



RESEARCH APPLICATION OF LARGE LANGUAGE MODELS FOR CORPORATE RISK

FINAL INTERNSHIP REPORT
DSBA 6400 – FALL 2024 - SPRING 2025
MASTER OF SCIENCE, DATA SCIENCE & BUSINESS ANALYTICS

SUBMITTED BY: JAKE BRULATO
FACULTY SUPERVISOR: ROBERT FOX
MENTOR: ANWESHA BHATTACHARYYA, LEAD QUANTITATIVE ANALYTICS
SPECIALIST

Contents

1. Executive Summary	3
2. Introduction.....	3
2.1. Business Objectives	3
2.2. Business Problem	3
2.3. Background.....	4
3. Methods.....	4
3.1. Barriers	5
3.2. Mentor's Role	5
4. Results and Interpretation	6
4.1. Internal Inventory of GenAI and LLMs	6
4.2. Benchmark Comparisons of Open-Sourced Models	6
4.3. Prompt Hub and Presentations Resources for AIA Seminar	8
4.4. ReAct (Reason + Action) Prompting.....	9
5. Discussion and Conclusion	10
6. Appendix.....	11
6.1. Internship Experience	11
6.2. References	11

1. Executive Summary

This report summarizes the internship experience from October 2024 to March 2025 with Wells Fargo's Decision Science & Artificial Intelligence (DSAI) team within the Corporate Risk division. The internship focused on enhancing internal model validation and tooling through the evaluation and implementation of open-source generative AI models. The deliverable was the development of a consolidated worksheet and visual resources to support model selection and prompt strategy for validators. These materials were integrated into internal wikis and presented in the monthly seminar series to support cross-team collaboration.

Advanced prompting frameworks, including Chain-of-Thought, ReAct (Reason + Action), and Retrieval-Augmented Generation (RAG), were researched and tested. Due to infrastructure limitations, the ReAct approach was adapted using RAG to simulate tool use through internal document retrieval. While not a full ReAct implementation, this adaptation enhanced model reasoning and performance in restricted environments.

The internship provided the opportunity to apply DSBA coursework in a high-compliance, corporate enterprise setting. It contributed to the organization's research of deploying secure, scalable generative AI solutions within corporate risk, and future exploration of agent-based frameworks and internal AI systems.

2. Introduction

2.1. Business Objectives

The following objectives were accomplished during the internship period:

- Conducted a comprehensive inventory of all Generative AI or Large Language Models currently and previously in use at Wells Fargo. Included consolidating model status into a single reference sheet for use by validators and relevant personnel.
- Researched benchmark performance for certain downloaded Hugging Face models, identifying optimal scenarios to support model validation efforts for team members.
- Developed resources—including visual aids and explanatory content—to support both validators and members of the Generative AI (GenAI) team in corporate risk. These materials were publicized on the internal knowledge-sharing wiki.
- Investigated the ReAct prompting framework, with the objective of replicating its logic within internal software environments. This research influenced the design of tools incorporating Retrieval-Augmented Generation (RAG) techniques to improve internal GenAI capabilities.

2.2. Business Problem

The primary business challenge addressed by this project involves creating a guide of generative resources when handling internal or sensitive information. There is a strategic need to establish a

secure, internally governed chatbot solution capable of meeting user demands for efficiency and performance.

The team seeks to evaluate and recommend optimal open-source models, such as LLaMA 3, for internal deployment, while encouraging a shift away from defaulting to third-party vendors. The internal solution must provide a sufficient level of functionality to support a range of workflows, minimizing potential data exposure risks associated with external platforms. This project supports Wells Fargo's broader goal of enabling responsible and secure adoption of generative AI technologies within corporate risk.

2.3. Background

Wells Fargo operates across numerous divisions, each with distinct business functions and risk profiles. Many of these functions involve proprietary processes or require reassessment due to evolving regulatory and operational considerations. Within this environment, the Corporate Risk division serves as a critical second line of defense, following initial evaluations by operational management. Responsibilities include conducting thorough reviews of risk-related requests and compiling findings into formal documentation for internal governance.

