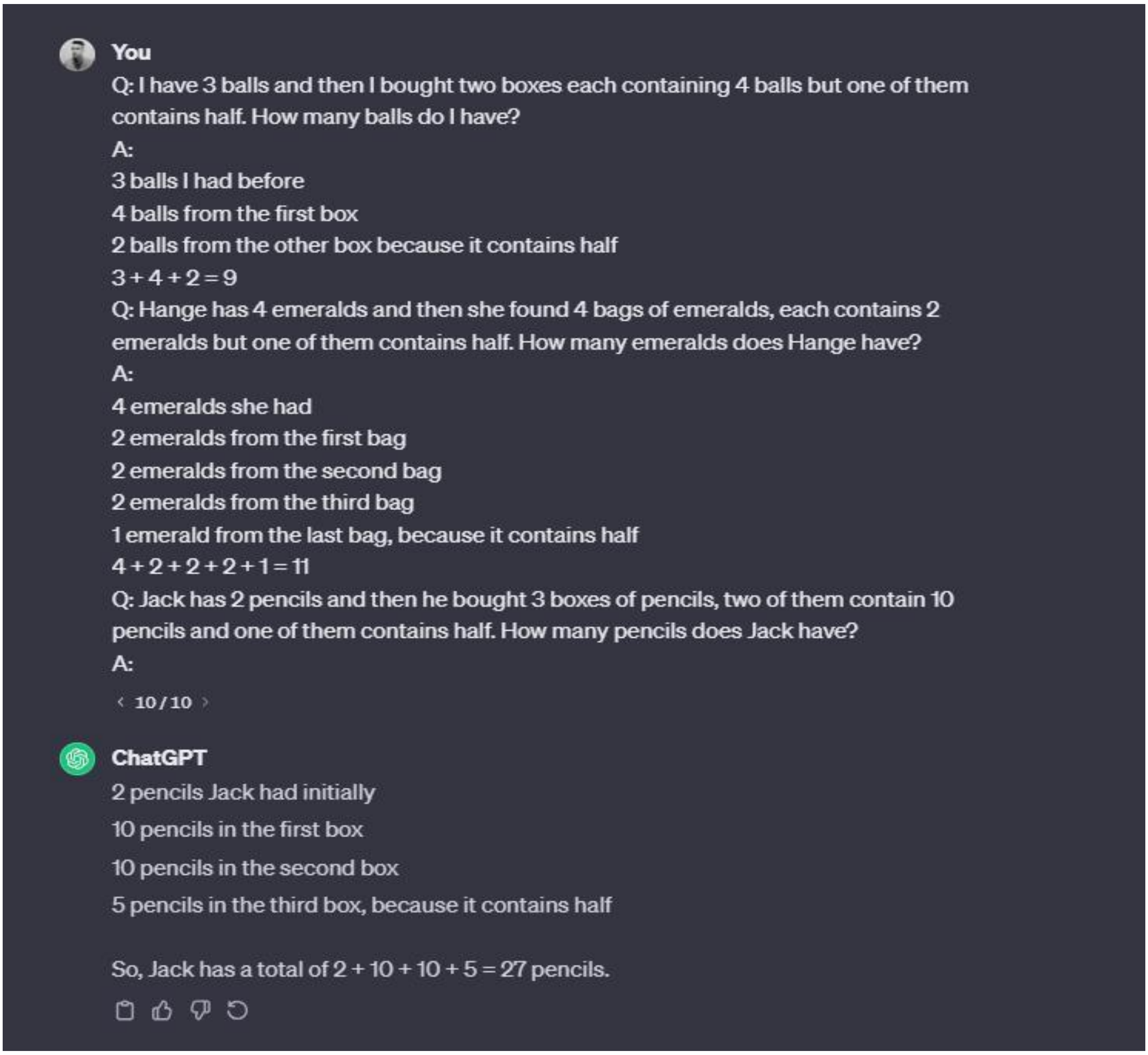

Figure 5. - The same problem but with chain-of-thought prompting. We notice that ChatGPT has learned how to solve it correctly depending on the series of thoughts [43].

5.4 Self-Consistency Prompting

This technique aims to subrogate the naive greedy decoding method employed in the process of chain-of-thought prompting [46]. The idea is to sample multiple, diverse reasoning paths through few-shot chain-of-thought prompts, and use the generations to select the most consistent answer. Self-consistency raises the intuition that a complex reasoning problem typically admits multiple different ways of thinking (increase the performance of prompting on tasks involving arithmetic and commonsense reasoning) leading to its unique correct answer (for full prompts, click here) [43].

6. ENHANCEMENT OF A PROMPT

Even with using the aforementioned techniques, LLMs are likely prone to make errors or undesired outputs. To make a prompt more precise and the output more accurate, a set of flavors can be added [47 – 50]:

- **Assigning a Task:** It is what we regularly do as illustrated in the above figures. Giving the LLM a certain task results in a specified outcome although it can push the model to generate fabrications.
- **Clarity of the Context:** This is an important trait of a good prompt. LLMs respond to prompts depending on the choice of words. Clearer and more decisive prompts drive LLMs' outcomes to be less fallible and more

advantageous (as shown in Figure 6). When the model was asked again using another context, the generated outcomes were better (Figure 7, you can find the full prompts here) [43].

