Proof-of-Concept Evaluation of RAG System

Xinyi Zhu
xinyi_zhu@ischool.berkeley.edu

Executive Summary

Through our four-week proof-of-concept (mini-POC), we found that a Retrieval-Augmented Generation (RAG) system can effectively support both engineering and marketing teams. Using LangChain, we configured and tested the system to maximize alignment with gold answers, iterating through various hyperparameters and settings. Our evaluation identified a configuration that performs best given the project's time and resource constraints. We recommend proceeding with investment in the RAG system, starting with a small pilot group to gather feedback before a full implementation. This system can streamline document search and question answering, improving productivity for both engineers and marketing staff.

1 Introduction

Our company has grown rapidly, now with 300 engineers across distributed teams in San Francisco, Austin, and remote locations, and a 40-person marketing team covering demand generation, content, product marketing, and customer education. We explored whether a Retrieval-Augmented Generation (RAG) system could improve employee productivity by answering questions and retrieving relevant information from document sources.

Our company is highly interested in rolling out new GenAI-based products. The proof-of-concept (POC) is designed to address common questions employees ask about Generative AI concepts. It uses LangChain to implement a RAG pipeline, drawing on RAG and NLP papers from ArXiv, blogs from Lily Wang on Open Domain Question Answering and related topics, and relevant Wikipedia articles. The system retrieves and synthesizes information from these document sources to provide accurate, context-relevant answers.

The POC is customized for two distinct audiences: research engineers and the marketing team. This separation is essential because the two audiences have distinct needs and seek different levels of information. For the engineering team, the use case is whether the RAG system can effectively answer technical questions about GenAI concepts, internal system architecture, and implementation details by retrieving from diverse documentation sources. For the marketing team, the use case is whether the system can quickly provide accurate, approved messaging on GenAI features, competitive positioning, and technical capabilities to accelerate content production.

2 Key Findings

1. Distinct prompts for research engineers and the marketing team improve overall accuracy and alignment with gold answers.

2. The chunking strategy significantly affects retrieval effectiveness and response quality.

3. Using multiple evaluation metrics (BERT F1, semantic similarity, LLM-as-a-judge) provides a more complete assessment of token-level precision, meaning alignment, and human-like quality.

4. Consumer-level LLMs like Cohere offer better performance and latency but come with higher costs and usage limitations.

5. Scaling considerations are important, as multiple users may access the RAG system concurrently for diverse tasks.

3 Methodology

3.1 Technical Approach

For our project, we selected the embedding model **multi-qa-mpnet-base-dot-v1**, which is specifically designed for semantic search—retrieving relevant text passages given a question or search query.

The model was trained on 215 million question-answer pairs from diverse sources and domains, including Google search queries. On the sentence transformer semantic search leaderboard, this model achieved the highest performance among semantic search models, with a leading score of 57.60, making it an ideal fit for our use case. We also tested the best general-purpose model, **all-mpnet-base-v2**, and found it performed approximately 1our Mistral pipeline.

To optimize retrieval, we experimented with document chunking strategies. Splitting documents into very small chunks (50 characters) led to fragmented, low-value information and significantly increased latency, especially at scale. With no overlap, critical information was sometimes truncated. Ultimately, we found that splitting documents into **1000-character chunks with a 100-character overlap** conserved enough context in each chunk and allowed the system to retrieve the necessary information to answer questions effectively.

Most importantly, we accommodated the two user personas by providing separate prompts for each use case. Each prompt was carefully crafted to meet the specific requirements and preferences of the engineering and marketing teams. We iteratively refined these prompts to minimize gaps between the system's responses and the gold answers, ensuring outputs were aligned with each persona's needs.

3.2 Testing and evaluation

We evaluated our RAG system using three complementary metrics, which were combined into a weighted final score for each response. This approach balances factual correctness, semantic meaning, and human-like quality, capturing multiple dimensions of answer quality. We began with a single metric and progressively added others to address limitations and ensure a more robust evaluation of the system's performance.

We used the BERT F1 score to measure token-level precision and recall against the gold answer, assigning it the highest weight (45%) because token-level accuracy is critical in question answering and retrieval. To account for sentence-level meaning and avoid penalizing semantically correct paraphrases, we used sentence transformers, weighted at 35%. Finally, to approximate human judgment, we implemented an LLM-as-a-judge, evaluating overall similarity. Given its susceptibility to hallucination, this metric received the lowest weight of 20%.