These review processes can be time-intensive, particularly when requests require deep evaluation. Resulting documents often range from 30 to 150 pages, depending on the complexity and justifiability of the content. Human reviewers are tasked with reading and analyzing these materials during quarterly reviews or reassessment cycles, which contributes to prolonged workflows and manual effort.

To streamline this process, the Corporate Risk team is exploring the integration of open-source pretrained Large Language Models to assist in internal review applications. These tools are being evaluated for their ability to process the full context of documents or targeted snippets to assist with summarization and content extraction. While current implementations show promise when analyzing smaller text segments, challenges remain in performance and contextual accuracy when processing longer-form documents, a notable consideration to other vendor models.

3. Methods

For each generative AI model under review, validators received input data directly from the original requestor in the form of a use case report. Following this, a consolidated document was developed through multiple rounds of iterative stress testing. This document serves as a formal "Risk Ranking" assessment that is submitted to Wells Fargo's internal review systems. While approved models progress through their lifecycle, this documentation is revisited and updated as necessary; however, this process is often time-consuming due to the depth and complexity of the assessments required.

To address the inefficiencies, research was initiated in collaboration with the ATOM sister team, with the goal of expediting the validation process using smaller generative AI models downloaded from Hugging Face. Evaluation focused on the following factors:

- a) The linguistic coherence and quality of the generated outputs,
- b) Adherence to predefined system guardrails (e.g., prompt constraints),
- c) Detection of toxicity or hallucinations that may lead to inaccurate or inappropriate outputs
- d) Overall model performance relative to the specific task and efficiency in terms of token usage.

The initial research effort involved the evaluation of seven generative models; however, this was later expanded to fourteen. These models varied in release dates, parameter sizes, and maximum token lengths, indicating evolution of generative models over the years. The goal was to improve performance on long-context tasks like document format generation by refining the prompting strategies and optimizing the flow of information into the models.

In-depth reviews were conducted of official research papers, architectural documentation, and chat templates associated with each model. Several advanced prompting techniques were tested including Chain-of-Thought prompting, Tree-of-Thought, ReAct (Reason + Action), and Retrieval-Augmented Generation (RAG) to enhance output reasoning and contextual understanding.

3.1. Barriers

Wells Fargo is a risk-averse company with high security in most, if not all their operations causing many barriers or standstills to occur during the period of the business project. The following is what was encountered:

- Permission access to specific level models that required approval in different time zones, causing delay in responses and access to necessary materials to create outputs. Primarily allocated for validation of certain models in a specific shared directory under the team. The issue was resolved approximately one week later only when the mentor emailed them to expedite it, causing a delay in the project for the team.
- The organization uses many different tools for coding with most divisions using different applications and clusters. Access to the cluster for specific notebooks to work with Large Language Models caused a delay in initial testing, creating a short halt for about half a week. The results led to the creation of two notebooks, one created specifically for CPU based usage, while the other had access to the Wells Fargo GPU cluster.
- Most users beforehand had only used CPU master-based environments and were unfamiliar with GPU based ones. A list of GPU master environments from the sister team was acquired, global cloning of said environments as each large language model had their own with allocated memory created issues in setup and delays costing approximately two weeks.
- Research into higher level prompting structures came with issues when it came to specific installations. Depending on the cloned master environment not having the correct compiler (Rust, etc.), proper installation of these packages for the research was not possible unless it was already internal.

3.2. Mentor's Role

Mentors played a critical role throughout the internship by providing consistent guidance and support during weekly or biweekly meetings. These sessions served as a forum for addressing questions, resolving issues, and aligning project goals. Mentor coordinated collaboration with the sister team, including initial planning meetings and periodic progress reviews. Detailed feedback

and analytical suggestions helped shape the structure of the work and provided clear directions for delivering a final product that met team expectations.

