Six Thinking Chatbots: A Creativity Technique deployed via a Large Language Model*

Alexander Rachmann[1]

[1]CBS International Business School, Hammer Landstraße 89, 41460 Neuss, Germany

Abstract
AI supported Creativity is a hot debated topic, due to the rise of LLMs. This paper explores a prototypical implementation of the classic Six Thinking Hats method via a LLM, using the OpenAI API. The results suggest that such AI support is useful in a creative process; but will not replace the human creativity.

Keywords
ai supported creativity, six thinking hats, large language model,

1. Introduction

Large Language Models (LLMs, singular: LLM), generative mathematical models of the statistical distribution of tokens in vast public corpus of human generated texts[1], are very prominent due to the success of ChatGPT[1].

The current Gartner Hype Cycle sees LLMs at their zenith [2]. Some practical disadvantages and weaknesses of the technologies are already known, such as hallucinations [3] or copyright violations [4]. It is currently unclear when and to what extent the disillusionment will occur and at what level productivity will level out. What is clear, however, is that this technology is predicted to have a major impact on creative work and knowledge work [5], [6], [7]. One research line towards this direction is ai driven creativity / ai driven inspiration / ai powered creativity [5], [8], [9], [10].

This paper describes a prototypical implementation of the Six Thinking Hats method via a LLM (*Six Thinking Chatbots*). It shows what a combination of creativity technique and a large language model might look like. Section 2 takes on related works. Section 3 describes the design assumptions, the program flow and according prompts. In section 4 the results of the prototype are evaluated. Section 5 closes this paper.

In: D. Mendez, A. Moreira, J. Horkoff, T. Weyer, M. Daneva, M. Unterkalmsteiner, S. Bühne, J. Hehn, B. Penzenstadler, N. Condori-Fernández, O. Dieste, R. Guizzardi, K. M. Habibullah, A. Perini, A. Susi, S. Abualhaija, C. Arora, D. Dell'Anna, A. Ferrari, S. Ghanavati, F. Dalpiaz, J. Steghöfer, A. Rachmann, J. Gulden, A. Müller, M. Beck, D. Birkmeier, A. Herrmann, P. Mennig, K. Schneider. Joint Proceedings of REFSQ-2024 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2024. Winterthur, Switzerland, April 8, 2024.

a.rachmann@cbs.de (A. Rachmann)
https://cbs.de/ (A. Rachmann)
0000-0001-8217-7149 (A. Rachmann)

[1]https://openai.com/chatgpt

2. Related works

Related work from the view of requirements engineering such as [11] describe the usage of LLM in each major stage of requirements engineering according to [12]. For the elicitation phase, a single-shot prompt is used. This is comparable to the webbrowser prompt in section 4.2 of this paper. This point of view to this topic is not well explored; first experiments such as [13] are being carried out.

Related work from the view of LLM usage is in full bloom. In the ChatGPT hype, the development of prompting strategies such as [14], [15], [16], [17] provide standardized approaches to prompting. These approaches take on the technical usage of prompting, such as the "write clear instructions", "provide reference texts" [17]. Though these are very useful, these approaches do often not tackle complex situations of human conversations.

The main new idea of this paper is the simulation of a human workshop, using the LLM in multiple, programmed steps.

3. Design

3.1. Design Considerations for a programmed Six Thinking Hats workshop

The Six Thinking Hats methodology is well known in Requirements Engineering. Although it entails several ways to use the different hats[2], this paper only explores one application i.e. the identification of one alternative from a group of three.

A practical workshop needs an introduction by the moderator (blue hat). Due to the OpenAI API, this type of introduction is not necessary or is covered by providing appropriate information before the prompt. Before the actual prompt, a system content is given that contains the content of an introduction to a classic workshop. The order of colors is white, green, yellow, black, red and blue.

To limit the necessary computing power, the contribution from each hat is limited to three bullet points, without explicit limit of words or characters.

3.2. Program flow

Python[3] was used as the implementation environment. A simple object-oriented design of the prototype was chosen. The source code is available at https://GitHub.com/rachmann-alexander/sixthinkingchatbots.

ChatGPT-4 was used as the LLM. In general, ChatGPT-4 is considered the market leader and is therefore suitable as an implementation tool for the prototype. It is unclear whether other LLMs would show a substantially different result. Choosing ChatGPT entails the use of the corresponding API, which in turn entails certain prompt forms.

Six Thinking Chatbots follows this flow:

[2]E.g. generating initial ideas (hat sequence: blue, white, green, blue), identifying solutions (blue, white, green, yellow, black, red, blue), etc. See also [18]).

[3]In the opinion of the author, one could have chosen for almost any other modern programming language, as no exclusive Python-specific capabilities were used.