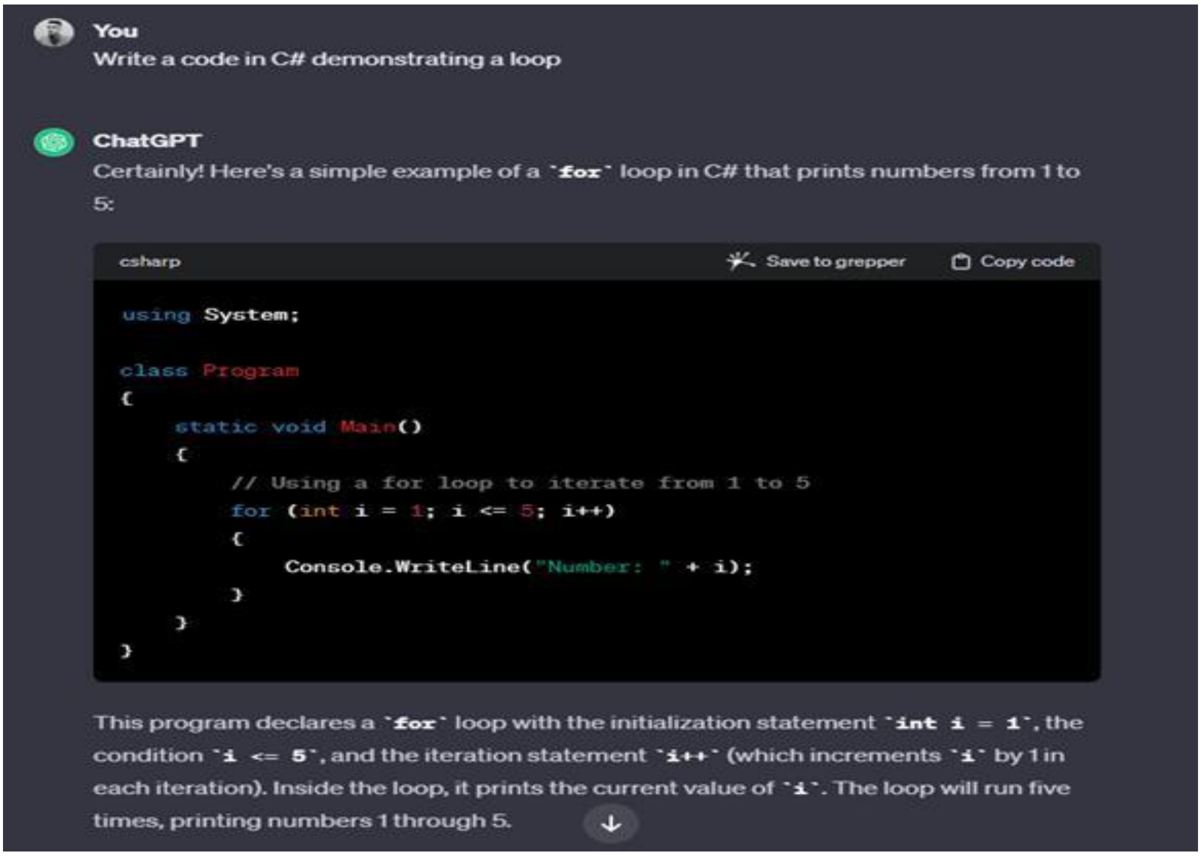

Figure 6. - C# for-loop example created by ChatGPT [43].