4. Results and Interpretation

4.1. Internal Inventory of GenAI and LLMs

A combined dataset was compiled from two primary sources: a public-facing list of generative AI models accessible to all Wells Fargo lines of business, and a redacted internal list provided by the Data Science and Artificial Intelligence (DSAI) team including the model validations the team assisted with. In total, 70 models were identified that met the criteria for classification as either Generative AI (GenAI) or Large Language Models (LLMs). Each model was categorized using a standardized framework that included the following attributes show in Figure 1:

- Line of Business: The organizational unit associated with the model's development or use.
- Model Name and Identification Number: Unique identifiers used for internal tracking.
- Usage Category: Models were assigned to one of four defined categories based on their primary function:
 1. Non-Generative Classification
 2. Extractive Methods (e.g., Semantic Search, Extractive Summarization, Extractive Question Answering)
 3. Generative Methods (e.g., Abstractive Summarization, Generative Classification, Generative Information Retrieval)
 4. General-Purpose Generation

Additional data fields included the model’s evaluated risk rank, derived through structured stress testing and review of the Model Development Document (MDD), its current lifecycle phase, and a descriptive summary outlining functionality and intended use cases.

LOB	Model Name	Model #	Usage Category	Usage Name	Risk Rank (RR)	Lifecycle Status	Description	Comment	Update date
Corporate & Investment Banking	Moody's Research Assistant Chatbot	15408	3	Generative Information Retrieval		5 In Development	Model is used to summarize information from Moody's CreditView Research with linked citations to original source. Research. The Chatbot accepts free-text prompts from users as inputs and outputs text results using generative AI.		
	Contact Center Agent Virtual Assistant (AVA)	15426	2	Extractive Information Retrieval		3 In Development	The tool accepts agents' inquiries in plain English via Microsoft Teams chat as inputs and outputs the search results as they serve customers.		
Commercial Banking	Loan Document Harmonization Project	15412	1	Non-Generative Classification		5 Retired	Model helps categorize Commercial Banking templates for loan documentation accepts loan documentation from existing repositories as inputs and outputs templated clauses from the loan documentation, organized by clause type.		
	Ipiphany Voice of Customer (VOC) Insights Tool	15436	3	Generative/Abstractive Summarization		5 In Development	Tool is used to accelerate voice of customer insights for the SDI Digital team. Accepts app reviews as inputs and outputs summarized themes based on the use of NLP/Generative AI.		

Figure 1: Redacted visual of the consolidated list with removal of sensitive tools that were added to internal resources.

4.2. Benchmark Comparisons of Open-Sourced Models

Models evaluated included Meta’s LLaMA 3, 3.1, and 3.2 Instruct (1B–8B), CodeLLaMA, Google’s Gemma and PaliGemma, Alibaba’s Qwen series. Later DeepSeek’s distilled variants, and SmolAI’s lightweight SmolVL models were assessed but removed. These models were

assessed using official documentation, arXiv research, and repository data, with evaluation focused on context length, generation capacity, fine-tuning compatibility, and benchmark performance. Key benchmarks included MMLU, HellaSwag, GSM8K, HumanEval, and MATH, with results summarized in Figure 2a.

