A red sign with white text

AI-generated content may be incorrect.

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

Text

Description automatically generated

Contents

[1. Executive Summary 2](#_Toc196233206)

[2. Introduction 3](#_Toc196233207)

[2.1. Business Objectives 3](#_Toc196233208)

[2.2. Business Problem 4](#_Toc196233209)

[2.3. Background 4](#_Toc196233210)

[3. Methods 4](#_Toc196233211)

[3.1. Barriers 5](#_Toc196233212)

[3.2. Mentor’s Role 6](#_Toc196233213)

[4. Results and Interpretation 6](#_Toc196233214)

[4.1. Internal Inventory of GenAI and LLMs 6](#_Toc196233215)

[4.2. Benchmark Comparisons of Open-Sourced Models 7](#_Toc196233216)

[4.3. Prompt Hub and Presentations Resources for AIA Seminar 7](#_Toc196233217)

[4.4. ReAct (Reason + Action) Prompting 8](#_Toc196233218)

[5. Discussion and Conclusion 9](#_Toc196233219)

[6. Appendix 10](#_Toc196233220)

[6.1. Internship Experience 10](#_Toc196233221)

[6.2. References 10](#_Toc196233222)

# 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. In alignment with Wells Fargo’s goal to reduce dependency on external vendors, models such as LLaMA, Mistral, Gemma, and Qwen were assessed for performance across a variety of tasks including summarization, code generation, and question answering.

A key 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 monthly seminars 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, enterprise setting. It contributed to the organization's broader strategy of deploying secure, scalable generative AI solutions within corporate risk, while also informing future exploration of agent-based frameworks and internal AI systems.

# Introduction

## 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. This included consolidating model status into a single reference sheet for use by validators and relevant personnel. Contributions were also made to the validation of several models listed in this resource.
* Researched benchmark performance for downloaded Hugging Face models, identifying optimal parameters to support model validation efforts and enhance evaluation consistency across the team.
* Developed centralized 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 formatted for publication on the internal knowledge-sharing wiki.
* Investigated the Reason + Action (ReAct) prompting framework, with the objective of replicating its logic within internal software environments. This research informed the design of tools incorporating Retrieval-Augmented Generation (RAG) techniques to improve internal GenAI capabilities.

## Business Problem

The primary business challenge addressed by this project involves reducing reliance on external generative AI tools—such as ChatGPT, Copilot, and Gemini—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.

This initiative 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 sufficient functionality to support a range of workflows, thereby minimizing potential data exposure risks associated with external platforms. Ultimately, the project supports Wells Fargo’s broader goal of enabling responsible and secure adoption of generative AI technologies within its operational ecosystem.

## 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. Its 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 significant manual effort.

To streamline this process, the Corporate Risk team is exploring the integration of open-source pretrained and fine-tuned generative AI tools into internal review applications. These tools are being evaluated for their ability to process full documents or targeted snippets to assist with summarization and content extraction. While current implementations show promise when analyzing smaller text segments, challenges remain in maintaining performance and contextual accuracy when processing longer-form documents—especially in comparison to more powerful commercial models.

# 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 ultimately served as a formal “Risk Ranking” assessment for submission to Wells Fargo’s internal review systems. As the model progresses through its lifecycle, this documentation is revisited and updated; however, this process is often time-consuming due to the depth and complexity of the assessments required.

To address the inefficiencies in this workflow, research was initiated in collaboration with a sister team, with the goal of expediting the validation process using smaller generative AI models hosted on Hugging Face. The evaluation framework focused on the following key 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, reflecting the rapid evolution of the open-source generative AI landscape over the past two years. The central aim was to improve performance on long-context tasks by refining the prompting strategies and optimizing the flow of information into the models.

To support this, in-depth reviews were conducted of official research papers, architectural documentation, and chat templates associated with each model. Additionally, several advanced prompting techniques were tested—including Chain-of-Thought prompting, Few-Shot learning, ReAct (Reason + Action), and Retrieval-Augmented Generation (RAG)—to enhance output reasoning and contextual understanding.

## Barriers

Wells Fargo is a risk averse company with high security in most, if not all of 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 delay in 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. Result 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.
* The teams application to write code had some environmental issues when it came time for further testing. Most users beforehand had only used CPU master based environments and were unfamiliar with GPU based ones. List of GPU master environments from 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 install of these packages for the research was not possible.

## Mentor’s Role

The mentor 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. The 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.