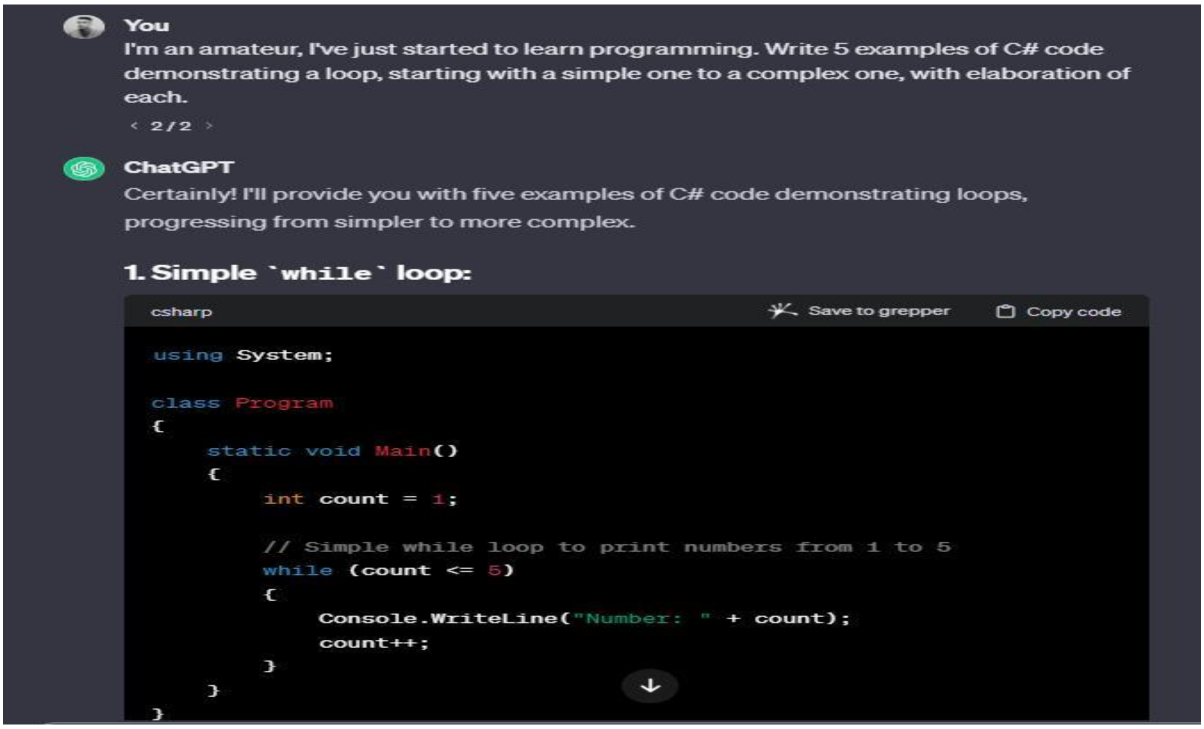

Figure 7. - C# Loop examples with more details and explanations, from simpler to more complex [43].

• **Assigning Roles:** Roles are very important to get specific information on a certain topic, generating subjective and robust outputs (Figure 8).