We used Gemma 2 as our LLM because it provides stable instruction following, strong semantic understanding and stable outputs, with easy access via Hugging Face and low response time. We tested multiple variants with 8-bit quantization, starting with the smallest 2B model, which proved too limited to handle long context —including the query, gold answer, and response— and tended to return repetitive "safe" scores. After testing, the 9B and 27B models showed similar judge performance, so we adopted the more efficient 9B version for robustness. We also found that removing the query from the context did not affect results, which simplified the input and reduced confusion.

For each configuration change, we first tested the Mistral pipeline on a hand-picked subset of questions, including the two project-required questions, one question with similar answers across use cases, and ten questions with significantly different answers per use case. After several major configuration changes, we evaluated the full test set to ensure incremental improvements over the prior baseline. Once the Mistral pipeline was optimized, we applied the same parameters to the Cohere pipeline for comparison.

4 Results and Findings

4.1 Proof of Concept Functionality

The POC demonstrates the core functionality of a RAG system, supporting both engineering and marketing use cases. Experimenting with different configurations of established technologies helped us identify a pipeline that best met our company's goals. This process also clarified implementation complexities, accuracy levels, and user experience considerations. Overall, evaluating RAG capabilities shows its potential to improve document search and boost employee productivity.

4.2 Lessons Learned

Through experimentation, we gained several technical insights, including the impact of the LLM choice, chunking strategy, embedding model, and evaluation metrics on overall performance. Non-technical insights revealed significant differences between the engineering and marketing teams: for the same question, each team expects very different levels of detail. Understanding these differences is crucial for tailoring prompts and ensuring the

system meets the needs of both audiences.

4.3 Challenges and Limitations

A major challenge was the free trial limits for consumer models like Cohere and GPT. Although these models are powerful, budget and usage restrictions prevented extensive experimentation, limiting our ability to fully explore their performance and scalability. We optimized the pipeline using Mistral due to Cohere's limited free trial (1,000 API calls/month, 20 calls/minute). When testing Cohere with the finalized pipeline, we adjusted for differences in response behavior, including adding delays between calls. Cohere slightly improved overall accuracy from 72Latency improved significantly based on load estimates: Mistral averaged 12s (peak 42s) per RAG response, while Cohere averaged 2s (peak 6.5s). Consumer-level LLMs offer faster responses and slightly higher accuracy, but come with higher costs. We also experimented with RAGAs using OpenAI's GPT-4 to evaluate answer correctness, faithfulness, context recall, and context precision. While answer correctness compares the response to the gold answer and faithfulness checks alignment with retrieved content, we quickly reached the free trial limit, as both metrics require LLM usage.

A major shortcoming of our model is its limited scalability. The pipeline is sequential, allowing only one user to run a single question at a time. While this worked well for our subset of test questions, processing the entire list of questions caused response times to increase dramatically, highlighting that this setup does not reflect a true multi-user production environment with employees across multiple locations. Additionally, our question-answer sets were restricted to NLP-related topics, limiting the system's exposure to diverse content. For production use, more varied question sets and document sources would be required to ensure robust performance.

A major limitation of our current system is the lack of safety controls. If extended to handle internal or proprietary documents, safety guardrails should be added to restrict access to highly confidential files. We could also block or flag inappropriate queries and monitor suspicious patterns. These measures would help reduce security and compliance risks while keeping the system useful.

4.4 Next Steps

If we had more time, we would refine the RAG prompts further, as our analysis indicates that prompt quality is the most important factor in system performance. The current prompts were based on simple heuristics derived from observing the golden answers for each use case. Although we performed several iterations of prompt tuning, more detailed and structured prompts could better align responses with user expectations. Conducting a user study with the engineering and marketing teams would help clarify their specific needs and preferred response formats. While LangChain is effective for exploratory work, its pip dependencies are complex and subject to change. For a more robust and maintainable deployment, our company should consider building custom integrations between pipeline components. If we invest in a paid Cohere Chat model tier, we should re-evaluate its capabilities—such as inference speed, accuracy, and scalability—to determine which tier best meets our production requirements.

