Artificial Intelligence, Social Computing and Wearable Technologies, Vol. 113, 2023, 25–37
https://doi.org/10.54941/ahfe1004176

TAUCHI-GPT: Leveraging GPT-4 to Create a Multimodal Open-Source Research AI Tool

Ahmed Farooq, Jari Kangas, and Roope Raisamo

Tampere Unit of Computer Human Interaction, Tampere University, Tampere, 33014, Finland

ABSTRACT

In the last few year advances in deep learning and artificial intelligence have made it possible to generate high-quality text, audio, and visual content automatically for a wide range of application areas including research and education. However, designing and customizing an effective R&D tool capable of providing necessary tool-specific output, and breaking down complex research tasks requires a great deal of expertise and effort, and is often a time-consuming and expensive process. Using existing Generative Pre-trained Transformers (GPT) and foundational models, it is now possible to leverage the Large Language Model GPTs already trained on specific datasets to be effective in common research and development workflow. In this paper, we develop and test a customized version of autonomous pretrained generative transformer which is an experimental open-source project built on top of GPT-4 language model that chains together LLM "thoughts", to autonomously achieve and regress towards specifics goals. Our implementation, referred to as TAUCHI-GPT, which uses an automated approach to text generation that leverages deep learning and output reflection to create high-quality text, visual and auditory output, achieve common research and development tasks. TAUCHI-GPT is based on the GPT-4 architecture and connects to Stable Diffusion and ElevenLabs to input and output complex multimodal streams through chain prompting. Moreover, using the Google Search API, TAUCHI-GPT can also scrap online repositories to understand, learn and deconstruct complex research tasks, identify relevant information, and plan appropriate courses of action by implementing a chain of thought (CoT).

Keywords: Artificial intelligence, Generative pre-trained transformers, Autonomous agents, Large language models (LLMS), ChatGPT, Chain of though (COT), Tree of thought (TOT), AI alignment, Human computer interaction, Research and development, Foundation AI models, Responsible AI (RAI)

INTRODUCTION

Artificial intelligence (AI) has recently witnessed remarkable advancements, with ChatGPT [OpenAI, 2023] emerging as a standout due to its exceptional capabilities in reasoning, comprehension, and interactive abilities [Wu et al., 2023]. The capacity to perform novel tasks based on instructions represents a crucial step towards achieving artificial general intelligence. Consequently, the impressive potential of large language models (LLMs) has sparked a

multitude of research areas, including in-context learning [Ram et al., 2023; Xie et al., 2021], chain-of-thought prompting [Pilault et al., 2023; Wei et al., 2022b], retrieve and read [Izacard and Grave, 2020; Zhang et al., 2021, 2022], and GPT-based intelligent systems [Zheng et al., 2023]. These domains aim to explore the vast potential of LLMs and offer abundant opportunities for constructing sophisticated AI systems.

LLMs, such as GPT-4 [Brown et al., 2020; OpenAI, 2023], LLaMA [Touvron et al., 2023], Flan-T5 [Chung et al., 2022], and PaLM [Chowdhery et al., 2022], have demonstrated a profound understanding of natural language and the ability to generate coherent, contextually appropriate responses. This progress has opened-up new avenues for challenging tasks that involve diverse data domains, including image and text processing, as well as the incorporation of domain-specific knowledge. In this context, LLMs play a crucial role, as their ability to comprehend and produce natural language empowers AI systems to gain a better understanding and address a wide range of challenges.

This paper introduces TAUCHI-GPT, an extension of the Automatic Machine Learning (AutoML) paradigm proposed by Zhang et al., (2023) and Auto-GPT (2023) opensource project that leverages LLMs to automate model training using pretrained datasets, user inputs and descriptions. The LLMs serve as an automatic training system, establishing connections with versatile models and processing inputs. Our primary objective was to utilize language as a universal interface and prompt for LLMs to engage with users. By incorporating both data and model descriptions into prompts, LLMs can effectively manage AI models for tasks such as data processing, model architecture design, and hyperparameter tuning. These models can also be invoked as required to address AI tasks and generate predicted training logs. However, integrating multiple AI models into LLMs necessitates a significant number of high-quality model descriptions (Mitchell et al., 2019) that provide well-defined descriptions, as well as data cards (Gebru et al., 2021). This approach facilitates the combination of diverse models through a language-based interface, thereby enabling the solution of complex AI tasks. It also enhances the transferability among models and datasets by capturing their underlying similarities. This technique is considerably recent, however, many researchers are exploring its potential in training (Lu et al., 2023a) and optimizing (ToT, Long, 2023 preprint) LLM responses as well as creating targeted models for instruct, chat and storytelling purposes.