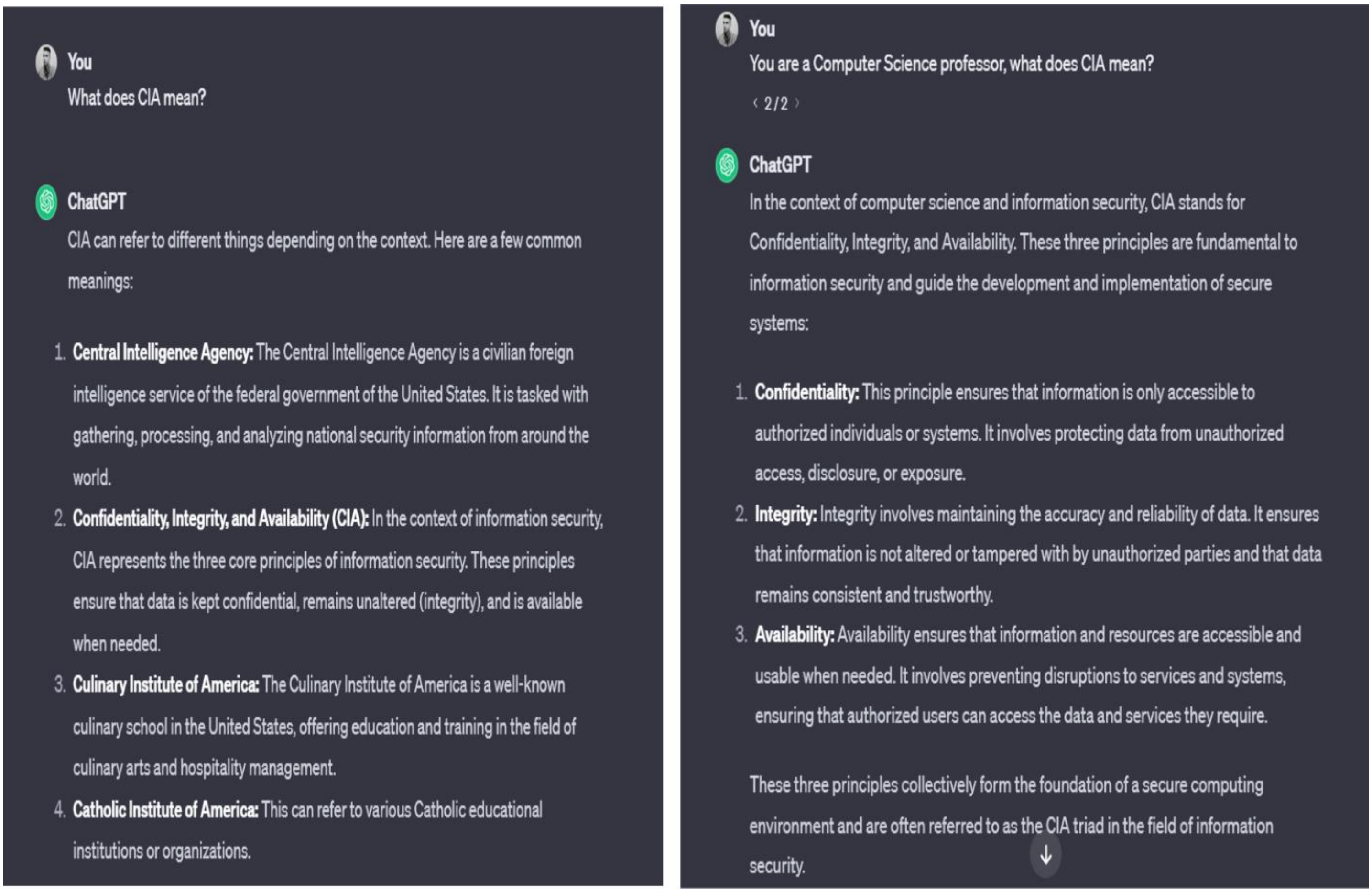

Figure 8. - How the roles affect the output [43].

• **Output Specification:** It is very useful and efficient to indicate the output type, for example, representing the number of cars in each state within a particular country in the form of a chart, extracting the correlation between multiple surveys and viewing the result as a table, converting a text into a hash code or generating a json file from a given paragraph (Figure 9).

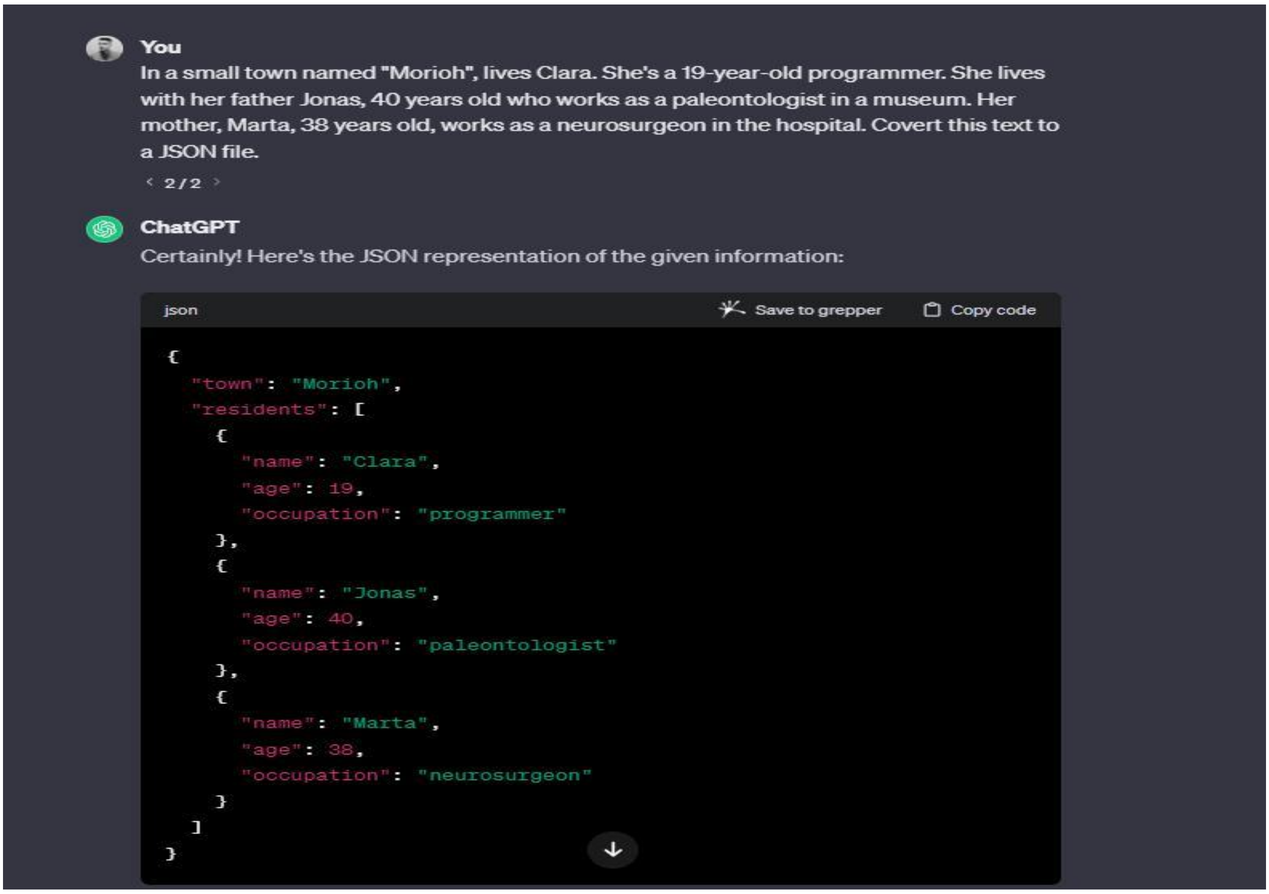

Figure 9. - **Indicating Output Example [43].**

• **Indicating Input Data:** This is helpful when a detailed input is prompted. It is favorable to show facts and rules in order to get a satisfactory output. Consider the following example in Figure 10 (for full prompts, click here) [43].

