Limitations of the Cohere Text Completion Application

The Cohere model demonstrates strong capabilities in several areas, particularly in creative writing and simplified explanations. It performs well when generating stories from imaginative prompts such as “Once upon a time, there was a robot who...”, producing coherent and engaging narratives with a clear structure. Additionally, the model excels at breaking down complex ideas into simple language, making it ideal for prompts like “Explain recursion like I’m five.” Its conversational tone makes the responses accessible and enjoyable to read, and it is also fairly effective at summarizing general content in a concise manner.

However, the model has notable limitations. It tends to struggle with logical reasoning and multi-step analytical tasks. Prompts that require deductive thinking or abstract reasoning—such as those involving syllogisms or technical logic—often result in vague or incorrect answers. Additionally, the model may include fabricated or outdated facts in its responses, which becomes especially problematic when users expect accurate or current information. It also has difficulty with very specific or niche academic topics, often returning generic statements. Another limitation is its occasional refusal to continue prompts that imply human experiences or traits, as seen when the model declined to finish a fictional story involving a robot with emotions, due to ethical safety settings.

To improve the application, several enhancements can be considered. First, offering users categorized prompt templates could guide more effective interactions with the model by aligning user expectations with the model’s strengths. Second, adding post-processing filters to flag excessively long outputs, repetitive responses, or hallucinated information could improve the overall quality of generated content. For fact-based tasks, integrating a lightweight validation tool, such as a reference check, could help verify factual accuracy. Additionally, tuning model parameters—such as adjusting temperature or max tokens based on the task type—would allow for better control over creativity and response depth. Logging user inputs and model outputs can also be beneficial for reviewing performance and identifying common failure patterns. Lastly, implementing an optional user feedback feature (e.g., thumbs up/down) could provide valuable insights into how the model is being used and which areas need refinement.