Paper CM07

LLMs Unleashed: A New Chapter in Clinical Programming?

Karma Tarap, Bristol Myers Squibb, Boudry, Switzerland

ABSTRACT

The emergence of Large Language Models (LLMs) within the Pharmaceutical Industry brings forth a wealth of opportunities alongside notable challenges. This paper sheds light on the potential of LLMs to significantly impact various aspects of the industry, particularly in data analytics and coding endeavors. We present the Analytics Assistant, an innovative web application leveraging the prowess of LLMs, aimed at offering targeted support to clinical programmers, while also discussing potential hurdles like data privacy and computational resources.

INTRODUCTION

In a realm where the intricacy of human language meets the computational prowess of machines, we find ourselves on the brink of significant advancements. Large Language Models (LLMs) like ChatGPT are making waves, promising a transformative impact across various sectors including the pharmaceutical industry. These models offer a glimpse into a future where tasks like code generation and data analysis could become notably streamlined. This paper delves into the world of LLMs, their journey from conception to the current state, and the prospective applications. We introduce the Analytics Assistant, a web application envisaged to tap into the potential of LLMs, aiming to provide a helping hand to analyts. Through this discourse, we seek to offer a realistic insight into how LLMs could not only enhance clinical programming but also significantly alter our interaction with this emerging technology.

BACKGROUND

What are Large Language Models?

Language models, a subset of machine learning models, excel at understanding and generating human language. They're trained on vast amounts of text, leveraging statistical methods to predict subsequent words in a sequence based on the preceding ones. The journey from simple n-gram models and probabilistic context-free grammars to today's sophisticated architectures underscores the remarkable strides made in this field. The landmark paper "Attention Is All You Need" (Vaswani et al., 2017), marked a pivotal point by introducing the Transformer architecture, a significant shift from traditional RNNs and LSTMs. This novel approach to handling sequences through self-attention mechanisms spurred the development of advanced language models like BERT, GPT-2, and eventually GPT-3. The innovations brought about by the Transformer architecture, such as eliminating recurrent layers, enabled parallel processing and effective management of long-term dependencies, paving the way for the creation of models with billions of parameters – the cornerstone of contemporary LLMs.

Rise of the Parameters

The advent of the Transformer architecture ignited the race for developing increasingly expansive language models. Parameter count became a crucial metric, with new models regularly showcasing billions of parameters (*Fig.1*). An augmented parameter count not only enhances a model's contextual understanding but also its capability to generate more nuanced and accurate responses (Chernyavskiy et al., 2021). However, this surge in parameters requires significant computational resources and extends training durations, presenting challenges in model training and deployment (A*nthaswamy, 2023*).

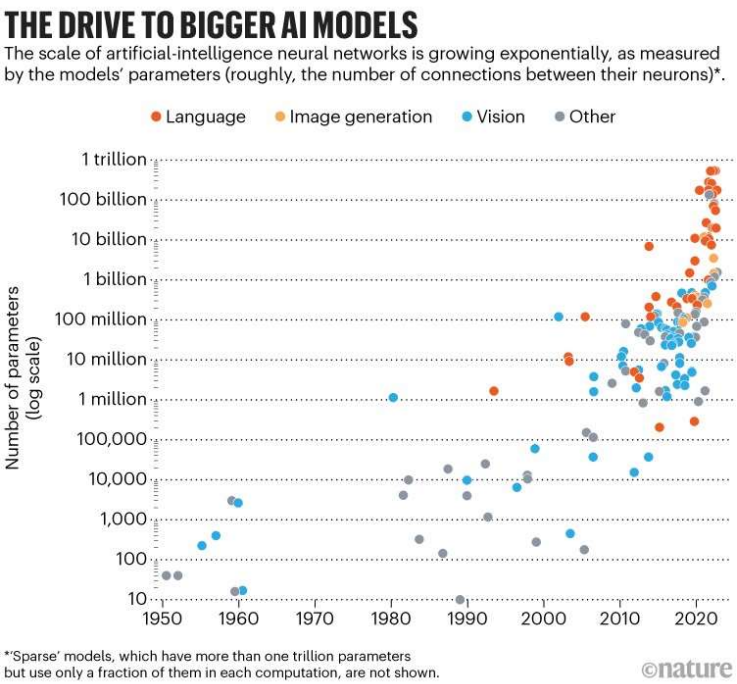

Figure 1. The Drive to Bigger AI Models (Source: Ananthaswamy, 2023).