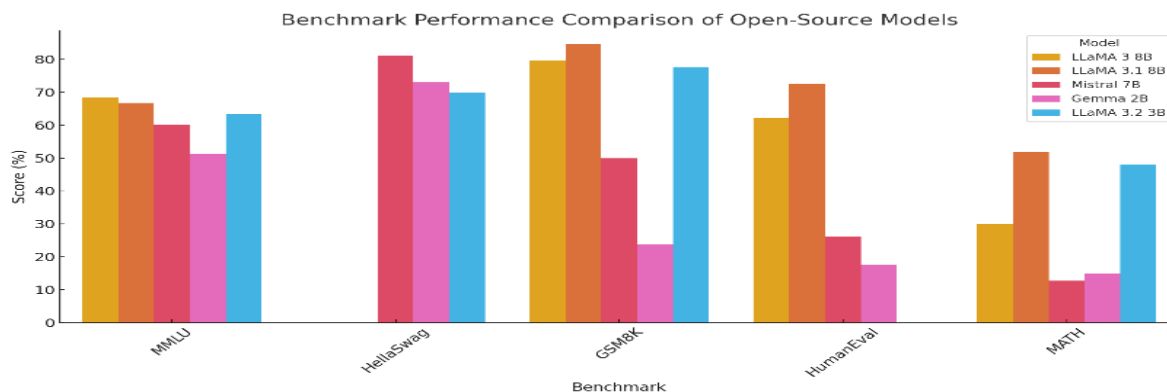


Figure 2a: Benchmark comparisons of main set of text generation models from initial research, not included models were added later in the project stage.

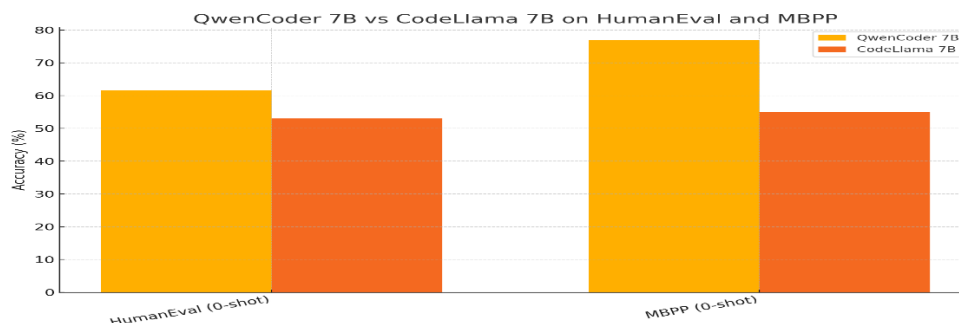


Figure 2b: Benchmark comparisons of code-based models, designed to be a potential alternative to chat-based text generation models above.

Figure 2a illustrates comparative performance across these tasks, showing that LLaMA 3.1 8B Instruct consistently led in reasoning (GSM8K – 84.5%), code generation (HumanEval – 72.6%), and math (MATH – 51.9%). It demonstrated high accuracy and task generalization, making it the most balanced model overall. LLaMA 3.2 3B also showed strong math performance (MATH – 48%).

As shown in the figure, Mistral 7B excelled in commonsense reasoning (HellaSwag – 81%) but lagged in mathematical and programming tasks. Models like Gemma 2B offered greater efficiency but performed significantly lower across benchmarks, indicating their suitability for lightweight or low complexity use cases within hybrid systems. LLaMA 3.1 8B Instruct is recommended as the primary baseline model for internal deployment due to its consistent, top-tier performance across diverse benchmark categories, as supported visually in Figure 2a.

4.3. Prompt Hub and Presentations Resources for AIA Seminar

A consolidated worksheet was created to highlight optimal open-source models for internal use, including LLaMA models, CodeLLaMA, Mistral, Gemma, PaliGemma, Qwen2-VL-Instruct, and QwenCoder. Each model was evaluated on tasks such as paraphrasing, summarization, grammar correction, question answering, QA/QC, code generation, and debugging.

Recommended model parameters and prompting techniques—such as zero-shot, few-shot, chain-of-thought, and tree-of-thought—were included based on testing results. Common testing errors were also documented to help users avoid known issues and improve the prompt.