5 Summary and Recommendations

In conclusion, we recommend building a RAG system for our company to facilitate document search and support both engineering and marketing teams. While the retrieval and evaluation pipelines can be shared, clear distinctions should be maintained between engineering and marketing during prompt tuning. Although the system requires monetary investment, it is expected to improve productivity by reducing the need for employees to manually search through large internal and external document sources. As a next step, we suggest conducting a qualitative pilot with a small pilot group (e.g., 5 engineers and 3 marketers) to gather feedback. From an architectural perspective, attention should be paid to system scalability, and deployment decisions should account for component costs and alignment with company budgets.

6 References

BERT: Devlin et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

Cohere LLM documentation. (https://docs.cohere.com)

OpenAI. (2025). ChatGPT (Version 5.1) [Large language model].(https://chat.openai.com/)

After completing the full draft of our report, we

used ChatGPT-5 to proofread, refine wording, and improve overall flow. We asked prompts such as "Does this make sense?" and "Make this flow better" to ensure clarity and readability. This aligns with expectations because we may use LLM to assist with re-writing portions of the report.

Sentence Transformers. (https://sbert.net/docs/sentence_transformer/pretrained_models.html)

A Appendix

A.1 Prompts

Table 1 shows several examples of prompts we used during development. We iterated on these prompts to address issues observed in evaluation, but additional prompt tuning is recommended to further strengthen system performance.

A.2 Error Discussion

For the best engineering and marketing examples, the model's responses closely match the gold answers in both length and content. The key concepts are captured accurately, and the overall structure aligns well with expectations.

For the worst engineering and marketing examples, we observe significant hallucination. The responses often include unnecessary or fabricated details, which we traced back to noisy retrieved context—such as references, tables, and lists—logged in our CSV outputs. In some cases, the model even replied that it could not answer the question based on the retrieved context (e.g., *"The provided context does not contain any information about a regularization term, epsilon, or a linear regression model. Therefore, I cannot answer the question based on the given context."*). These failures occurred when the retrieval system surfaced mostly noisy or incomplete chunks instead of meaningful content. Pre-processing the source documents to filter out low-quality text before chunking would help reduce irrelevant context and lower hallucination rates.

For the worst marketing examples, the responses tend to be overly long. Although they may contain only a couple of sentences, each sentence is dense with unnecessary detail. Additional prompt refinement would likely improve conciseness and focus for this use case.

In tables 2 to 5, we show the best five and worst five examples for both the engineering and marketing pipelines.

Table 1: Sample prompts used for each pipeline.

Prompt	**Research Pipeline**	**Marketing Pipeline**
Best Prompt: limits response length	You are an AI assistant for engineers, who require pretty detailed information when they ask questions. Please answer the question below only based on the context information provided. Give your response in 3-4 sentences.	You are an AI assistant for the marketing team and supporting staff, who will ask questions around GenAI in order to better understand the products and the field as a whole, but a lot more high level answers would likely be needed. Please answer the question below only based on the context information provided. Give your response in 2-3 short sentences.
Model should focus on context given	You are an AI assistant for engineers, who require pretty detailed information when they ask questions. Please answer the question below only based on the context information provided.	You are an AI assistant for the marketing team and supporting staff, who will ask questions around GenAI in order to better understand the products and the field as a whole, but a lot more high level answers would likely be needed. Please answer the question below only based on the context information provided.
Different requests for engineers and marketing teams	You are an AI assistant for engineers, who require pretty detailed information when they ask questions.	You are an AI assistant for the marketing team and supporting staff, who will ask questions around GenAI in order to better understand the products and the field as a whole, but a lot more high level answers would likely be needed.
Same basic prompt to make sure pipeline works	Please answer the question below.	Please answer the question below.

Table 2: Best 5 examples for engineering pipeline.

Query	**Response**	**Gold Answer**	**LLM as a Judge Explanation**	**Final Score**
1. How does the RoFormer model perform on semantic text matching tasks, such as those found in the CAIL2019-SCM dataset?	The RoFormer model demonstrates strong performance on semantic text matching tasks, particularly with long texts, as evidenced by its application to the CAIL2019-SCM dataset. When compared to BERT and WoBERT models, RoFormer achieves comparable results with a 512-token cutoff, slightly outperforming BERT. Notably, with a longer input length of 1024 tokens, RoFormer significantly outperforms WoBERT by an absolute improvement of 1.5%, achieving a test accuracy of 69.79%. This highlights RoFormer's ability to handle longer sequences effectively in semantic matching tasks.	Based on the results from the experiments conducted on the CAIL2019-SCM dataset, RoFormer consistently outperforms its alternatives, including BERT and WoBERT, in dealing with long texts and semantic text matching tasks. When compared with BERT and WoBERT on the same pre-training data with short text cut-offs, RoFormer shows comparable performance to WoBERT and slightly better performance than BERT. However, when the maximum input text length is increased to 1024, RoFormer significantly outperforms WoBERT by an absolute improvement of 1.5%, showcasing its effectiveness in semantic text matching tasks.	The model response accurately summarizes the key findings of RoFormer's performance on the CAIL2019-SCM dataset, including its superiority with long texts and the specific performance improvement over WoBERT.	0.8664237256668930