RELATED WORK

The pursuit of developing intelligent systems capable of reasoning has long been a central objective in the field of artificial intelligence (Wos et al., 1984; Hayes-Roth et al., 1983; Fagin et al., 2003). Recent advancements in large language models (LLMs) have unlocked new possibilities for machine reasoning, thanks to their emergent properties and in-context learning capabilities (OpenAI and GPT-4; Bubeck et al., 2023; Wei et al., 2023). Notably, researchers have discovered that employing techniques such as chain-of-thought prompting can elicit step-by-step solutions for mathematical and

logical reasoning tasks from LLMs (Drori et al., 2022). Further investigations have explored approaches like sampling multiple solutions and using self-consistency or complexity-based criteria to determine optimal responses (Wang et al., 2023). Experimental evaluations have also been conducted to assess the performance of different prompts (Fu and Peng, 2023). One noteworthy technique, the self-taught reasoner (STaR) (Zelikman et al., 2022), involves an LLM generating reasoning chains and subsequently discarding those that yield incorrect answers. The model is then fine-tuned using the remaining valid reasoning chains.

Reasoning With Large Language Models

However, while these techniques demonstrate promising potential, they often require substantial human involvement. For instance, chain-of-thought prompting necessitates the creation of carefully crafted examples, limiting its scalability. Consequently, researchers have begun exploring the realm of automatic prompt generation. Early explorations in this area include AutoPrompt (Shin et al., 2020), prefix-tuning (Li and Liang 2021), and parameter-efficient prompt tuning (Lester et al., 2021). Recent studies have further intensified focus on this research direction. In a notable investigation (Cobbe et al., 2021), the authors experiment with training verifiers to evaluate whether the solutions provided by an LLM for mathematical problems are logically correct. Effective verification could provide an alternative avenue for prompt evaluation. Another approach, automatic prompt engineer (Zhou et al., 2023), explores a method for selecting the best prompt from a set of model-generated candidates. The study (Shum et al., 2023) proposes a three-phase augment-prune-select methodology. Initially, multiple chain-of-thought candidates are generated, followed by pruning based on the match between derived answers and ground truths. Finally, a policy gradient-based approach is employed to select the optimal combination of rationale chains for chain-of-thought prompting.

Augmenting LLMs With Additional Agents

Recent research has also investigated the augmentation of LLMs with additional agents to enhance their capabilities, an area closely related to our current work. For instance, Auto-GPT (2023) combines GPT-4 with supplementary modules, including an execution agent and a memory unit, enabling the chaining of LLM "thoughts" to autonomously accomplish user-defined goals. PromptPG (Lu et al., 2023a) proposes an approach that employs policy gradient learning to select in-context examples from a limited amount of training data for prompt learning. The PromptPG agent learns to identify optimal in-context examples from a candidate pool, maximizing prediction rewards on provided training examples during interaction with the GPT-3 environment. DEPS (Wang et al., 2023) leverages multi-step reasoning and sub-task error correction to address complex tasks with long-range dependencies. Notably, DEPS offers explanations for errors in sub-tasks within a trial, exhibiting remarkable performance.

ReAct (Yao et al., 2023) utilizes emergent properties present in LLMs, such as traces of verbal reasoning, to enable agents to reason and take action, yielding impressive results on various text-based benchmarks. Building upon ReAct, Reflexion (Shinn et al., 2023) equips agents with dynamic memory and self-reflection capabilities, enhancing their reasoning trace and task-specific action selection. To achieve complete automation, a simple and effective heuristic model was developed by the authors which identified hallucination instances and prevented repetitive action sequences.