Models	Token Limits	Parameter Changes	Testing Errors	Best Prompting Practices	Paraphrase Prompt	Summarization Prompt	Grammar Check	Q/A Prompt
Llama 3 - 70, 8B	• Llama 3: 8,192	Temperature: .7 top_p: .9 max_new_tokens: 10000 do_sample: true	Llama 3 sometimes has failed to halt in its generation, making incomplete or too lengthy responses. Depending on what was prompted, the style control can be inconsistent, defining specifics with context examples gives it the ability to be more through in generation. Sometimes repeats generation, define in prompt to be unique and avoid repetition.	• Best to keep the prompt limited to a single task based on token length for text generation. • Create concise constraints to avoid overloading the prompt. • Error correction prompts help with paraphrasing and checking, Llama 3 handles these well. • Binary questions for accuracy to ensure their is clarity when making decisions. • If complexity is desired, make instructional prompts that clearly outline the structure of the task and the output. • Few Shot and Zero Shot prompting work well, but anything beyond that like Chain-of-Thought prompting will underperform.	• Llama 3 and its instruct are specifically tuned for instructional tasks because of its robust instruction following ability in text generation. • Examples: • "Rewrite this text with [format: creative/ informal] tone: Original: "This device works efficiently under heavy load" Format: "the device operates with exceptional efficiency under high workloads" Now rephrase this: "[Insert Text]" • "Reword this text in the style of the following examples: Example 1: "He quickly solved the problem" -> "He resolved the issue in no time" Example 2: "Her voice was soothing the crowd" -> "Her voice calmed everyone present" Now, paraphrase: "[insert text]"	• Strong contextual understanding allows it to adapt different styles and lengths, condensing it into concise summaries. • Examples: • "Summarize the text into a hierarchy of main points and subpoints. Use this format as a guide: Main point: The park is a popular destination. Subpoint: Known for its diverse wildlife. Subpoint: Offers scenic trails and paths. Now Summarize this text like it: "[Insert Text]" • Summarize the following text into different lengths of 500/200/100 words: • Write an abstractive summary of the following text or extract the main important key sentences.	• Llama 3's pretraining allows it to have diverse grammar checking abilities and multi layer error detection. • Examples: • "Here's how grammar corrections are explained: Example: "She go to the market yesterday" Correction: "She went to the market yesterday" Explanation: The verb 'go' should be in the past tense to match the time indicator 'yesterday' Now check the grammar of this text and explain your corrections: "[Insert Text]" • "Identify and fix grammar mistakes in this text without changing its meaning:"	• Examples: • "Answer questions using the guidance: • Example 1: Q: "What is the capital of France" A: "The capital of France is Paris" • Example 2: Q: "How does photosynthesis work" A: "Photosynthesis is the process by which plants convert sunlight into energy" Now answer this question: "[Insert Text]" • "If the user's question lacks ask a clarifying question first, examples as guidance. • Example: Q: "What are the effects of climate change" A: "Could you specify whether global or regional effects?" Now answer this question: "[Insert Text]"

Figure 3: Redacted section of the prompt hub's detailed tab, offering model-specific paraphrasing, summarization, grammar checks, and Q&A prompts. Includes best practices, parameter settings, and example prompts to help users structure inputs.

Side-By-Side Specifics – Llama Models

Llama 3	Llama 3.1	Llama 3.2
<ul style="list-style-type: none">Considered the baseline, tries to include as much as possible.Has the ability to go further if prompted correctly, will keep a lot of jargon.Tends to keep it limited to simple structure text generation.Focuses on listing the features but makes it too long impacting readability.Works for simple information or clarification but will have more difficulty in long handling in specific contexts.Recommend:<ul style="list-style-type: none">Short paraphrasing, summarization, and grammar checkingQ/A and QAQC is possible but would best be done by another model.	<ul style="list-style-type: none">Adds more depth to generation, describing specifics like user roles and the amount of people (20k data scientists, etc.)More clarity and better sentence structure in the output.Seamless transitions to other topics, making it more refined than Llama 3Highlights most of the technical features and their use cases.Most robust, handles large streams of data making very good for large documents.Recommend:<ul style="list-style-type: none">Most tasks, especially with large document streams chunked properly.Q/A and QAQC works best along with the other prompt types.	<ul style="list-style-type: none">Latest model with smallest parameter count, designed for resource constrained environments.Strikes a balance between detail and simplicity, making well thought and concise prompts.Language reasoning is good but will hallucinate if you don't specify that you want to keep the original text. (See Grammar Refined for example)A simpler less refined prompt worked better, keeping the original text with improvements.Recommend:<ul style="list-style-type: none">Zero-Shot to Few-Shot concise prompts for paraphrasing and summarization.Grammar Checking, Q/A, and QAQC best with Zero-Shot as those tend to hallucinate if too wordy.