Public's Imagination: The ChatGPT Phenomenon

The graph below showcases the swift user adoption across various platforms; notably, ChatGPT amassed 100 million users within two months post-release (*Fig.2*). But what fueled this swift adoption?

Among LLMs, ChatGPT uniquely resonated with the public owing to its conversational proficiency, diverse knowledge base, and human-like reasoning. Although built on GPT-3's architecture, ChatGPT underwent a distinct fine-tuning process incorporating conversational data and user feedback. This resulted in a versatile model capable of managing a myriad of tasks and dialogues, thereby appealing to a broader user base.

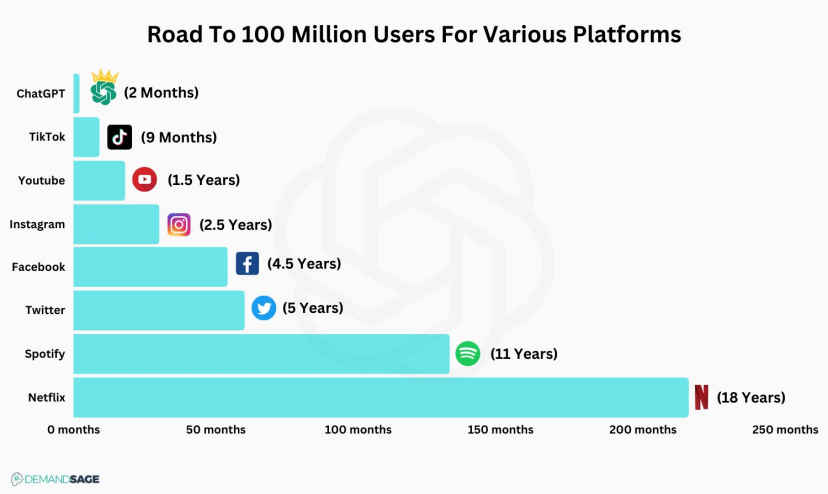

Figure 2. Road to 100 Million Users for Various Platforms (Source: Demand Sage, n.d.)

OPTIMIZING LLMs THROUGH PROMPT ENGINEERING

Transitioning from the broad discussion on the evolution and impact of LLMs, we now delve into a more nuanced aspect crucial for harnessing the potential of these models: prompt engineering. This key technique entails crafting precise queries or instructions to elicit accurate and useful outputs from the model. The manner in which a question or command is phrased significantly influences the model's understanding and the subsequent response, making prompt engineering a vital skill for those looking to effectively leverage LLMs.

Recent headlines spotlighting lucrative salaries and burgeoning demand for a new role, "Prompt Engineers," reflect the growing recognition of prompt engineering's importance in the industry. The following table illustrates a variety of prompt engineering techniques that showcase the different approaches to interacting with LLMs and guiding their responses to achieve desired outcomes:

Technique	Description	Example
Zero-shot	Straight-up query.	"What is the capital of France?"
Few-shot *(Brown et al., 2020)*	Providing previous examples to guide the model.	"Question 1: What is the age of the patient? Answer: 45 Question 2: What are the symptoms? Answer: Fever and cough"
Reflexion *(Shinn, N et al, 2023)*	Forces self-reflection on the model's part. This technique improved GPT-4's accuracy from 80% to 91%.[1]	"Review your response and propose any modifications that can improve your response."
Chain of Thought *(Wei et al., 2023)*	Long-form, multi-step reasoning.	The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1. A: Adding all the odd numbers (9, 15, 1) gives 25. The answer is False. The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1. A:
Tree of Thought *(Yao et al., 2023)*	Branching paths of reasoning	Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point then they leave. The question is...

Table 1: Examples of a selection of prompt-engineering techniques

The effectiveness of prompt engineering is underscored by a simple application of the reflexion technique, which remarkably boosted GPT-4's accuracy from 80% to an impressive 91% on a human evaluation dataset (Shinn, N et al, 2023).

APPLICATIONS AND CHALLENGES IN CLINICAL PROGRAMMING

A promising application of LLMs in the clinical realm lies in ad-hoc data analysis. However, leveraging public LLMs like ChatGPT brings forth concerns surrounding Patient Confidentiality, Accuracy, and Reproducibility. Implementing dedicated architectures and permission layers can address these issues, ensuring compliant data interaction. Patient Confidentiality is of paramount importance, and public LLMs, not designed for handling sensitive information, necessitate robust safeguards. Accuracy and Reproducibility, crucial for clinical applications will also have to be addressed.