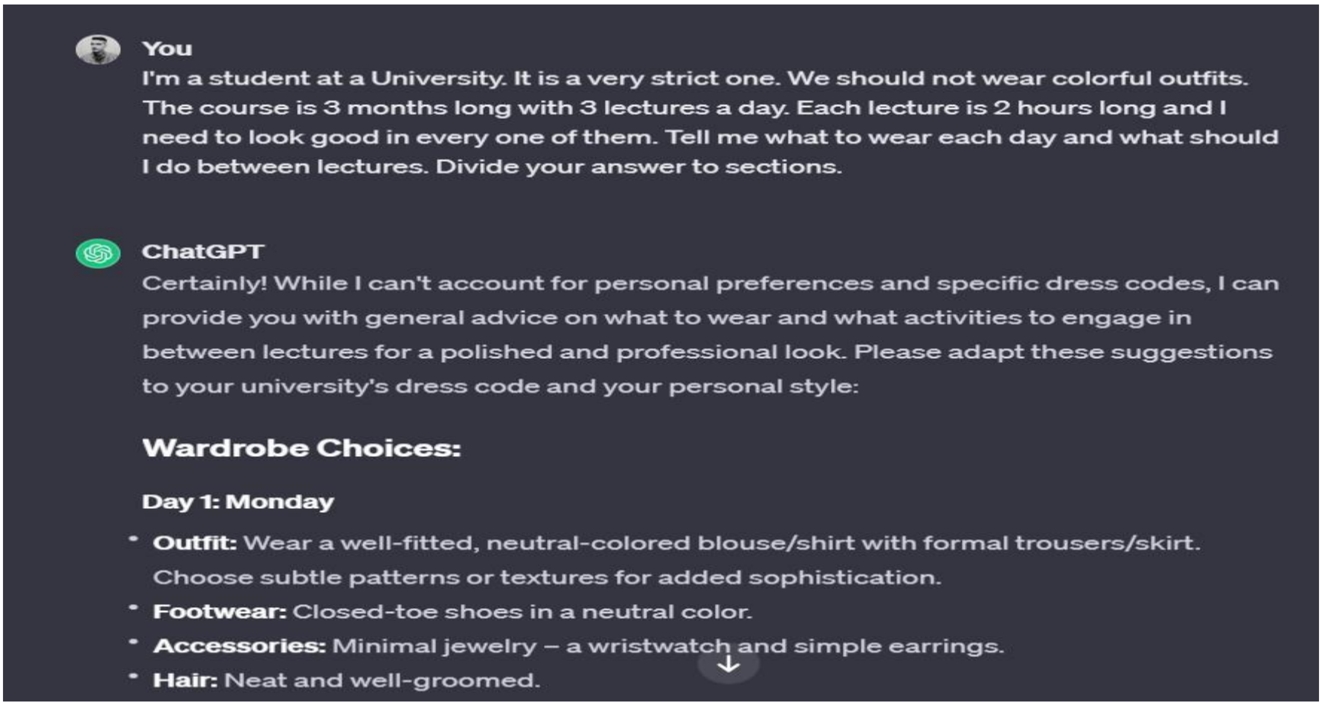

Figure 10. - **Indicating input data example [43].**

7. DISCUSSION

LLMs are the peak of AI nowadays thus improving them has become a common procedure [51]. They can do several tasks effortlessly.

They ease the process of a certain job and reduce the time required to do it, subsequently resulting in increasing the levels of efficiency and efficacy for both the user and the computer. Regrettably, people treat these models as chatbots only (using zero-shot prompting). If a model was asked the following, list 100 tasks you can do, the answer would be astonishing. The model of course cannot fulfill all of them, yet there are tasks that the model is capable of performing that exceed our expectations. On the other hand, these models are not infallible (even with the use of prompting techniques and prompting elements). They may generate misleading information or hallucinations, be biased toward what they were trained on, adhere to the rules and guidelines of their respective founding institutions, and so forth. Although the efforts to diminish these flaws are ongoing, LLMs are still primitive. Accordingly, the importance of prompting became evident.

As shown in this endeavor, the relationship between prompts and the outcome is strong, and learning how to deal with them not only enhances the performance of LLMs but also mitigates their faults. Additionally, prompting without considering the choice of words and prompting techniques may lead to the occurrence of their faults frequently.

In summary, it is of utmost importance to emphasize the significance of prompt engineering in the current state of LLMs. As users, researchers, and developers navigate the landscape of these powerful models, a strategic and mindful approach to crafting prompts becomes imperative for unlocking their full potential while mitigating the risks associated with their inherent limitations. The ongoing evolution of prompt engineering techniques will likely play a significant role in advancing the capabilities and reliability of LLMs in the future.

8. Conclusion

Prompting is a very important skill that enables anyone to get a better result from using the LLMs. Using the aforementioned techniques and elements is beneficial to the improvement of the AI models. These techniques can get even better accompanied by some tactics and ethical considerations.