Figure 4a: Example of information on the technical slides, others included code, and more drawn-out examples for validators with specific models.

Alongside the worksheet, two presentations were developed, the first provided a technical comparison of models from similar generations or architectures, enabling side-by-side visualization of performance and structural differences in a format presented in figure 4a. The second was tailored specifically for a one-hour seminar delivered to the GenAI team within the

ATOM group, designed to show specifics of the models in a condensed format like in figure 4b with non-main text generation models moved in the appendix.

Llama 3 Vs. 3.1 Vs. 3.2 - Paraphrasing (Refined)

Llama 3: Straightforward baseline, lacks breakdowns and nuanced explanations	Llama 3.1: Expanded and integrated descriptions with clear distinctions	Llama 3.2: Organized into structured format, making it more engaging
The Central Analytics Support & Enablement (CASE) platform serves over 20,000 data professionals, offering features like the intent-driven chatbot CASEY, the collective intelligence framework, conditional ticketing, and MS Graph-enabled search. This unified platform enables product and support teams to efficiently address customer needs, manage documentation, and handle inquiries.	Our enterprise support platform, Central Analytics Support & Enablement (CASE), serves over 20,000 data professionals, including data scientists, analysts, and developers. CASE features a chatbot, "CASEY," powered by MS Copilot Studio, as well as the collective intelligence framework, a conditional ticketing system, and MS Graph-enabled search. This integrated platform enables product and support teams to efficiently manage customer needs, document product knowledge, and respond to inquiries in a unified environment.	The Central Analytics Support & Enablement (CASE) platform serves over 20,000 data professionals, providing a unified solution for product and support teams. Key features include: <ul style="list-style-type: none"> CASEY, a non-Gen AI chatbot built on MS Copilot Studio The Collective Intelligence (CI) framework A conditional ticketing system MS Graph-enabled search

Strikes a balance between detail and simplicity, and changed format without instruction

• With a more specific goal in mind focusing on clarity and conciseness, we can see that more notable changes occur, specifically with format or length.

Figure 4b: Condensed slide format, showcasing relevant points, annotations with an end conclusion for each slide. Designed to take a fraction of someone's time.

4.4. ReAct (Reason + Action) Prompting

After the completion of foundational prompt engineering resources, research efforts expanded to investigate more advanced prompting strategies. One primary area of focus was the ReAct framework, as outlined in "ReAct: Synergizing Reasoning and Acting in Language Models" (Yao et al., 2022). This technique builds upon traditional prompting methods, such as chain-of-thought and tree-of-thought, by introducing an iterative reasoning loop consisting of Thought, Action, and Observation phases.

```

### Thought

The task seems straightforward - comparing the key features of two plans, specifically focusing on how they handle healthcare costs. Upon reviewing the given text, I notice that there's some overlap between the HSA Plan and the Copay Plan with HRA regarding their structure and payment mechanisms. This implies that understanding how these similarities differ might hold the key to identifying the primary distinctions between the two plans.

### Action: Step 1

I'll start by looking into the way the plans deal with annual deductibles and out-of-pocket expenses. It appears that for the HSA Plan, charges for in-network services do not affect the annual deductible and out-of-pocket limits when compared to out-of-network expenses. On the contrary, the Copay Plan with HRA allows for cross-accumulation between its in-network Tiers 1 and 2 deductibles.

### Observation

This comparison suggests that the HSA Plan maintains separate deductibles and out-of-pocket limits for in-network and out-of-network expenses, whereas the Copay Plan with HRA combines them through cross-accumulation.

### Action: Step 2

Considering the unique characteristics of each plan, my next move would be to examine how prescription drugs fit into their respective frameworks. According to the provided content, the HSA Plan requires meeting the annual deductible before applying copays to prescription medications, except for those listed under preventive drug therapy. In contrast, the Copay Plan with HRA does not require meeting the deductible before copays apply to prescription medications.

### Observation

These findings indicate that the HSA Plan imposes stricter requirements for accessing prescription medication benefits than the Copay Plan with HRA, where copays begin immediately without needing to satisfy a deductible.

### Final Answer

In summary, the major differences between the Copay Plan with HRA and the HSA Plan lie in their approaches to handling annual deductibles and out-of-pocket expenses, particularly when it comes to combining in-network and out-of-network expenses. Additionally, the HSA Plan places stricter conditions on accessing prescription medication benefits compared to the Copay Plan with HRA. These distinct features highlight the varying strategies employed by each plan to manage patient costs and ensure healthcare accessibility.assistant