Imagine an application tailored for clinicians, programmers, and statisticians, enabling natural language querying of clinical data. Such a system, while harnessing LLMs' strengths, would incorporate safeguards for CAR (Confidentiality, Accuracy, Reproducibility). With a well-architected framework, this application could execute tasks ranging from data summarization to complex analysis, democratizing access to clinical insights.

ANALYTICS ASSISTANT: A CASE STUDY

Here, we introduce the Analytics Assistant. The core goal of this system is to lower the barrier to generating clinical insights, particularly through code generation, by smartly utilizing metadata extracted from SDTM/ADaM datasets. This metadata, devoid of any confidential information, is then interacted with a large language model.

The uniqueness of our approach lies in leveraging the SDTM metadata, which proves to be more insightful compared to traditional database schema. For instance, the codelists within the metadata enable the LLM to discern structures in the data such as available lab tests in the LB domain, thus allowing robust specification and relationship inference between lab test (LBTEST) and lab values (LBSTRESN) without having to worry about misspellings or case-sensitivity of filter clauses.

In refining the interaction with the LLM, prompt-engineering best practices are employed, alongside fine-tuning the model on SDTM rules from the SDTM IG and, in the case of SAS, fine-tuning it on SAS due to LLMs like ChatGPT's limited proficiency with it. Task-specific prompting further delineates table from plot generation, enhancing the clarity of requests made to the LLM.

The generated code (Appendix A) is surprisingly effective and is executed locally on our data. The efficacy extends beyond code generation; the local execution ensures what we see is a harmonious interplay of code and data, addressing confidentiality and accuracy concerns. Moreover, since the LLM produces code, reproducibility is inherently built in. The generated code comes well-documented, allowing us to verify the intent effortlessly and open up additional use cases, such as for SAS programmers interested in learning R.

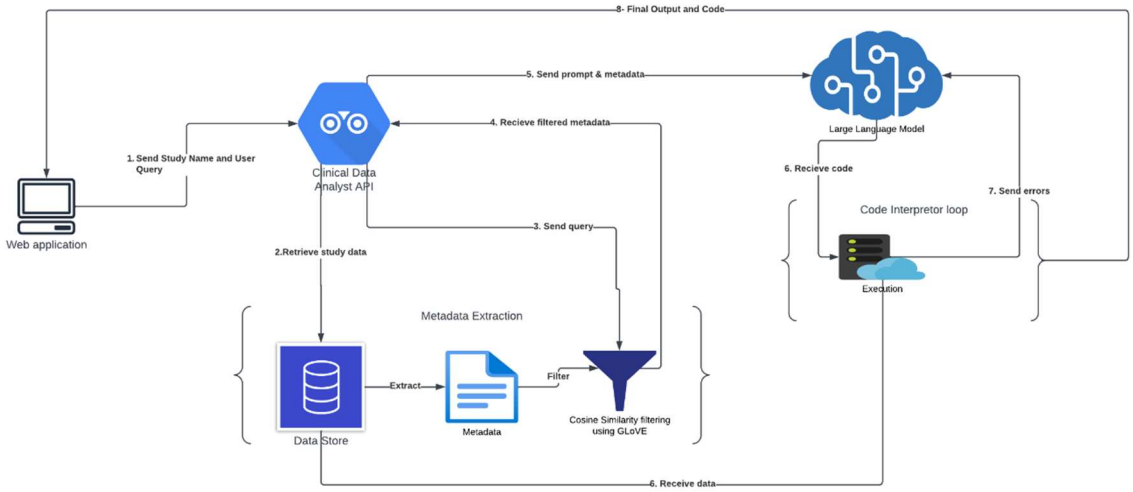

Figure 3. Architecture of Analytics Assistant

A feedback layer *(Fig.3)* evaluates the execution and the generated output. If discrepancies arise, the error trace and history are fed back to the LLM for another round of generation, incrementing the 'temperature' to encourage more creative solutions from the LLM. This feedback loop iterates up to three times, post which either the final output is presented (*Fig.4*), or a message for further user clarification is displayed.

.