While our approach shares some similarities with these approaches using reflection OODA model (Blaha L. M., 2018), we did not incorporate a memory module and additional agents for automatic prompt generation, as introduces by colleagues at Harvard, Theta Labs (Long, 2023 preprint) ToT controller. Their model hypothesizes that ToT system can explicitly allow for backtracking, when necessary and can not only enable the system to recover from mistakes but also potentially expand the solution search space. As this was not our goal with TAUCHI-GPT, we simply utilized different reflection cycles for the output generated by the LLM and had users rate the quality of the responses according to their needs and original prompts.

FOUNDATIONAL MODELS AND TAUCHI-GPT

With the emergence of ChatGPT (OpenAI, 2023), Bard (REF), Claude (REF), and other large language model (LLM)-based chatbots (Lu et al., 2023a) has garnered considerable attention in the field of foundational models worldwide. Foundational models, such as LLMs, are AI models that undergo pretraining on extensive and diverse datasets, enabling their adaptation to a wide range of tasks and substantial enhancements in productivity (Bommasani et al., 2021). Given the ongoing exploration of their potential through various projects, it is widely anticipated that foundational models will serve as the fundamental building blocks for future AI systems.

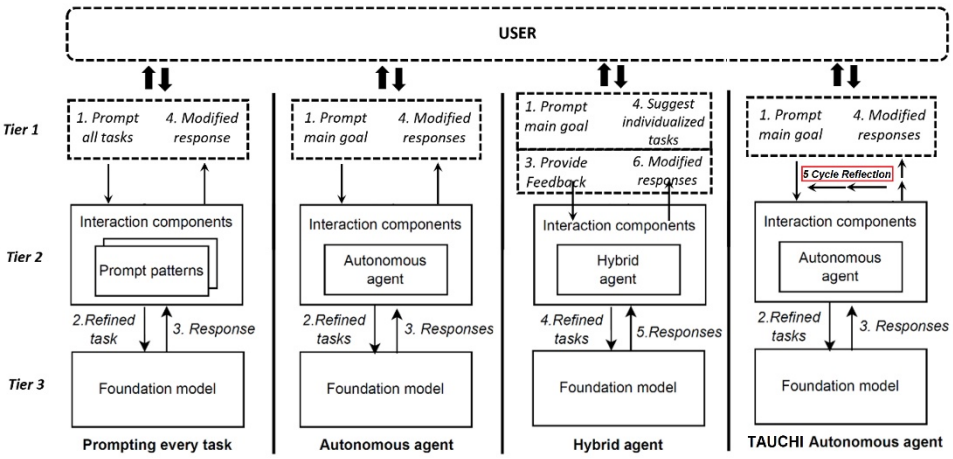

Figure 1: Techniques of hierarchical communication (Lu et al., 2023a) with foundational models and TAUCHI-GPT structure.

Nevertheless, the design of foundation model-based systems is still in its nascent stages and lacks systematic exploration. There exists limited understanding regarding the implications of incorporating foundation models into software architecture. Furthermore, the opaque nature of these models and their rapid advancements have raised significant concerns regarding AI risks (Van-Dis et al., 2023). Consequently, there is a pressing need for concrete solutions in responsible AI (RAI), particularly in the context of foundation model-based systems. The demand is specifically focused on responsible-AI-by-design approaches that address these concerns proactively (Lu et al., 2023b).

Utilizing Foundation Models as Connectors

Within software architecture, software connectors play a pivotal role as fundamental entities for enabling interactions among software components (Mehta et al., 2000). These connectors provide various services, including communication, coordination, conversion, and facilitation. As the user initiates a task, interaction components refine it further to ensure accuracy and address responsible AI (RAI) considerations as discussed by Lu et al., (2023a). The foundation model assumes an architectural function by providing the following connector services to establish connections with other non-AI/AI components within the organization. Essentially, there are four key types of connectors services as explained below:

Communication Connector: Foundation models, such as LLMs, act as communication connectors, enabling the transfer of data between software components. For example, an LLM can receive task text descriptions from users, extract the meaning and intention from the text, and subsequently transfer the extracted task information to other components for further processing. This may involve transmitting the information to an AI model or a robotics system to carry out specific tasks (Shen et al., 2023; Driess et al., 2023).