```

Figure 5a: Image of the output for the ReAct framework, showcasing that the model is critically thinking in steps and following the logic for the "Reason".

Figure 5a demonstrates the ReAct prompting sequence using publicly available test data to remain compliant with internal security requirements. The model successively refines its

understanding of two healthcare plans by breaking the task into smaller reasoning steps. Each action is followed by a corresponding observation before proceeding to the next step, ultimately leading to a well-supported final answer.

Although ReAct was originally designed to leverage external tools, such as live internet searches, for its “action” phase, Wells Fargo’s infrastructure and data governance policies restrict such external connectivity. To address this, an internal alternative was developed using Retrieval-Augmented Generation (RAG). This approach simulates external tool use by retrieving information from internally stored and pre-approved documentation.

Figure 5b illustrates the internal RAG implementation, where the model queries a structured database of chunked documents and uses relevance-scored sources to inform its final answer. This not only supports the action phase of the ReAct loop but also ensures that all content used remains compliant with data privacy policies.

```
Sources:

[1] Relevance score: 0.53
Preview: Once enrolled in COBRA, let BenefitConnect™ | COBRA know about any changes in addresses of family members (see “Plan contact information” below). You should also keep a copy, for your records, of any notices you send to the COBRA Administrator

[2] Relevance score: 0.53
Preview: To receive this second qualifying event extension of COBRA continuation coverage, BenefitConnect™ | COBRA must be notified of the second qualifying event within 60 days after the second qualifying event occurs to maintain extension rights under COBRA. To notify BenefitConnect™ | COBRA of the second qualifying event, call 1-877-29-COBRA (1-877-292-6272) (858-314-5108 International only) or online at https://cobra.ehr.com. Failure to notify BenefitConnect™ | COBRA within 60 days of the second qualifying event will make the qualified beneficiary ineligible for the extension rights under COBRA. Are there other coverage options besides COBRA continuation coverage? Yes
```

Figure 5b: Relevance scores from the RAG, showcasing the search function is finding the information and supplying context for the answer, helping the ReAct logic via the “Action”.

While this implementation improved performance over standard chain-of-thought prompting, especially in tasks requiring contextual lookup, it was ultimately classified as a ReAct-inspired RAG variation, rather than a true ReAct implementation. The absence of real-time tool use limited its full expressiveness. Toward the conclusion of the internship, conversations began regarding the exploration of agentic frameworks to support future tool integration. However, due to time constraints, this initiative was transitioned to other team members, while the existing RAG workflow was shared with colleagues for continued development and adoption.

5. Discussion and Conclusion

Recent studies have highlighted growing interest in open-source generative AI as alternatives to commercial models in risk-sensitive environments, a trend strongly supported by the findings of this internship. The work completed demonstrates that internally governed models—such as LLaMA 3.1 8B and Mistral 7B—can be effectively benchmarked, deployed, and optimized for a range of tasks relevant to corporate risk, including summarization, validation support, and documentation assistance. As shown in this study, LLaMA 3.1 8B consistently outperformed peers in code generation, math, and reasoning tasks, suggesting it is a suitable baseline for internal use.