1. The user delivers a problem statement.
2. The messages for white, green, yellow, black, and red are computed. The messages to each hat follow the structure:
   - System Content gives the LLM the context to interpret the prompt. System Content is a necessary parameter to the ChatGPT API.
   - User content, also a necessary parameter to the ChatGPT API, is subdivided in two entities:

     Problem Statement, a description of the topic to-be-discussed by the hats.

     Task, a description of what the hat should do.

   The messages are submitted to ChatGPT API and responses are received.
3. Compute the prompt for blue hat; submit the prompt to ChatGPT API. The prompt to blue follows the same structure as above, but incorporate the responses to the other hats. The response for blue is received.
4. All prompts and responses are exported to a Markdown-formatted document.

3.3. Prompts to the hats

OpenAI expects the user to give context to a certain prompt, which is given in the system content. This context gives the LLM a better understanding of what the task should be. Each hat receives a different system content, depending on its nature as defined in Six Thinking Hats Methodology. Table 1 lists the system content of each hat. Each prompts of all hats, except for blue Hat, contain a problem statement and a generic task. Table 2 gives the content for the white hat, table 3 for the blue hat.

The blue hat owns the moderator role and has therefore the task to summarize the outputs of the other hats. The prompt to the blue hat is therefore more complex. At first, the content is given, then the task itself. After that all other contributions are given. Table 4 gives an example for a response to the blue hat.

Table 1
System Contents of the hats. All system contents start with “You are a participant of a workshop”. The content in the table complete the prefix.

Hat	System content
White hat	Your decision making is fact based and your argumentation is very structured.
Green hat	Your decision making is based on the possibilities of innovations.
Yellow hat	You are an optimist.
Black hat	You are a pessimist.
Red hat	Your decision making is emotion based.
Blue hat	You are the moderator of a workshop, using the Six Thinking Hats methodology. You are neutral to the contributions from the participants and value each contribution.

Table 2
Example of a prompt, structured by system content (example of the White Hat), problem statement (example according to "Sales Channel" as used in the prototype) and task (generic for all Hats except Blue Hat).

System content: You are a participant of a workshop. Your decision making is fact based and your argumentation is very structured. **Prompt (Problem statement)**: The goal of the workshop is to evaluate three alternatives to a problem. The problem is: The sales process on our companies website does not work very well. Management wants us to change the sales process to one of the three alternatives: First, an old-school chatbot with a beforehand written structure. Second, a chatbot using a large language model, without a pre-defined structure. Third, an old-school contact form. **Prompt (Task)**: Describe your point of view to this problem, using only three bullet points.

Table 3
Example of the prompt to blue hat. The problem statement is the same as in the prompts to the other hats and therefore emitted ("...") in the table. The prompt contain the contribution from all other hats; this is quite long and therefore also emitted in the table. All contents are available at the GitHub Repository.

System content: You are the moderator of a workshop, using the Six Thinking Hats methodology. You are neutral to the contributions from the participants and value each contribution. **Prompt (Task)**: Summarize the contributions of the participants, as listed beneath. Think about the problem statement and decide, based on the contributions of the participants, which alternative should be chosen. **Prompt (Problem statement)**: The problem statement of the workshop is as follows: ... **Prompt (Contribution White)**: Contribution from a fact-oriented participant: ... **Prompt (Other Contributions)**: Contribution from ...hat: ...

Table 4
Example of the response from blue hat.

Response from the LLM to the Blue Hat Prompt
After analyzing all of these contributions, it appears that the majority lean towards investing in a large language model chatbot for the company. Those in favor of this option highlighted several benefits including its adaptability to changing customer needs, the potential for learning and improvement over time, and the ability to provide tailored, efficient service on the fly. They also lauded its innovative approach and the potential for the drastically improved client interaction it could offer. It's recommended to consider the concerns raised by the pessimist too, ensuring to mitigate potential issues like consistency, the maintenance of brand tone, and avoiding technical glitches. Availability of resources, cost implications and consideration for the long-term impact are also factors that should be taken into account when implementing a large language model chatbot solution. The participants showed less enthusiasm for the old-school chatbot and the contact form, identifying limitations like lack of flexibility and evolution, impersonal service, slow turnaround, and inability to handle complex inquiries. These options could result in a process that may feel outdated and inefficient, potentially deterring customers rather than facilitating sales and improving the customer experience. Therefore, based on these contributions, the large language model chatbot stands out as the most promising alternative.

4. Evaluation

4.1. Evaluation question and methods

One can evaluate the Six Thinking Chatbots in terms of:

- Q1: Would a regular prompt deliver the same results with less effort?
- Q2: Are the arguments by the colored hats disjoint? Did the hats take different position?
- Q3: Did the blue hat summarize the contributions from the other chatbots correctly?
- Q4: How “creative” is the script? How much do the responses differ from each other, if the script runs several times?
- Q5: Are the contributions valid, do they represent the state of the art or are they comparable to a workshop with people?