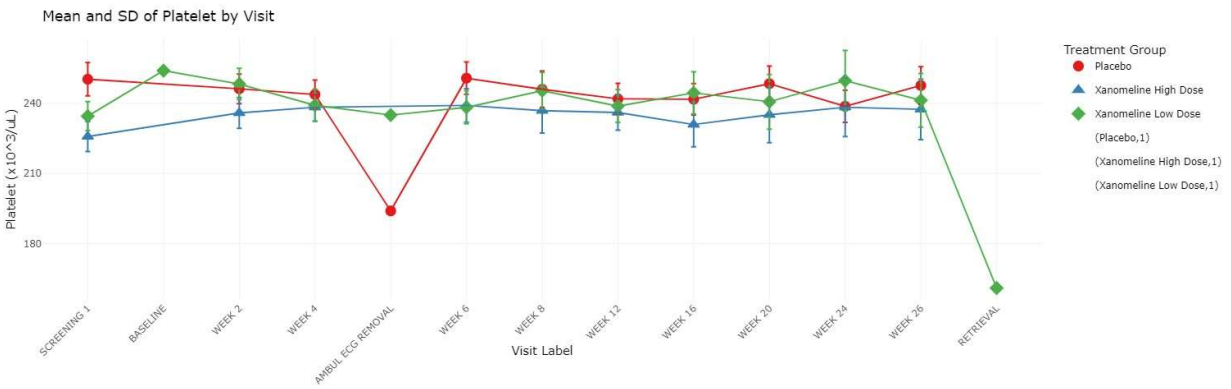

Figure 4. Output generated by Analytics Assistant for the prompt "“Plot the mean and standard deviation of platelet levels across different visits, separated by actual treatment groups. Exclude any unscheduled visits. Connect the mean values with lines and use different colors for each treatment group”

Finally, the nature of user prompts handled is broad, ranging from merging datasets, complex data derivations, to navigating deliberately vague requirements. For instance, a vague intent like, "I am concerned about the safety of actual treatments," metamorphoses into bar plots of adverse event severity by treatment group.

FUTURE OUTLOOK AND CHALLENGES

The domain of LLMs is continuously evolving, with the rise of open-source models and a burgeoning ecosystem of tools and applications painting a promising picture for clinical programming (*Fig 6*). These advancements go beyond merely enhancing data analysis—they pave the way for innovative solutions in patient engagement and healthcare management. The advent of open-source models is a significant milestone for the pharmaceutical industry, offering the possibility of local model hosting to ensure patient data protection. However, challenges such as the infrastructure requirements for model deployment, including the need for multiple high-end GPUs, remain. Yet, the open-source community is tirelessly working towards overcoming these hurdles, with initiatives like LLaMA2 (Touvron, H. et al, 2023) and llama.cpp (L. Team, 2023) showcasing the ongoing efforts to run the LLaMA model using 4-bit integer quantization on a MacBook, exemplifying the continuous endeavor to mitigate such challenges.

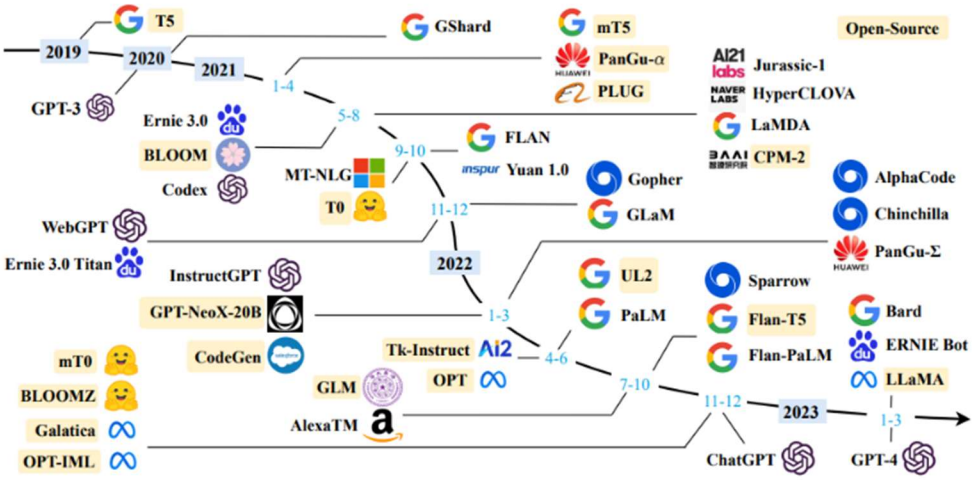

Figure 5. A timeline of existing large language models (having a size larger than 10B) in recent years. (Source: Zhao et al, 2023)

Proceed with Caution: The Hype vs Reality

Gartner, a reputable research and advisory firm known for its accurate market analysis, currently places generative AI and prompt engineering at the peak of inflated expectations in its AI Hype Cycle (*Fig. 7*). This position often signals a potential trough of disillusionment ahead, where the initial excitement might be followed by a phase of negative sentiment if the technology fails to meet overinflated expectations. It's prudent to temper our expectations and perceive these technologies as tools to address concrete use cases, rather than panaceas for all challenges.