Coordination Connector: Foundation models facilitate coordination among different software components in their computations. For instance, an LLM can be employed to plan complex tasks or workflows, coordinating the planning, selection, and collaboration of multiple AI models through a text-based interface (Shen et al., 2023). The LLM decomposes the task into subtasks, establishes dependencies, and determines the execution order for these subtasks. Additionally, the LLM requires access to model descriptions (e.g., functionality, architecture, domains) to match tasks with the appropriate models.

Conversion Connector: Foundation models serve as interface adapters, enabling seamless communication between software components that employ different data formats. For instance, an LLM can analyse text task descriptions provided by users and parse them into machine-readable templates (e.g., task ID, type, dependencies, arguments) to facilitate execution by an AI model.

Facilitation Connector: Foundation models can be integrated as facilitation connectors to optimize interactions between components. For example,

an LLM can be employed to maintain chat logs, make decisions regarding running commonly used and time-consuming models locally, manage resource dependencies during task execution, and summarize the task execution process and inference results (Shen et al., 2023).

By leveraging foundation models as connectors, software architecture can harness their capabilities in enabling communication, coordination, conversion, and facilitation among diverse software components.

Communication With Foundation Models

The system predominantly employs a web interface to facilitate communication with the foundation models, where users provide prompts in the form of instructions, questions, or statements. Prompt engineering is a crucial process in shaping the output of foundation models to align with specific requirements [Fig. 1]. While using the default web interface can be inefficient for complex tasks or workflows, necessitating prompts at each step, prompt patterns are commonly utilized to ensure accuracy and address responsible AI (RAI)-related concerns. Several prompt patterns are available, including zero/one/few-shot prompts, retrieval/internet-augmented prompts, chain of thought prompts, think aloud prompts, bot team prompts, negative prompts, and multiple-choice prompts. The selection of prompt patterns should consider various factors such as the system's goal, target users, context, and specific tasks. Each pattern may incur different costs and exhibit varying levels of complexity.

To mitigate the need for extensive prompt engineering and its associated costs, autonomous agents like Auto-GPT[1], BabyAGI[2], and AgentGPT[3] can be employed. With autonomous agents, users only need to provide the overarching goal, and the agent autonomously decomposes it into a set of tasks, leveraging other software components, the internet, and tools to accomplish these tasks automatically. However, relying solely on autonomous agents may introduce accuracy issues, as they may not fully comprehend users' intentions. In contrast, hybrid agents such as GodMode[4] (Shen et al., 2023) and AI chain[5] (Driess et al., 2023) involve users in the loop to confirm plans and provide feedback. This interactive approach enhances both the accuracy and RAI-related properties of task execution.

Design and Modelling TAUCHI-GPT

As discussed, the ability to generate high-quality text, visual, and auditory output that aligns with specific research objectives is of great importance in the field of artificial intelligence. Existing work in autonomous agents can leverage various LLM, such as GPT-4, to parse and restructure complex instruction and relevant tasks. Currently models and approached discussed

[1] https://github.com/Significant-Gravitas/Auto-GPT

[2] https://github.com/yoheinakajima/babyagi

[3] https://github.com/reworkd/AgentGPT

[4] https://godmode.space

[5] https://aichain.online

above have been successfully utilized for simplified tasks i.e., creating, summarizing, and optimizing text and image-based input. However, to address the complexities of research and development tasks, a multimodal approach that combines different modes of input and output is required. In this section, we discuss TAUCHI-GPT, an automated approach to multimodal text generation that extends the capabilities of the GPT-4 architecture to facilitate research and development within an academic setting for a wide range of experienced university researchers.

TAUCHI-GPT builds upon the GPT-4 architecture, incorporating additional modules and functionalities to enable multimodal text generation. The command line prompt system connects with Stable Diffusion and ElevenLabs to handle complex multimodal input and output streams. By leveraging these resources, TAUCHI-GPT can generate text, visual, and auditory output that aligns with specific research goals. To ensure the generation of relevant and accurate responses, TAUCHI-GPT employs the Google Search API to scrape online repositories and gather information related to complex research tasks. This allows the system to understand the task at hand, identify relevant information, and plan appropriate courses of action. By assimilating this information, TAUCHI-GPT can generate multimodal responses that fulfil specific research goals.