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
2. How does planning and reflection contribute to the iterative improvement of autonomous agents?	Planning and reflection contribute to the iterative improvement of autonomous agents by enabling them to break down complex tasks into manageable subgoals and critically evaluate past actions. Through planning, agents optimize their actions for both immediate and long-term effectiveness, while reflection allows them to identify and correct mistakes, such as inefficient trajectories or hallucinations, by incorporating self-criticism and learning from failures. These processes are integrated into the agent's working memory, providing context for future decision-making and ensuring continuous refinement of strategies over time. Together, they enhance the agent's ability to adapt and improve in dynamic environments.	Planning and reflection play a crucial role in the iterative improvement of autonomous agents by enabling them to break down tasks into manageable subgoals, learn from past actions, and refine their strategies for future steps. Through task decomposition, agents can efficiently handle complex tasks, while self-criticism and self-reflection allow them to identify mistakes and make necessary adjustments, ultimately improving the quality of their final results. By synthesizing memories and guiding future behavior based on past events, the reflection mechanism helps agents make higher-level inferences over time, leading to continuous learning and enhancement in performance.	The model response accurately captures the key points of the golden truth, highlighting the roles of planning and reflection in iterative improvement, task decomposition, self-criticism, and continuous learning. It provides a clear and concise explanation.	0.8559496328910020

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
3. What licensing terms are associated with source-available models like Mistral AI's language models?	Mistral 7B is released under the Apache 2.0 license, which is a permissive open-source license allowing free use, modification, and distribution of the software, including for commercial purposes, provided proper attribution is given. This licensing choice facilitates easy deployment and integration across various platforms and tools, such as Hugging Face, vLLM, and SkyPilot, while also supporting fine-tuning for diverse tasks. The Apache 2.0 license aligns with Mistral AI's goal of promoting affordability, efficiency, and high performance in language models for real-world applications.	Source-available models like Mistral AI's language models, including Mistral 7B, are released under the Apache 2.0 license. This license allows for more permissive use and modification of the models, providing flexibility for users to adapt the models to their specific needs. Additionally, the release of Mistral 7B includes a reference implementation for easy deployment on various cloud platforms and integration with tools like the vLLM inference server and Hugging Face for streamlined usage. This licensing approach aims to facilitate widespread adoption and adaptation of the models for different tasks and applications.	The model response accurately describes the Apache 2.0 license and its implications for Mistral 7B. It also correctly mentions relevant tools and platforms for deployment and integration.	0.842546308848343

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
4. What kind of learning challenges does the attention mechanism address in neural machine translation?	The attention mechanism in neural machine translation (NMT) addresses the challenge of memorizing long source sentences by creating customizable shortcut connections between the context vector and the entire source input, rather than relying on a single fixed-length context vector derived from the encoder's last hidden state. This design prevents the model from forgetting the initial parts of the input as it processes longer sequences. Additionally, it learns and controls the alignment between the source and target sequences, ensuring that relevant information from the source is effectively utilized for translation.	The attention mechanism in neural machine translation addresses the challenge of memorizing long source sentences by creating shortcuts between the context vector and the entire source input. This allows for more efficient learning of dependencies between source and target sequences, regardless of the in-between distance. Additionally, the attention mechanism enables the model to learn correlations between current words and previous parts of the sentence, improving the overall translation quality. By adapting different types of attention weight matrices, the model can further enhance its learning capabilities.	The model response accurately describes the function of the attention mechanism in NMT, highlighting its ability to handle long sentences and learn alignments between source and target sequences. It also correctly points out the advantage of using customizable connections over a fixed-length context vector.	0.8393191182350630