Moreover, the research explored advanced prompting methods, including the ReAct (Reason + Action) framework. While ReAct typically relies on external tools for dynamic interaction, this

study adapted its logic using Retrieval-Augmented Generation (RAG) to simulate internal search behavior. This adjustment proved valuable in enhancing reasoning depth while remaining compliant with Wells Fargo’s infrastructure limitations. The adapted framework, as illustrated in Figures 4a and 4b, highlights the practical application of iterative reasoning even in closed environments. Although this adaptation was not considered a full ReAct implementation, it still contributed to stronger, more informed outputs than traditional chain-of-thought prompting alone.

These findings emphasize the importance of continuing internal development of AI tooling that minimizes reliance on third-party models. The centralized worksheet and visual prompt hub presented to the ATOM team now serve as an accessible resource for validators to identify appropriate models and avoid common prompt-related pitfalls.

Environmental setup barriers and infrastructure delays impacted early-stage testing and reduced the available time for agentic framework exploration. Further analysis is recommended to expand the internal RAG framework and investigate agent-based systems with proper tool-use integration. It supports Wells Fargo’s broader goal of secure AI adoption and highlights opportunities to improve both operational efficiency and technical capability.

6. Appendix

6.1. Internship Experience

The internship provided an opportunity to apply both technical and analytical skills developed through the DSBA program in a professional, real-world setting. The scope of the work aligned closely with interests in open-source and generative AI models, offering a hands-on environment to explore advanced techniques in prompt engineering, model evaluation, and applied research.

Team members actively supported exploration by encouraging experimentation with diverse modeling techniques and offering constructive feedback on proposed improvements. Collaboration across teams, including with quantitative analysts, further enhanced the learning experience, frequent walkthroughs and visual explanations enabled a deeper understanding of adjacent workflows and data pipelines.

The internship also inspired new directions for personal development, including conceptualizing a consolidated web-based interface for interacting with multiple large language models, an idea that stemmed directly from the tasks and tools encountered during the project. Overall, the experience fostered professional growth and sparked creative ideas for future work, both in industry and in independent technical exploration.

6.2. References

Yao, S., Zhao, J., Kassawat, S., Lin, B., Lin, X., Narayanaswamy, B., & Rajani, N. (2022). ReAct: Synergizing reasoning and acting in language models (arXiv:2210.03629). arXiv. <https://arxiv.org/pdf/2210.03629>

Prompt Hackers. (n.d.). Gemma 2 2B vs. LLaMA 3.2 3B model comparison.
<https://www.prompthackers.co/compare/gemma-2-2b/llama-3.2-3b>

Prompting Guide. (n.d.). Prompt engineering guide. <https://www.promptingguide.ai/>

Lin, S., Yu, X., Gao, Y., Yu, Z., Zhang, X., Wang, X., & Zhu, Q. (2025). AgentVerse: A general evaluation benchmark for large language model agents (arXiv:2504.09781). arXiv.
<https://arxiv.org/abs/2504.09781>

Zhang, Y., Zhou, K., Lin, Z., Liang, P. P., Gu, J., & Xu, M. (2024). GROOT: A benchmark for multi-hop question answering with GPTs (arXiv:2412.01262). arXiv.
<https://arxiv.org/abs/2412.01262>

Meta AI. (n.d.). LLaMA Cookbook: Examples and utilities for running LLaMA models. GitHub.
<https://github.com/meta-llama/llama-cookbook>

Inferless. (2024, March 20). Exploring LLMs speed benchmarks: Independent analysis – Part 3.
<https://www.inferless.com/learn/exploring-llms-speed-benchmarks-independent-analysis---part-3>

DeepMind. (2024, February). Gemma: Open models built from research and aligned for safety.
<https://storage.googleapis.com/deepmind-media/gemma/gemma-report.pdf>

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Staerman, J., ... & Scialom, T. (2024). LLaMA 3: Open foundation and instruction models (arXiv:2407.21783). arXiv.
<https://arxiv.org/abs/2407.21783>

Jiang, Z., Tang, M., Savy, T., Weber, T., Xu, M., & Scialom, T. (2023). Mistral: Faster language models with better reasoning (arXiv:2310.06825). arXiv. <https://arxiv.org/abs/2310.06825>