TAUCHI-GPT incorporates several techniques to enhance the quality and relevance of its generated output. First, fine-tuning of the GPT-4 model is performed for each module, allowing the system to adapt to specific research domains and improve output coherence. Additionally, custom pre-processing and post-processing steps were implemented to filter input and output, ensuring the generation of high-quality multimodal text. Integration with other multimodal AI tools and the World Wide Web further enriched TAUCHI-GPT's capabilities and enhanced the diversity of its generated output. Finally, techniques for improving and regulating system/module output were also employed, enabling the system to learn and adapt over time. To validate these optimizations, we created two versions of TAUCHI-GPT, one with no reflection cycles and classified Chain of Thought (CoT) called System A; and the other with 4–5 reflection cycles and enhanced Chain of Though (CoT) called System B and had participants evaluate each version separately.

USER CENTRIC TESTING (PILOT)

To validate our approach, we conducted a user-centric evaluation of TAUCHI-GPT, an automated multimodal text generation system for research & development tasks. The goal of the pilot study was to assess its performance and effectiveness in three core classification modules: "Composing Scientific Publications", "Creating SW Systems using Python", and "Designing User Experiments". To achieve this, we recruited 18 university researchers who incorporated TAUCHI-GPT into their daily research tasks over a period of 3 weeks with a min of 40 tasks.

The participants were divided into three groups, with each group assigned to one of the core classification modules. These modules were pretrained and filtered to ensure that the generated output was relevant and aligned with the

specific goals of each module. Each group consisted of both novice and expert researchers, allowing us to gather diverse perspectives on TAUCHI-GPT's performance.

Additionally, each group was provided two versions of TAUCHI-GPT, System A with no reflection cycles and classified Chain of Thought (CoT); and System B with 4–5 reflection cycles and enhanced Chain of Though (CoT). Half of the participants in each group started with System A, while the other half started with System B and 1.5 weeks later switched to the other system.

To evaluate the participants' experiences with the different versions of TAUCHI-GPT, we employed the NASA-TLX and customized questionnaires as well as structured interview after the end of the three-week trial. Both NASA-TLX and customized questionnaire was designed to measure workload and user experience, assessing multiple dimensions such as demand, and usability of novel systems. Additionally, structured interviews were conducted after the three-week trials to gather qualitative feedback and delve into participants' perceptions; and identify any challenges they encountered. Detailed log files were also collected at the end of the testing period for all the sessions and relevant questions were drafted for each participant accordingly.

Results of Pilot Study

Overall, the participants found both versions of TAUCHI-GPT to be more useful and informative compared to their default search engines. On average participants rated System B (with 4–5 reflection cycles and enhanced CoT) to be more reliable and functional for their workflow. However, they reported that "System B" was noticeably slower than "System A". Additionally, some participants informed the command line prompting interface to be clunky and difficult to use.

Looking at the results from NASA-TLX questionnaire we can see that all three groups found "System B" to be better optimized for their workflow compared to System A (Fig. 2). The SW development group rated System B's outputs the highest whereas "Experiment Design" group did not perceive many differences between the two versions of TAUCHI-GPT. This may have been because participants in the Software Development group had a stronger background with computational tools while Experiment Design group mostly consisted of social scientist and Publication / Documentation group was a mixture of both.

Results from the post-trial questions (Fig. 3) also showed that participants preferred the optimized version of TAUCHI-GPT. Nevertheless, both version of TAUCHI-GPT reduced the need for their search engine of choice (Bing and Google). However, the participants felt that the output from TAUCHI-GPT could not be trusted explicitly. Participants were not informed that the Large Language Model behind the system was GPT4 as that may have influenced their responses. Additionally, they found the optimized system to be easier to use and more useful in providing the necessary information. The system was also less likely to generate irrelevant information and participants found they needed to use smaller prompts and make fewer queries.