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
5. What are the main components that complement the central controller in an autonomous agent system?	In an LLM-powered autonomous agent system, the central controller (the LLM) is complemented by three key components: **Planning**, **Memory**, and **Tool Use**. Planning involves subgoal decomposition and reflection/refinement, enabling the agent to handle complex tasks and improve over time. Memory includes both short-term (in-context learning) and long-term (external vector stores for information retention). Tool use allows the agent to access external APIs for additional information, code execution, and proprietary data, enhancing its capabilities beyond the model's pre-trained knowledge.	In an autonomous agent system powered by LLM, the main components that complement the central controller include Planning, Memory, and Task Decomposition. Planning involves breaking down tasks into subgoals and reflecting on past actions for refinement. Memory allows the agent to store and retrieve information for decision-making. Task Decomposition focuses on the reliability of the natural language interface and the use of expert modules for routing inquiries. Together, these components enhance the overall functionality and efficiency of the autonomous agent system.	The model response accurately identifies the key components and provides a good explanation of their functions. It also introduces the concept of 'Tool Use', which is a relevant addition to the core components.	0.8301112489726720

Table 3: Worst 5 examples for engineering pipeline.

Query	**Response**	**Gold Answer**	**LLM as a Judge Explanation**	**Final Score**
1. What is the purpose of using a regularization term (like the epsilon multiplied by the identity matrix) in the linear regression model described in the code?	The provided context does not contain any information about a regularization term, epsilon, or a linear regression model. Therefore, I cannot answer the question based on the given context.	The purpose of using a regularization term (epsilon multiplied by the identity matrix) in the linear regression model described in the code is to prevent overfitting by adding a penalty term to the loss function. This regularization term helps to control the complexity of the model by discouraging large coefficients, leading to a more stable and generalizable solution. By incorporating this regularization, the model can better handle noise and outliers in the data, ultimately improving its performance on unseen data points.	The model incorrectly states that the context lacks information about regularization. The golden truth clearly describes the purpose of regularization in the context of a linear regression model.	0.44812042641359900

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
2. Can you name some notable large language models and their respective creators or companies?	Based on the provided context, some notable large language models and their creators or companies include: **BloombergGPT** by Bloomberg (Wu et al., 2023), **ChatLaw** by J. Cui et al. (2023), and **Toolformer** by T. Schick et al. (2023). Additionally, **Chain of Thought prompting** was introduced by J. Wei et al. (2022) to elicit reasoning in large language models, and **Re³** (Recursive Reprompting and Revision) was developed by K. Yang et al. (2022) for generating longer stories. These models highlight advancements in specialized and general-purpose language model applications.	Some notable large language models include Mistral 7B developed by a team of engineers including Albert Q. Jiang and Guillaume Lample, as well as Chinchilla developed by the research team at DeepMind. Additionally, The Pile dataset was created by Leo Gao, Stella Biderman, and others, while the Realm model was developed by Kelvin Guu and his team. These models have been recognized for their superior performance and efficiency in language generation and natural language processing tasks.	The model response does not mention any of the large language models listed in the golden truth. It instead provides examples of other models and techniques.	0.4733161951001840

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
3. What is the significant enhancement in Claude 2.1 compared to its previous version?	The provided context does not contain any information about Claude 2.1 or its enhancements compared to previous versions. Therefore, I cannot answer the question based on the given context.	The significant enhancement in Claude 2.1 compared to its previous version Claude 2 is the expanded context window, which has been increased from 100,000 tokens to 200,000 tokens. This allows Claude 2.1 to process and analyze larger amounts of text, enabling it to provide more comprehensive summaries and assistance with tasks. Additionally, Claude 2.1 has improved its performance in handling complex queries and requests, showcasing advancements in its capabilities for natural language processing and understanding.	The model incorrectly states it cannot answer the question due to lack of information, when the golden truth is provided in the prompt.	0.49971322063980200

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
4. Who proposed the first significant statistical language model?	The context provided does not contain information about who proposed the first significant statistical language model. The references listed focus on recent advancements in large language models, multimodal reasoning, and related topics, but do not address the historical development or early contributions to statistical language modeling. Therefore, based on the given context, it is not possible to answer the question.	The first significant statistical language model was proposed in 1980. This model was a probabilistic model of natural language. IBM conducted 'Shannon-style' experiments during the 1980s to identify potential sources for language modeling improvement by observing and analyzing the performance of human subjects in predicting or correcting text.	The model fails to answer the question despite having the information in the golden truth. It incorrectly claims the context lacks information.	0.5365552137354210