8.1 Prompting Strategies for Flawless LLM Interactions

To ensure that the output is as required or as close as it can get, a number of strategies can be taken into consideration:

• **Providing reference text:** Language models can generate fake answers, especially when asked about specific topics or for citations and URLs. Just as a student can enhance their performance on an examination by using a sheet of notes, providing reference text to these models can assist these models in furnishing responses with fewer instances of fabrication.

• **Simplifying complex tasks:** It is an effective method to divide a task into simpler ones, in a way similar to software engineering when a problem is decomposed into smaller modules. It has been observed that complex tasks are more prone to errors compared to their simpler counterparts. Furthermore, complex tasks can often be constructed from the outputs of previous simpler tasks, as long as they can be used as inputs to later complex tasks.

• **Giving the model time to "figure it out":** If an individual was asked to multiply 15 by 25, they might not know it instantly, but with time, they can still work it out. Similarly, models are prone to make more reasoning errors when trying to answer directly, rather than taking time to answer. A chain of thought can help the model reason its way toward reliable correct answers.

• **Using additional tools:** Reduce the weaknesses of the model by prompting the outputs of other tools. For example, a code execution engine like Replit can help the model do the math and run code. This combination is a more efficient and reliable way to get the best of both.

8.2 Future Directions and Ethical Considerations

Prompt Engineering holds immense promise for the future of human-computer interaction. Future research should focus on:

- **Developing advanced prompting techniques that address model bias, explainability, transparency, and evolving user needs**.
- **Exploring the ethical implications of Prompt Engineering:** including issues of fairness, manipulation, and privacy.
- **Establishing guidelines and best practices for responsible development and deployment of Prompt Engineering technologies**.

Here are some specific examples of how Prompt Engineering could be developed in the future:

• **Using AI to help create more accurate and effective prompts:** For example, AI could be used to analyze user data and training data to identify patterns and relationships that could be used to improve prompts.
• **Developing user-assistance tools to create more efficient prompts**: These tools could include a graphical user interface or step-by-step instructions to help users create accurate and effective prompts.
• **Creating standards for evaluating the efficiency of Prompt Engineering:** These standards could help ensure that Prompt Engineering techniques are effective and reproducible.

By conducting more research and developing advanced techniques, we can improve Prompt Engineering and make it a more powerful and effective tool for human-computer interaction. In addition, it is important to take into account the ethical implications of Prompt Engineering. For example, Prompt Engineering could be used to create misleading or harmful content, or it could be used to reinforce bias or discrimination. It is important to develop Prompt Engineering techniques in a responsible and ethical way. Here are some specific examples of ethical considerations for Prompt Engineering:
• **Fairness:** Prompt Engineering should not be used to create discriminatory or unfair outputs.
• **Manipulation:** Prompt Engineering should not be used to manipulate or deceive users.
• **Privacy:** Prompt Engineering should be used in a way that respects user privacy.
By carefully considering these ethical considerations, we can ensure that Prompt Engineering is used for good.

Funding

None

ACKNOWLEDGEMENT

I express my sincerest gratitude to my supervisor, Prof. Bashar M. Nema, for their invaluable guidance and support throughout this research journey. Their expertise and insightful feedback were instrumental in shaping my understanding of prompt engineering and refining my work. I am also grateful to the College of Science and the Computer Science department at the University of Mustansiriyah for providing the resources and environment conducive to academic exploration.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] Q. Zhu and J. Luo, “Generative Pre-Trained Transformer for Design Concept Generation: An Exploration,” Proc. Des. Soc., vol. 2, pp. 1825–1834, May 2022, doi: https://doi.org/10.1017/pds.2022.185
[2] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, “Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models.” arXiv, Mar. 08, 2023. doi: https://doi.org/10.48550/arXiv.2303.04671.
[3] B. Meskó, “Prompt Engineering as an Important Emerging Skill for Medical Professionals: Tutorial,” J. Med. Internet Res., vol. 25, p. e50638, Oct. 2023, doi: https://doi.org/10.2196/50638.
[4] M. T. Younis, N. M. Hussien, Y. M. Mohialden, K. Raisian, P. Singh, and K. Joshi, “Enhancement of ChatGPT using API Wrappers Techniques,” Al-Mustansiriyah J. Sci., vol. 34, no. 2, pp. 82–86, 2023, doi: https://doi.org/10.23851/mjs.v34i2.1350.
[5] M. Haqi Al-Tai, B. M. Nema, and A. Al-Sherbaz, “Deep Learning for Fake News Detection: Literature Review,” Al-Mustansiriyah J. Sci., vol. 34, no. 2, pp. 70–81, Jun. 2023, doi: https://doi.org/10.23851/mjs.v34i2.1292.
[6] Y. Yao, J. Duan, K. Xu, Y. Cai, E. Sun, and Y. Zhang, “A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly.” arXiv, Dec. 04, 2023, doi: https://doi.org/10.48550/arXiv.2303.11717.
[7] J. Yang et al., “Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.” arXiv, Apr. 27, 2023, doi: https://doi.org/10.1016/j.cose.2023.103424.
[8] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, “Large language models in medicine,” Nat. Med., vol. 29, no. 8, Art. no. 8, Aug. 2023, doi: https://doi.org/10.1038/s41591-023-02448-8.
[9] S. E. Spatharioti, D. M. Rothschild, D. G. Goldstein, and J. M. Hofman, “Comparing Traditional and LLM-based Search for Consumer Choice: A Randomized Experiment.” arXiv, Nov. 08, 2023, doi: https://doi.org/10.48550/arXiv.2307.03744.
[10] Jain, R., Gervasoni, N., Ndhlovu, M., & Rawat, S. (2023). “A Code Centric Evaluation of C/C++ Vulnerability Datasets for Deep Learning Based Vulnerability Detection Techniques.” In Proceedings of the 16th