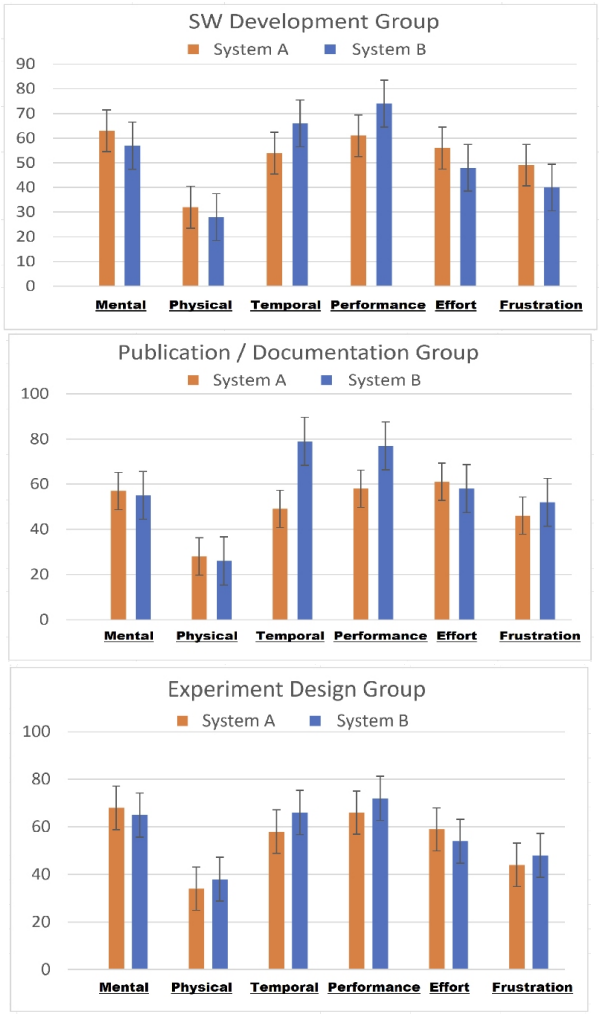

Figure 2: (Top) NASA-TLX scores for SW development group, (middle) for publications and documentation group (bottom) for experiment design group.

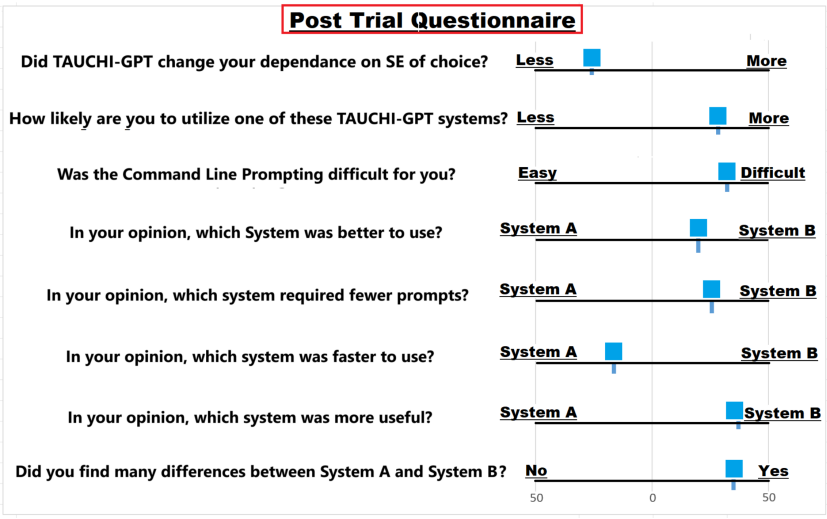

Figure 3: Post-test user experience questionnaire for TAUCHI-GPT.

However, participant showed their frustration at the speed of responses as System B (optimized using CoT and 4–5 reflection cycles) took longer to generate the necessary output. Participants were not told that the reflection cycles were recursive in nature and needed additional computation to yield system responses. And as recorded earlier participants felt the system would have been more productive if it did not entirely utilize Windows command line prompts.

CONCLUSION

This research builds on top of existing work in developing autonomous agent-based LLM interfaces by creating TAUCHI-GPT, an opensource conversation AI leveraging GPT4 and Chain of Thought (CoT) prompting techniques. The study establishes the positive impact of TAUCHI-GPT on research tasks and highlights its potential for enhancing productivity in academic and professional settings. The study also identifies avenues for future research, including exploring the benefits of increased reflection cycles, qualifying use cases, and optimizing the system through the utilization of open-source LLMs and customized AutoML implementations. By continuing to advance and refine Tauchi-GPT, researchers can unlock its full potential in supporting knowledge creation and accelerating research endeavours.