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
5. What are some of the limitations of traditional position encoding methods in the architecture of pre-trained language models (PLMs), and what novel approach does the paper propose to address these issues?	Traditional position encoding methods in pre-trained language models (PLMs) face limitations, particularly in their incompatibility with linear self-attention mechanisms, as highlighted in the context. Absolute position encoding, while effective in some cases, does not inherently capture relative positional information, which is crucial for understanding the sequential order of words. Relative position encoding approaches, on the other hand, often struggle with efficiency and scalability, especially for long texts. To address these issues, the paper introduces **Rotary Position Encoding (RoPE)**, a novel method that uses rotation matrices to encode position information. RoPE decays with increasing relative distance, aligning well with natural language encoding needs, and is shown to outperform traditional methods in experiments on long text benchmarks, as demonstrated by the proposed **RoFormer** model.	One limitation of traditional position encoding methods in PLMs is that they may not enable length extrapolation of pre-existing models, leading to the need for substantial pre-training costs. The paper proposes a novel approach called Position Interpolation, which extends existing PLMs without deviating far from existing definitions of position encoding or attention mechanisms. This method allows for much extended context windows for text modeling, leading to significant perplexity gains and improved model performance.	The model response focuses on Rotary Position Encoding (RoPE) instead of the Position Interpolation method described in the golden truth.	0.5552477544172330

Table 4: Best 5 examples for marketing pipeline.

Query	**Response**	**Gold Answer**	**LLM as a Judge Explanation**	**Final Score**
1. How does the uniqueness of a reparameterized reward function within an equivalence class impact the selection of optimal policies in constrained reinforcement learning problems?	The uniqueness of a reparameterized reward function within an equivalence class ensures that all reward functions in the same class induce the same optimal policy in constrained reinforcement learning problems. This means that despite variations in the reward functions, the optimal policy remains consistent, simplifying policy selection and ensuring robustness in the learning process.	The uniqueness of a reparameterized reward function within an equivalence class ensures that different reward functions from the same class will lead to the selection of the same optimal policy in constrained reinforcement learning problems.	The model response accurately paraphrases the golden truth, correctly explaining the concept of equivalence classes and their impact on policy selection in constrained reinforcement learning.	0.9257242933524510
2. What are some of the concerns related to fine-tuning QA models with common datasets?	One concern with fine-tuning QA models using common datasets is the significant overlap between questions in the train and test sets, which can lead to inflated performance metrics. Additionally, models may still generate toxic, biased, or factually incorrect outputs, even after fine-tuning, highlighting ongoing challenges in alignment and safety.	One concern of fine-tuning QA models with common datasets is the significant overlap between questions in the training and test sets, which can impact the model's performance.	The model response accurately identifies the concern of data overlap and its impact on performance. It also adds valuable points about potential model outputs even after fine-tuning.	0.8870985315158200

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
3. How does planning and reflection contribute to the iterative improvement of autonomous agents?	Planning and reflection contribute to the iterative improvement of autonomous agents by breaking down complex tasks into manageable subgoals, optimizing actions for efficiency, and identifying inefficiencies or hallucinations. Reflection allows agents to self-critique past actions, learn from mistakes, and refine future decisions, while planning ensures actions are aligned with both immediate and long-term goals. Together, they enable agents to adapt and improve over time, enhancing performance in real-world tasks.	Planning and reflection contribute to the iterative improvement of autonomous agents by allowing them to break down tasks into manageable subgoals, learn from past actions, correct mistakes, and refine their approach for future steps. This process leads to better decision-making, higher-quality results, and overall improvement in performance over time.	The model response accurately captures the key benefits of planning and reflection for autonomous agents, including task decomposition, learning from mistakes, and goal alignment. It also highlights the iterative nature of improvement.	0.8717052423962970