Innovations in Software Engineering Conference (ISEC '23). ACM, New York, NY, USA (Article 6, pp. 1–10). https://doi.org/10.1145/3578527.3578530.
[11] S. Wu et al., “BloombergGPT: A Large Language Model for Finance.” arXiv, May 09, 2023. Accessed: Dec. 05, 2023, doi: https://doi.org/10.48550/arXiv.2303.17564.
[12] D. Baidoo-Anu and L. Owusu Ansah, "Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning," SSRN Electron. J., 2023, doi: https://doi.org/10.2139/ssrn.4337484.
[13] A. B. Mbakwe, I. Lourentzou, L. A. Celi, O. J. Mechanic, and A. Dagan, “ChatGPT passing USMLE shines a spotlight on the flaws of medical education,” PLOS Digit. Health, vol. 2, no. 2, p. e0000205, Feb. 2023, doi: https://doi.org/10.1371/journal.pdig.0000205.
[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv, May 24, 2019, doi: https://doi.org/10.48550/arXiv.1810.04805.
[15] C. Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” arXiv, Sep. 19, 2023, doi: https://doi.org/10.48550/arXiv.1810.04805.
[16] T. B. Brown et al., “Language Models are Few-Shot Learners.” arXiv, Jul. 22, 2020. doi: https://doi.org/10.48550/arXiv.2005.14165.
[17] Scao, Teven Le et al., “BLOOM: A 176B-Parameter Open-Access Multilingual Language Model,” Cornell Univ. - ArXiv, Nov. 2022, doi: https://doi.org/10.48550/arxiv.2211.05100.
[18] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.” arXiv, Jul. 28, 2021, doi: https://doi.org/10.48550/arXiv.2107.13586
[19] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large Language Models are Zero-Shot Reasoners.” arXiv, Jan. 29, 2023, doi: https://doi.org/10.48550/arXiv.2205.11916
[20] F. Petroni et al., “Language Models as Knowledge Bases?,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics, 2019, pp. 2463–2473, doi: https://doi.org/10.18653/v1/D19-1250.
[21] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, “A Unified MRC Framework for Named Entity Recognition,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online: Association for Computational Linguistics, 2020, pp. 5849–5859, doi: https://doi.org/10.18653/v1/2020.acl-main.519.
[22] M. Shanahan, K. McDonell, and L. Reynolds, “Role-Play with Large Language Models.” arXiv, May 25, 2023, doi: https://doi.org/10.48550/arXiv.2305.16367.
[23] H. Zhou, A. Nova, H. Larochelle, A. Courville, B. Neyshabur, and H. Sedghi, “Teaching Algorithmic Reasoning via In-context Learning.” arXiv, Nov. 15, 2022, doi: https://doi.org/10.48550/arXiv.2211.09066.
[24] S. Arora et al., “Ask Me Anything: A simple strategy for prompting language models.” arXiv, Nov. 19, 2022, doi: https://doi.org/10.48550/arXiv.2210.02441.
[25] J.-W. Bullée and M. Junger, “Social Engineering,” in The Palgrave Handbook of International Cybercrime and Cyberdeviance, T. J. Holt and A. M. Bossler, Eds., Cham: Springer International Publishing, 2020, pp. 849–875. doi: https://doi.org/10.1007/978-3-319-78440-3_38.
[26] F. Mouton, L. Leenen, and H. S. Venter, “Social engineering attack examples, templates and scenarios,” Comput. Secur., vol. 59, pp. 186–209, Jun. 2016, doi: https://doi.org/10.1016/j.cose.2016.03.004.
[27] S. Gupta, A. Singhal, and A. Kapoor, “A literature survey on social engineering attacks: Phishing attack,” in 2016 International Conference on Computing, Communication and Automation (ICCCA), Apr. 2016, pp. 537–540. doi: https://doi.org/10.1109/CCAA.2016.7813778.
[28] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, “Advanced social engineering attacks,” J. Inf. Secur. Appl., vol. 22, pp. 113–122, Jun. 2015, doi: 10.1016/j.jisa.2014.09.005.
[29] L. Xiangyu, L. Qiuyang, and S. Chandel, “Social Engineering and Insider Threats,” in 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Oct. 2017, pp. 25–34. doi: https://doi.org/10.1109/CyberC.2017.91.
[30] F. Salahdine and N. Kaabouch, “Social Engineering Attacks: A Survey,” Future Internet, vol. 11, no. 4, Art. no. 4, Apr. 2019, doi: https://doi.org/10.3390/fi11040089.
[31] M. Scanlon, F. Breitinger, C. Hargreaves, J.-N. Hilgert, and J. Sheppard, “ChatGPT for digital forensic investigation: The good, the bad, and the unknown,” Forensic Sci. Int. Digit. Investig., vol. 46, p. 301609, Oct. 2023, doi: https://doi.org/10.1016/j.fsidi.2023.301609.
[32] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, “CAMEL: Communicative Agents for ‘Mind’ Exploration of Large Language Model Society.” arXiv, Nov. 02, 2023, doi: https://doi.org/10.48550/arXiv.2303.17760.
[33] Y. Zhou et al., “Large Language Models Are Human-Level Prompt Engineers.” arXiv, Mar. 10, 2023, doi: https://doi.org/10.48550/arXiv.2211.01910.