The findings of the user study provide compelling evidence of the positive impact of TAUCHI-GPT on the performance and completion times of the three research tasks. Participants reported significant improvements in their ability to generate high-quality text, visual, and auditory output with minimal human input. They expressed satisfaction with the system's capability to be customized and primed according to their specific needs. The user-friendly nature of TAUCHI-GPT and its high efficiency were also highlighted by the participants, particularly after they became proficient in utilizing the command line interface prompting schema.

Moreover, the inclusion of reflection cycles in the study revealed notable differences in the reliability of the system's output. Participants consistently rated reflection cycles of 4–5 as yielding the highest quality output. This suggests that increasing the number of reflection cycles could potentially yield further improvements. Nonetheless, it is essential to conduct additional research to understand the benefits and potential delays associated with reflection cycles greater than 5, as this could impact workflow efficiency. Further investigation is necessary to explore and qualify the use cases and effectiveness of TAUCHI-GPT. However, the present study demonstrates the feasibility of integrating AI tools into a central interface to optimize research activities and enhance productivity for both experienced and young researchers. Yet, more comprehensive research is needed to delve into the specific contexts and domains where TAUCHI-GPT can deliver the greatest benefits.

Our future work will focus on leveraging open-source large language models (LLMs) and customized AutoML implementations. This approach will

involve utilizing the LangChain model and offline document repositories to fine-tune responses tailored to specific research workflows. By harnessing the vast resources of these models and repositories, TAUCHI-GPT can be further optimized to deliver even more accurate and contextually appropriate outputs. Additionally, efforts will be made to optimize the application and enhance the response time of TAUCHI-GPT using the Tree of Thought (ToT) approach. By incorporating this approach, the system will be able to dynamically explore different reasoning paths and generate more efficient responses, thereby reducing response time and improving overall user experience.

ACKNOWLEDGMENT

This work was partially funded by Tampere Institute of Advances Studies and Tampere Unit of Computer Human Interaction. Authors would also like to acknowledge the Auto-GPT and AutoML opensource projects and development frameworks as this work largely builds on top of the current development in autonomous agents and the associated code base from their GitHub libraries.

REFERENCES

Agent-GPT Opensource Project (last accessed on 31.05.2023) https://github.com/reworkd/AgentGPT.

AIChain Opensource Project (last accessed on 31.05.2023) https://aichain.online.

Auto-GPT An Autonomous GPT4 Generative Pre-trained Transformer, Experiement and Opensource Project (last accessed on 31.05.2023) https://github.com/Significant-Gravitas/Auto-GPT.

BabyAGI Opensource Project (last accessed on 31.05.2023) https://github.com/yoheinakajima/babyagi.

Blaha, L. M., 2018. Interactive OODA processes for operational joint human-machine intelligence. In NATO IST-160 Specialist's Meeting: Big Data and Military Decision Making.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E. and Brynjolfsson, E., 2021. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. Advances in neural information processing systems, 33, pp. 1877–1901.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S. and Nori, H., 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S. and Schuh, P., 2022. Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S. and Webson, A., 2022. Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R. and Hesse, C., 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T. and Huang, W., 2023. Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378.

Fu, Y., Peng, H., Sabharwal, A., Clark, P. and Khot, T., 2022. Complexity-based prompting for multi-step reasoning. arXiv preprint arXiv:2210.00720.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D. and Crawford, K., 2021. Datasheets for datasets. Communications of the ACM, 64(12), pp. 86–92.

GodMode Opensource Project (last accessed on 31.05.2023) https://godmode.space

Hayes-Roth, F., Waterman, D. A. and Lenat, D. B., 1983. Building expert systems. Addison-Wesley Longman Publishing Co., Inc.

Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, Roman Wang, Nikhil Singh, Taylor L. Patti, Jayson Lynch, Avi Shporer, Nakul Verma, Eugene Wu, and Gilbert Strang. (2022) A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. Proceedings of the National Academy of Sciences, 119(32), aug 2022.