To answer Q1 and Q4, one needs to generate material for comparison. As comparison for Q1, the StrategyQA-Prompting style was chosen [19]. The prompt is structured in a context-part (see table 5), which informs the LLM about the workshop. The second part is the task itself. The third part defines the output format[4]. This last part was not necessary in Six Thinking Chatbots, because the export is handled by the script.

Table 5
Prompt to generate comparison material. This prompt is not deployed via software, so that the code repository is not equipped with code that will only be executed once. All contents are available at the GitHub Repository. The Web browser interface of ChatGPT was used execute this prompt.

The goal of the workshop is to evaluate three alternatives to a problem. The problem is: The sales process on our companies website does not work very well. Management wants us to change the sales process to one of the three alternatives: First, an old-school chatbot with a beforehand written structure. Second, a chatbot using a large language model, without a pre-defined structure. Third, an old-school contact form. Evaluate which solution one would prefer. Give at most 15 arguments. After listing the arguments, write a conclusion including a selection on a solution. Use markdown syntax.

The response to this prompt is structured along the three alternatives (table 6); each alternative is structured in a pro- and contra-section with each three to five arguments.

To answer Q4, one needs to compute several results. After developing the Six Thinking Chatbots script (and thereby generating several preliminary results), five results were computed. All five results are available at GitHub. Only one Text (29.12.2023, 13.06.md in the GitHub Repo) was used a standard analysis material in tables 7 and 8.

[4]The prompt and its response is available in GitHub, exactly at https://GitHub.com/rachmann-alexander/sixthinkingchatbots/blob/main/opt/StrategyQA

Table 6
Abbreviated Response to prompt of table 5, markdown syntax is emitted, focus on LLM

Response
The current sales process on the company's website is under performing, and management is considering three alternatives to improve it: 1. Old-school chatbot with Predefined Structure 2. chatbot using a Large Language Model 3. Old-School Contact Form Evaluation Criteria: The evaluation will consider various factors such as user experience, cost, implementation time, maintenance, and effectiveness among others. ... chatbot using a Large Language Model - Pros: 1. Flexibility: Can handle a wide range of queries with natural, conversational responses. 2. Scalability: Can improve over time with more data and interactions. ... - Cons: 1. Cost: Generally more expensive to implement and maintain. 2. Complexity: Requires more sophisticated development and maintenance. ...

4.2. Semantic cohesion of the Responses (Q1-Q3)

To answer Q1 and Q2, table 7, page 8, shows the arguments made by the chatbots. One sees, that the arguments are mostly disjoint; this holds within the results of the colored hats as well as between the Six Thinking Chatbots result and the Web browser result. The chatbots did contribute different aspects and therefore fulfilled the expected outcome. To answer Q3, see table 8. Blue hat voted for the AI chatbot, which represents the majority of the votes of the other chatbots.

Table 7

Comparison of Arguments between Six Thinking Chatbots and Web browser Prompt. Legend: white, green, yellow, black, red, Webbrowser prompt. A X indicates which chatbots made an argument.

Argument	W	G	Y	B	R	WP
Factual Analysis	X					
Objective Comparison	X					
Long-Term Impact and Scalability	X					
Innovation & AI-Driven		X	X			
Enhanced User Experience		X			X	X
Limitation of Alternatives		X				
Adaptability / Limited Flexibility of old school chatbot			X		X	X
Potential for Growth / Scalability			X			X
Non-comprehensiveness				X		
Potential for confusion				X		
Outdated				X	X	X
Consistency / Reliability						X
Costs						X
Speed						X
Ease of Implementation of old school chatbot / form						X
Advanced Capabilities of LLM						X
Complexity of LLM						X
Data Privacy of LLM						X
Data Collection of old school form						X
Limited information via old school form						X

Table 8

Comparison of Conclusions by the chatbots. Legend as in table 7, plus blue. The "+"-sign indicates an pro-argument for the proposed solution, a "-" indicates a contra-argument for the solution and a blank indicates no clear position of the chatbot.

Alternative	W	G	Y	B	R	Bl	WP
Old school chatbot		-	-	-		-	
AI chatbot		+	+	-	+	+	+
Old school contact form		-		-	-	-	

4.3. Similarity between exports (Q4)

To answer Q4, five texts were generated. To compute how similar these texts are, one can use the shingle algorithm to compute the jaccard similarity. The jaccard similarity describes how similar two documents are to each other. A similarity of 1 means the two documents are identical, a 0 means the documents do not have an intersection. There is no definition on how the jaccard similarity must be in order to identify two documents as the same or nearly same. [20] The average similarity of different texts is 0,608. Considering, that there is no common understanding of how similar this is, one may conclude, that the prompts are always distinctively different, but always alike.