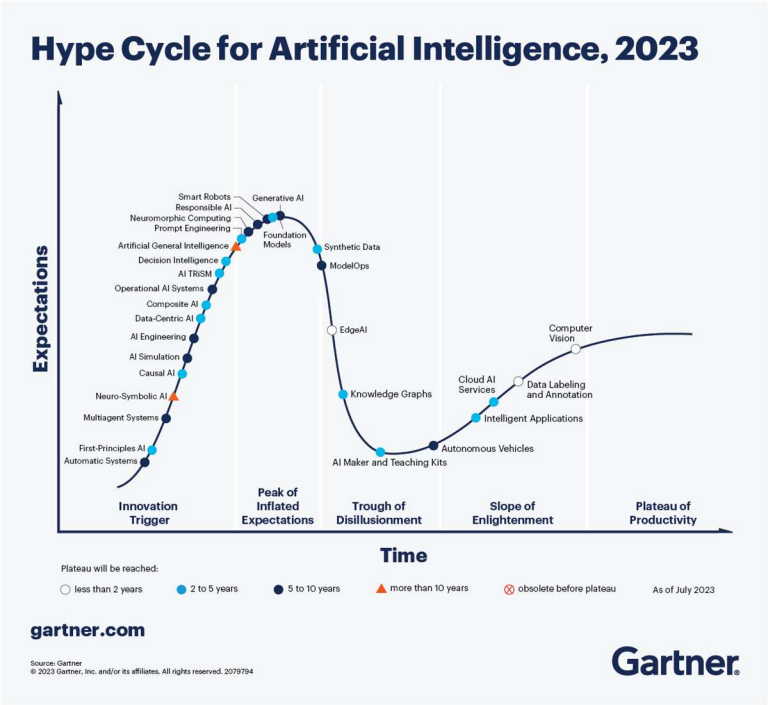

Figure 6.Hype Cycle for Artificial Intelligence, 2023 (Source: Gartner. 2023)

In the exciting landscape of Large Language Models (LLMs), it's easy to get swept up by the hype. As we navigate this emerging technology, it's crucial to maintain balanced expectations and focus on practical, solvable use cases. Several key challenges need to be addressed to unlock the true potential of LLMs, ranging from model reliability issues like hallucinations to the need for more effective prompting and fine-tuning. Moreover, the right infrastructure, company-specific adaptations, and human oversight are critical for successful implementation. Initiatives like Llama.cpp highlight the ongoing work to tackle the challenges of model size and efficiency.

In this fast-evolving landscape of Large Language Models, there's never been a more exciting time to be a part of this field. The models are increasingly powerful and incredibly accessible, opening doors to new possibilities. While challenges remain, the ongoing endeavor to overcome these hurdles is where the true beauty of this technological journey lies, paving the way towards a future replete with unprecedented opportunities.

CONCLUSION

The journey into the domain of Large Language Models (LLMs) opens a realm of promise. The allure of simplified processes and augmented efficiency is tangible, yet it carries a measure of caution. The buzz around LLMs can be enthralling, but the true merit is discovered in their pragmatic application to resolve real-world challenges. The narrative of the Analytics Assistant in this paper provides a glimpse into the possibilities while also highlighting the hurdles awaiting resolution. This untapped potential of LLMs is a call to action for us to dig deeper, to innovate, and to devise practical solutions to the challenges ahead. It's about advancing cautiously, learning from each endeavor, and progressively unlocking the potential of LLMs in a responsible, ethical, and meaningful way. A vast horizon awaits exploration, and though the path ahead is laden with challenges, it’s ripe with the promise of discovery and significant progress. Is this indeed a new chapter in clinical programming? The answer lies in the hands of the broader community, whose actions and explorations will shape the future of this exciting frontier.

REFERENCES

OpenAI, Chatgpt https://chat.openai.com, (2023), Accessed: 2023-07-30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention Is All You Need. arXiv preprint arXiv:1706.03762. Available at: https://doi.org/10.48550/arXiv.1706.03762

Chernyavskiy, A., Ilvovsky, D. and Nakov, P., 2021. Transformers: "The End of History" for NLP? arXiv preprint arXiv:2105.00813. Available at: https://arxiv.org/abs/2105.00813.