Izacard, G. and Grave, E., 2020. Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282.

Lester, B., Al-Rfou, R. and Constant, N., 2021. The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

Li, X. L. and Liang, P., 2021. Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190.

Long, J., 2023. Large Language Model Guided Tree-of-Thought. arXiv preprint arXiv:2305.08291.

Lu, P., Qiu, L., Chang, K. W., Wu, Y. N., Zhu, S. C., Rajpurohit, T., Clark, P. and Kalyan, A., 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. arXiv preprint arXiv:2209.14610.

Lu, Q., Zhu, L., Xu, X. and Whittle, J., 2023. Responsible-AI-by-Design: A Pattern Collection for Designing Responsible AI Systems. IEEE Software.

Lu, Q., Zhu, L., Xu, X., Xing, Z. and Whittle, J., 2023a. A Framework for Designing Foundation Model based Systems. arXiv preprint arXiv:2305.05352.

Mehta, N. R., Medvidovic, N. and Phadke, S., 2000, June. Towards a taxonomy of software connectors. In Proceedings of the 22nd international conference on Software engineering (pp. 178–187).

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D. and Gebru, T., 2019, January. Model cards for model reporting. In Proceedings of the conference on fairness, accountability, and transparency (pp. 220–229).

OpenAI. 2023. Gpt-4 technical report. arXiv.

Parikh, R., 1997. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Reasoning about knowledge. MIT Press, Cambridge, Mass., and London1995, xiii+ 477 pp. The Journal of Symbolic Logic, 62(4), pp. 1484–1487.

Pilault, J., Garcia, X., Bražinskas, A. and Firat, O., 2023. Interactive-Chain-Prompting: Ambiguity Resolution for Crosslingual Conditional Generation with Interaction. arXiv preprint arXiv:2301.10309.

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K. and Shoham, Y., 2023. In-context retrieval-augmented language models. arXiv preprint arXiv:2302.00083.

Shen, Y., Song, K., Tan, X., Li, D., Lu, W. and Zhuang, Y., 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. arXiv preprint arXiv:2303.17580.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E. and Singh, S., 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. arXiv preprint arXiv:2010.15980.

Shinn, N., Labash, B. and Gopinath, A., 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. arXiv preprint arXiv:2303.11366.

Shum, K., Diao, S. and Zhang, T., 2023. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. arXiv preprint arXiv:2302.12822.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. and Rodriguez, A., 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Van Dis, E. A., Bollen, J., Zuidema, W., van Rooij, R. and Bockting, C. L., 2023. ChatGPT: five priorities for research. Nature, 614(7947), pp. 224–226.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E. and Zhou, D., (2022). Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.

Wang, Z., Cai, S., Liu, A., Ma, X. and Liang, Y., 2023. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. arXiv preprint arXiv:2302.01560.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q. and Zhou, D., (2022). Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903.

Wos L., Overbeck R., Lusk E, and Boyle J. (1984). Automated Reasoning: Introduction and Applications. Prentice Hall Professional Technical Reference, originally published 1984.

Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z. and Duan, N., 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671.

Xie, S. M., Raghunathan, A., Liang, P. and Ma, T., 2021. An explanation of in-context learning as implicit bayesian inference. arXiv preprint arXiv:2111.02080.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y., 2022. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.

Zelikman, E., Wu, Y., Mu, J. and Goodman, N., 2022. Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems, 35, pp. 15476–15488.

Zhang, S., Gong, C. and Liu, X., 2022. Passage-Mask: A Learnable Regularization Strategy for Retriever-Reader Models. arXiv preprint arXiv:2211.00915.

Zhang, S., Gong, C., Wu, L., Liu, X. and Zhou, M., (2023). AutoML-GPT: Automatic Machine Learning with GPT. arXiv preprint arXiv:2305.02499.

Zheng, M., Su, X., You, S., Wang, F., Qian, C., Xu, C. and Albanie, S., 2023. Can GPT-4 Perform Neural Architecture Search?. arXiv preprint arXiv:2304.10970.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H. and Ba, J., 2022. Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910.