# Results and Interpretation

## 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 I 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.

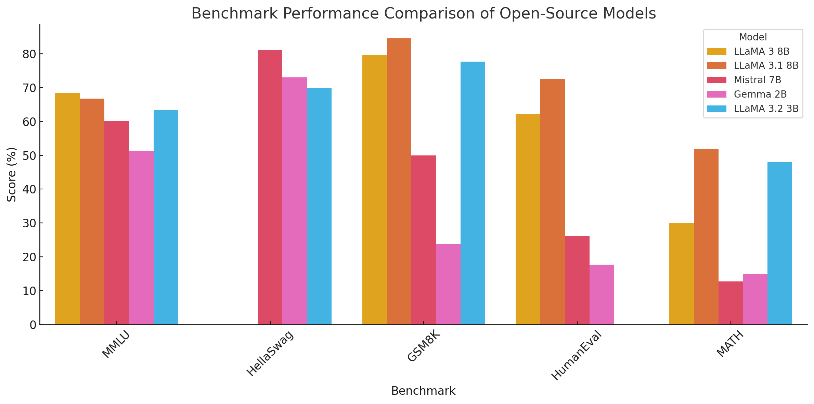
A screenshot of a computer

AI-generated content may be incorrect.

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

## 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, DeepSeek’s distilled variants, and SmolAI’s lightweight SmolVL models. 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 2**.



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

**Figure 2** 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 2.

## 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 prompt.

*A close-up of a screen

AI-generated content may be incorrect.*

*Figure 3: Redacted cell of detailed tab of prompt hub*

Along with this sheet, two presentations were crafted- one designed to show a more technical comparison between generations of models with similar architecture or release for comparison purposes, so viewers can visualize the differences between these models side by side.

## ReAct (Reason + Action) Prompting

Following 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" (arXiv:2210.03629). 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.

A close-up of a document

AI-generated content may be incorrect.

*Figure 4a: Simplified tab of prompt hub for users to see which model is*

Figure 4a 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 4b 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.

A close-up of a computer error

AI-generated content may be incorrect.

*Figure 5a: Simplified tab of prompt hub for users to see which model is*

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.

# Discussion and Conclusion

Recent studies have shown increasing interest in the use of open-source generative AI tools as alternatives to large commercial models, particularly in risk-sensitive environments. The findings of this internship strongly support those observations. 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.

# Appendix

## 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.

The mentor played a pivotal role in refining the intern's problem-solving approach, providing strategic guidance that emphasized scope awareness and practical impact. 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 the industry and in independent technical exploration.

## References

Yao S, Zhao J, Kassawat S, et al. ReAct: Synergizing reasoning and acting in language models. arXiv. https://arxiv.org/pdf/2210.03629. Published October 2022. Accessed April 22, 2025.

Prompt Hackers. Gemma 2 2B vs. LLaMA 3.2 3B model comparison. https://www.prompthackers.co/compare/gemma-2-2b/llama-3.2-3b. Accessed April 22, 2025.

Prompt Engineering Guide. https://www.promptingguide.ai/. Accessed April 22, 2025.

Lin S, Yu X, Gao Y, et al. AgentVerse: A general evaluation benchmark for large language model agents. arXiv. https://arxiv.org/abs/2504.09781. Published April 2025. Accessed April 22, 2025.

Zhang Y, Zhou K, Lin Z, et al. GROOT: A benchmark for multi-hop question answering with GPTs. arXiv. https://arxiv.org/abs/2412.01262. Published December 2024. Accessed April 22, 2025.

Meta AI. LLaMA Cookbook: Examples and utilities for running LLaMA models. GitHub. <https://github.com/meta-llama/llama-cookbook>. Accessed April 22, 2025.

Inferless. Exploring LLMs speed benchmarks: Independent analysis – Part 3. <https://www.inferless.com/learn/exploring-llms-speed-benchmarks-independent-analysis---part-3>. Published March 20, 2024. Accessed April 22, 2025.

DeepMind. Gemma: Open models built from research and aligned for safety. <https://storage.googleapis.com/deepmind-media/gemma/gemma-report.pdf>. Published February 2024. Accessed April 22, 2025.

Touvron H, Lavril T, Izacard G, et al. LLaMA 3: Open foundation and instruction models. *arXiv*. <https://arxiv.org/abs/2407.21783>. Published July 2024. Accessed April 22, 2025.

Jiang Z, Tang M, Savy T, et al. Mistral: Faster language models with better reasoning. *arXiv*. <https://arxiv.org/abs/2310.06825>. Published October 2023. Accessed April 22, 2025.