Ananthaswamy, A. (2023). In AI, is bigger always better? Nature, 615, 202-205. https://doi.org/10.1038/d41586-023-00641-w

Demand Sage. (n.d.). Road to 100 Million Users For Various Platforms. [online] Available at: https://www.demandsage.com/chatgpt-statistics/ [Accessed on: 10 October 2023].

X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang et al., Zero-shot information extraction via chatting with chatgpt, arXiv preprint arXiv:2302.10205 (2023).

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., 2020. Language Models are Few-Shot Learners. arXiv. Available at: https://arxiv.org/abs/2005.14165

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K. and Yao, S., 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv preprint arXiv:2303.11366. Available at: https://arxiv.org/abs/2303.11366.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. and Zhou, D., 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv preprint arXiv:2201.11903. Available at: https://arxiv.org/abs/2201.11903.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y. and Narasimhan, K., 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv preprint arXiv:2305.10601. Available at: https://arxiv.org/abs/2305.10601.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Canton Ferrer, C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T., 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv preprint arXiv:2307.09288. Available at: https://arxiv.org/abs/2307.09288.

H. Face, Transformers language modeling https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling, (2023).

L. Team, Llama.cpp https://github.com/ggerganov/llama.cpp, (2023).

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A Survey of Large Language Models. arXiv. Available at: https://arxiv.org/abs/2303.18223.

Gartner. (2023). Hype Cycle for Emerging Technologies, 2023. [online] Available at: https://www.gartner.com/en/documents/4597499 [Accessed on: Accessed on: 10 October 2023].

APPENDIX A

Code generated by Analytics Assistant for the prompt “Plot the mean and standard deviation of platelet levels across different visits, separated by actual treatment groups. Exclude any unscheduled visits. Connect the mean values with lines and use different colors for each treatment group” after selecting for the R programming language.

```r
# Step 1: Merge the lb and dm datasets based on 'STUDYID' and 'USUBJID' columns
# lb and dm are assumed to be pre-existing datasets
merged_data <- lb %>%
  inner_join(dm, by = c('STUDYID', 'USUBJID'))  # Perform inner join on STUDYID and USUBJID

# Step 2: Filter for Platelet test results and preprocess
platelet_data <- merged_data %>%
  filter(LBTEST == 'Platelet') %>%  # Keep only the rows where LBTEST is 'Platelet'
  filter(!startsWith(VISIT, 'UNSCHEDULED')) %>%  # Remove rows with unscheduled visits
  arrange(VISITNUM) %>%  # Sort data by VISITNUM (visit number)
  mutate(VISIT = factor(VISIT, levels = unique(VISIT), ordered = TRUE))  # Convert VISIT to ordered factor

# Step 3: Compute summary statistics (Mean and SE) grouped by VISIT and ACTARM
summary_data <- platelet_data %>%
  group_by(VISIT, ACTARM) %>%  # Group data by VISIT and ACTARM
  summarise(
    Mean = mean(LBSTRESN, na.rm = TRUE),  # Calculate the mean of LBSTRESN, ignoring NA values
    SE = sd(LBSTRESN, na.rm = TRUE) / sqrt(n()),  # Calculate the standard error
    .groups = 'drop'  # Drop the grouping
  )

# Step 4: Generate the ggplot
ggplot(summary_data, aes(x = VISIT, y = Mean)) +  # Base ggplot mapping VISIT to x-axis and Mean to y-axis
  geom_point(aes(color = ACTARM, shape = ACTARM), size = 3) +  # Add points colored and shaped by ACTARM
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE, color = ACTARM), width = 0.2) +  # Add error bars
  geom_line(aes(group = ACTARM, color = ACTARM)) +  # Add lines connecting points within each ACTARM group
  labs(
    title = 'Mean and SD of Platelet by Visit',  # Add plot title
    x = 'Visit Label',  # Label for the x-axis
    y = 'Platelet (x10^3/uL)'  # Label for the y-axis
  ) +
  theme_minimal() +  # Apply minimal theme
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis text
    legend.title = element_text(face = 'bold'),  # Make legend title bold
    legend.position = 'bottom'  # Position legend at the bottom
  ) +
  scale_color_brewer(name = 'Treatment Group', palette = 'Set1') +  # Set color scheme for ACTARM
  scale_shape_manual(name = 'Treatment Group', values = c(16, 17, 18, 19, 20))  # Manually set shape values
```

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Karma Tarap
Bristol Myers Squibb
Boudry / 2017
Email: karma.tarap@bms.com

Brand and product names are trademarks of their respective companies.