Name: Ashutosh Kumar Jha
Degree: BTech
Year: 2nd
Sem: 4th
Branch: DSAI
Course: DAV
ID: 12340390

1. Baseline Evaluation

We started by loading the **NTSE_QA.csv** dataset and running the benchmark using the model **Gemma 3:4B** with a fixed, unmodified prompt in order to get the baseline evaluation of the model. This initial experiment was crucial in establishing a **baseline accuracy** which came around **48%**. Understanding this baseline is important because it provides a **reference point** for evaluating future improvements. By assessing how the model performs “out of the box” without any modifications, we gain insight into its raw capabilities. Any enhancements or fine-tuning efforts can then be measured against this benchmark to determine their effectiveness. This step sets the foundation for optimizing the model further, allowing us to track progress systematically and make data-driven improvements.

2. Improving Performance

2.1. Prompt Engineering

To enhance the model's performance, we refined the **system prompt** to be more **concise, clear, and structured**, ensuring it reinforces the strict output format. Additionally, we introduced **few-shot examples** when necessary to provide better guidance. These refinements play a crucial role in improving accuracy. A **clearer prompt** minimizes ambiguity, reducing the chances of the model misinterpreting instructions. By explicitly emphasizing the **required JSON format**, we ensure **output consistency**, preventing unnecessary text from interfering with answer extraction. Furthermore, even a **few well-chosen examples** can significantly help the model

understand what constitutes a “correct answer,” making its responses more precise and aligned with expectations. These adjustments create a more structured and effective interaction, ultimately boosting overall accuracy.

2.2. Retrieval-Augmented Generation (RAG)

To further improve accuracy, we implemented a **retrieval mechanism** that fetches relevant context from a **locally curated knowledge base** and appends the most pertinent snippets to the prompt. This ensures that the model has access to additional, **domain-specific information**, rather than relying solely on its **pre-trained knowledge**. This approach significantly enhances performance in multiple ways. First, by providing **enhanced context**, the model can generate more **accurate and informed responses** based on real, relevant data. Second, it helps in the **reduction of hallucination**, as the model now has direct evidence to support its answers, minimizing the chances of incorrect guesses or fabricated information. Lastly, by retrieving and appending only the **most relevant snippets**, we ensure that the model receives **targeted information** without unnecessary noise, making the responses both precise and contextually appropriate. This strategic retrieval process makes the model more reliable and effective in handling knowledge-based queries.

2.3. LLM Agents (Reasoning Agent)

To further enhance accuracy, we introduced a **second-step validation process** using a **reasoning agent** that reviews the initial answer produced by the primary model. This agent acts as a **verification layer**, ensuring that the response is accurate and well-reasoned before being finalized. This approach offers several key benefits. First, it adds a **double-checking mechanism**, much like having a second pair of eyes to catch potential errors or inconsistencies that the initial model pass might have overlooked. Second, the reasoning agent enables **refinement**, meaning that if the primary response is incorrect or incomplete, the agent can adjust and improve the answer based on both the **original query** and the **candidate response**. Finally, this multi-step process increases **robustness**, as it forces the system to be more deliberate and thoughtful, reducing the likelihood of impulsive or inaccurate responses. By implementing this verification step, we significantly improve the reliability and precision of the model’s output.

2.4. Self-Consistency Decoding

To further improve accuracy, we implemented a **multi-response generation approach**, where the model produces multiple answers for the same question, and the final

response is selected based on **frequency or confidence levels**. This method enhances reliability in several ways. First, it introduces a **diversity of thought**, reducing the impact of a single outlier response that might be incorrect. By considering multiple outputs, we increase the chances of arriving at a well-reasoned answer. Second, using a **consensus mechanism**, such as majority voting or confidence-based selection, allows us to choose the most **statistically reliable** answer, rather than relying on just one model pass. Finally, this approach helps with **error averaging**, smoothing out randomness and mitigating inconsistencies that might arise from minor fluctuations in model behavior. By integrating this selection strategy, we improve the stability and overall accuracy of the system's responses.

2.5. Chain-of-Thought (CoT) Prompting

To further enhance accuracy, we incorporated **internal step-by-step reasoning**, often referred to as the **chain-of-thought (CoT) approach**. This method encourages the model to **think through** each step of the problem internally before arriving at a final answer, even if the reasoning itself is not explicitly displayed in the output. This strategy improves accuracy in several ways. First, by following a **step-by-step analysis**, the model reduces the likelihood of errors caused by hasty conclusions. It carefully evaluates each aspect of the question, leading to more **deliberate and accurate** responses. Second, this approach is especially useful for **complex problem-solving**, where breaking down a problem into smaller logical steps ensures that **no critical detail is overlooked**. Lastly, by forcing the model to **validate its own reasoning**, we enhance **logical consistency**, reducing contradictions and improving overall answer reliability. This hidden but structured thought process makes the model's responses more accurate and well-reasoned.

3. Comparing Results

Improving the accuracy of a language model requires a systematic approach, with each technique addressing specific weaknesses in the baseline system. We began with a simple, unmodified prompt to establish an initial accuracy benchmark. From there, we applied a series of enhancements—ranging from prompt engineering and retrieval-augmented generation (RAG) to advanced techniques like reasoning agents, self-consistency decoding, and chain-of-thought prompting. Each of these refinements incrementally improved performance by clarifying instructions, reducing hallucinations, reinforcing logical consistency, or leveraging consensus mechanisms. The results of our step-by-step optimizations are summarized in the table below.

Accuracy by different techniques

Technique	Accuracy	Change in accuracy	Increment / Decrement
Baseline (Unmodified prompt)	48.00%	-	-
Prompt Engineering (Improved clarity, structured examples)	49.00%	1%	Increment
Retrieval-Augmented Generation (RAG) (External knowledge integration)	52.00%	3%	Increment
LLM Agents (Reasoning Agent) (Secondary verification step)	53.00%	1%	Increment
Self-Consistency Decoding (Multiple responses, consensus-based selection)	53.00%	0% (nearly same)	Increment
Chain-of-Thought Prompting (Step-by-step internal reasoning)	49.00%	4%	Decrement
Best Combined Approach (Optimal mix of all techniques)	54.00%	-	-

These results highlight how targeted refinements can significantly enhance model accuracy. While prompt engineering provides a strong initial boost, further gains come from supplying external context (RAG), refining answers through verification (LLM agents), and improving logical consistency (self-consistency decoding and chain-of-thought prompting). Interestingly, combining all these techniques yielded the highest accuracy, reinforcing the idea that no single method is a silver bullet—rather, a layered, complementary approach is key to optimizing performance.

Conclusion

Our approach demonstrates that optimizing a language model’s accuracy is not about a single breakthrough but rather a series of deliberate, complementary enhancements. Each technique—whether it be refining the prompt, integrating external knowledge, adding verification layers, or leveraging structured reasoning—addresses a specific weakness in the baseline system. By systematically applying and combining these methods, we were able to push the model's performance far beyond its initial capabilities.

The key takeaway is that no single strategy guarantees optimal results on its own. Instead, the most effective solution arises from a hybrid approach that balances clarity, context, verification, and reasoning. Moving forward, additional fine-tuning and domain-specific adaptations can further enhance accuracy, ensuring that the model continues to improve as it encounters more complex and nuanced queries.