4.4. Relation to a real world workshop (Q5)

The answer to question 4 is not as easy to compute as the other answers. First, the quality of the outcomes of a workshop depend highly on the participants, their motivation, the moderator, the conduction of the workshop, etc. If a person workshop would provide different or better outcomes, is not easy to determine. However, the arguments made by the chatbots are mostly common knowledge and practice. It would be no surprise, if these outcomes were elaborated by non-experts.

The chosen application scenario was generic. It is also unclear whether the same results would arise with very specific use cases or domain experts.

5. Conclusion

This paper introduces a prototypical implementation of the Six Thinking Hats method, called Six Thinking Chatbots. The source code is available at GitHub, as are all evaluation cases. The experiment showed that the Six Thinking Chatbots deliver distinctively different results than regular prompting (Q1). The different hats play out their specific role (Q2 and Q3). The script delivers always distinctively different outputs (Q4). It is unclear, if the responses of the chatbots are different / in what way they are different to responses of humans (Q5). However, the author would use the script only to support human creativity (such as a preparation to a Six Thinking Hats workshop), not as a substitution for human creativity.

References

[1] M. Shanahan, Talking about large language models, Communications of the ACM 67 (2024) 68–79.

[2] L. Perri, What's new in artificial intelligence from the 2023 gartner hype cycle, 2023. URL: https://www.gartner.com/en/articles/what-s-new-in-artificial-intelligence-from-the-2023-gartner-hype-cycle.

[3] H. Ye, T. Liu, A. Zhang, W. Hua, W. Jia, Cognitive mirage: A review of hallucinations in large language models, 2023. arXiv:2309.06794.

[4] A. Karamolegkou, J. Li, L. Zhou, A. Søgaard, Copyright violations and large language models, arXiv preprint arXiv:2310.13771 (2023).

[5] N. Jia, X. Luo, Z. Fang, C. Liao, When and how artificial intelligence augments employee creativity, Academy of Management Journal (2023).
[6] E. Creely, Conceiving creativity and learning in a world of artificial intelligence: A thinking model, in: Creative provocations: Speculations on the future of creativity, technology & learning, Springer, 2023, pp. 35–50.
[7] D. H. Cropley, K. E. Medeiros, A. Damadzic, The intersection of human and artificial creativity, in: Creative provocations: Speculations on the future of creativity, technology & learning, Springer, 2023, pp. 19–34.
[8] K. Liu, S. Reddivari, K. Reddivari, Artificial intelligence in software requirements engineering: State-of-the-art, in: 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI), IEEE, 2022, pp. 106–111.
[9] J. J. Y. Chung, Artistic user expressions in ai-powered creativity support tools, in: Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, 2022, pp. 1–4.
[10] Y. Jeon, S. Jin, P. C. Shih, K. Han, Fashionq: an ai-driven creativity support tool for facilitating ideation in fashion design, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–18.
[11] C. Arora, J. Grundy, M. Abdelrazek, Advancing requirements engineering through generative ai: Assessing the role of llms, arXiv preprint arXiv:2310.13976 (2023).
[12] A. v. Lamsweerde, Systematic requirements engineering-from system goals to uml models to software specifications, 2008.
[13] A. Brand, Linkedin post: Was wurde denn nun eigentlich aus dem vortrag in leipzig zu meinem selbstversuch?, 2023. https://www.linkedin.com/posts/anja-brand_selbstversuch-covid-chatgpt-activity-7136342918503489537-W0Jm/.
[14] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, Q. Yang, Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1–21.
[15] Z. Yu, L. He, Z. Wu, X. Dai, J. Chen, Towards better chain-of-thought prompting strategies: A survey, arXiv preprint arXiv:2310.04959 (2023).
[16] J. Kim, S. Park, K. Jeong, S. Lee, S. H. Han, J. Lee, P. Kang, Which is better? exploring prompting strategy for llm-based metrics, arXiv preprint arXiv:2311.03754 (2023).
[17] OpenAI, Prompt engineering, 2024. https://platform.openai.com/docs/guides/prompt-engineering.
[18] E. De Bono, Six Thinking Hats: The multi-million bestselling guide to running better meetings and making faster decisions, Penguin uk, 2017.
[19] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, J. Berant, Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies, Transactions of the Association for Computational Linguistics 9 (2021) 346–361.
[20] C. C. Aggarwal, Information Retrieval and Search Engines, Springer International Publishing, Cham, 2022, pp. 257–302.