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
4. How does the release of successive models in a language model series typically improve functionality?	The release of successive models in a language model series typically improves functionality by increasing model size, which often leads to emergent capabilities and better performance. Additionally, these models benefit from refined training strategies, such as optimized learning rate schedules and hyperparameter tuning, to enhance efficiency and reduce loss. These advancements collectively contribute to more robust and versatile applications in real-world scenarios.	The release of successive models in a language model series typically improves functionality by enhancing adaptability, performance, and efficiency through advancements in training methods and model architecture.	The model response accurately captures the essence of model improvement in language model series, highlighting model size, training strategies, and real-world applications.	0.8668781164106230
5. What types of memory are leveraged in autonomous agents, and how do they differ?	Autonomous agents leverage **short-term memory** and **long-term memory**. Short-term memory refers to in-context learning, which is finite and limited by the model's context window, while long-term memory involves external vector stores for infinite retention and fast retrieval of information over extended periods.	Short-term memory and long-term memory are utilized in autonomous agents. Short-term memory is used for in-context learning, while long-term memory allows agents to retain and recall information over extended periods. Long-term memory often leverages external storage for fast retrieval.	The model accurately describes the roles of short-term and long-term memory in autonomous agents. It correctly links short-term memory to in-context learning and long-term memory to external storage for persistent information.	0.8635894908866950

Table 5: Worst 5 examples for marketing pipeline.

Query	**Response**	**Gold Answer**	**LLM as a Judge Explanation**	**Final Score**
1. What are the implications of fine-tuning large language models on datasets of varying sizes in contrast to using few-shot learning techniques?	Fine-tuning large language models on larger datasets significantly improves performance compared to few-shot learning, even on both small and large datasets. However, scaling to larger datasets requires high-quality data and careful management of ethical and privacy concerns, such as toxic language and private information. This highlights the need for responsible dataset collection and introspection to ensure model effectiveness and integrity.	Fine-tuning with DPO surpasses PPO-based RLHF in sentiment control across generations and improves response quality in summarization and single-turn dialogue while being simpler to implement and train.	The model response does not address the comparison between DPO and PPO-based RLHF for sentiment control or other aspects mentioned in the golden truth.	0.4203192778431650
2. What is the significant enhancement in Claude 2.1 compared to its previous version?	The context provided does not contain any information about Claude 2.1 or its enhancements compared to previous versions. Therefore, I cannot answer the question based on the given context.	The significant enhancement in Claude 2.1 compared to its previous version is the expansion of the context window from 9,000 tokens to 100,000 tokens, as well as the ability to upload PDFs and other documents for reading, summarizing, and task assistance.	The model failed to identify the relevant information within the provided context and instead stated it lacked information.	0.5172540260865800

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
3. What approaches or methods are suggested for improving the alignment of language models with human preferences?	The paper suggests using reinforcement learning from human feedback (RLHF) to fine-tune language models, specifically GPT-3, to align with human preferences. This involves hiring labelers to provide feedback on model outputs, training a reward model to predict preferred outputs, and fine-tuning the model to act in accordance with user intentions, including being helpful, honest, and harmless.	Direct Preference Optimization (DPO) is a suggested approach for improving the alignment of language models with human preferences. It optimizes the same objective as existing reinforcement learning with a KL-divergence constraint but is simpler to implement and train.	The model response describes RLHF, a different technique for aligning language models, not DPO.	0.5599730147692330
4. What is the purpose of using a regularization term (like the epsilon multiplied by the identity matrix) in the linear regression model described in the code?	The provided context does not mention a regularization term or linear regression model, so I cannot answer the question based on the given information. Regularization terms are generally used to prevent overfitting by penalizing large coefficients, but this is not discussed in the context provided.	The purpose of using a regularization term in the linear regression model is to mitigate or resolve catastrophic extrapolation error, making the extrapolated values comparable to those within the original range.	The model failed to answer the question as it claims the context lacks information about regularization. However, the question directly asks about regularization in linear regression.	0.5615724708799800

Query	Response	Gold Answer	LLM as a Judge Explanation	Final Score
5. What methods or approaches are being investigated or utilized to optimize or adapt machine learning models?	Several methods are being explored to optimize and adapt machine learning models, including low-rank adaptation (LoRA) for efficient fine-tuning of large language models, instruction tuning for specific tasks like long text generation, and retrieval-augmented generation (RAG) to enhance model performance with external knowledge. Additionally, techniques like active retrieval and parameter-efficient transfer learning are being investigated to improve model efficiency and generalization.	The methods or approaches being investigated or utilized include large-scale pre-training on general domain data and adaptation to particular tasks or domains, adding adapter layers, and optimizing some forms of adaptations.	The model response focuses on parameter-efficient fine-tuning techniques rather than the core methods mentioned in the golden truth, which emphasize pre-training and adaptation strategies.	0.5701734430869260