[34] G. Mialon et al., “Augmented Language Models: a Survey.” arXiv, Feb. 15, 2023, doi: https://doi.org/10.48550/arXiv.2302.07842.
[35] J. White et al., “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT.” arXiv, Feb. 21, 2023. Accessed: Nov. 17, 2023. [Online], doi: https://doi.org/10.48550/arXiv.2302.11382.
[36] T. L. Scao and A. M. Rush, “How Many Data Points is a Prompt Worth?” arXiv, Apr. 06, 2021, doi: https://doi.org/10.48550/arXiv.2103.08493.
[37] S. H. Bach et al., “PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts.” arXiv, Mar. 29, 2022. Accessed: Nov. 25, 2023. [Online], doi: https://doi.org/10.48550/arXiv.2202.01279.
[38] J. Wei et al., “Emergent Abilities of Large Language Models.” arXiv, Oct. 26, 2022, doi: https://doi.org/10.48550/arXiv.2206.07682.
[39] A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision.” arXiv, Feb. 26, 2021. Accessed: Dec. 06, 2023. [Online], doi: https://doi.org/10.48550/arXiv.2103.00020
[40] J. Oppenlaender, “A taxonomy of prompt modifiers for text-to-image generation,” Behav. Inf. Technol., vol. 0, no. 0, pp. 1–14, 2023, doi: https://doi.org/10.1080/0144929X.2023.2286532.
[41] J. Wei et al., “Finetuned Language Models Are Zero-Shot Learners.” arXiv, Feb. 08, 2022. doi: https://doi.org/10.48550/arXiv.2109.01652.
[42] H. Xu et al., “ZeroPrompt: Scaling Prompt-Based Pretraining to 1,000 Tasks Improves Zero-Shot Generalization.” arXiv, Oct. 30, 2022, doi: https://doi.org/10.48550/arXiv.2201.06910.
[43] A. I. Nezer, “Nezer10/Prompt-Engineering.” Dec. 06, 2023. Accessed: Dec. 06, 2023. [Online]. Available: https://github.com/Nezer10/Prompt-Engineering
[44] R. L. Logan IV, I. Balažević, E. Wallace, F. Petroni, S. Singh, and S. Riedel, “Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models.” arXiv, Jul. 01, 2021, doi: https://doi.org/10.48550/arXiv.2106.13353.
[45] J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” arXiv, Jan. 10, 2023, doi: https://doi.org/10.48550/arXiv.2201.11903
[46] X. Wang et al., “Self-Consistency Improves Chain of Thought Reasoning in Language Models.” arXiv, Mar. 07, 2023, doi: https://doi.org/10.48550/arXiv.2203.11171.
[47] M. Shanahan, K. McDonell, and L. Reynolds, “Role-Play with Large Language Models.” arXiv, May 25, 2023, doi: https://doi.org/10.48550/arXiv.2305.16367.
[48] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, “CAMEL: Communicative Agents for ‘Mind’ Exploration of Large Language Model Society.” arXiv, Nov. 02, 2023, doi: https://doi.org/10.48550/arXiv.2303.17760.
[49] S. K. K. Santu and D. Feng, “TELeR: A General Taxonomy of LLM Prompts for Benchmarking Complex Tasks.” arXiv, Oct. 24, 2023, doi: https://doi.org/10.48550/arXiv.2305.11430.
[50] R. Logan Iv, I. Balazevic, E. Wallace, F. Petroni, S. Singh, and S. Riedel, “Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models,” in Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 2824–2835. doi: https://doi.org/10.18653/v1/2022.findings-acl.222.
[51] A. H. Ali, M. G. Yaseen, M. Aljanabi, S. A. Abed, and C. Gpt, “Transfer Learning: A New Promising Techniques,” Mesopotamian J. Big Data, pp. 29–30, Feb. 2023, doi: https://doi.org/10.58496/MJBD/2023/004