

Employing GenAI Tools in the Complete Process

In this skills evaluation test, I applied the various features of Generative AI (GenAI) tools throughout the lifecycle of the project, right from defining the problem to writing code and scripting the final report. The following steps provide more details on the use of GenAI at each stage of the process:

Problem Understanding

Objective:

To come up with a complication that emerges out of the ‘order dependency problem’ of large language models, particularly focusing on MCQs.

Use of GenAI:

Initial Research: For the initial stage of understanding the order dependency problem, I harnessed LLMs (such as GPT-4), as well as other literature and definition searching tools. On that prompt, the model was able to lay out how the issue is seen in forms other than MCQ’s such as customer service chatbots, decision support systems and recommendation systems when I used the model’s capability to explain order dependency in other contexts.

Example Requests: I also requested the LLM model to help in formulating some simplified non-technical examples of order dependency that reinforced the problem to be solved before elaborating the technical solutions.

2. Code Generation and Development

Objective:

To design an application which attempts to investigate the sensitivity of LLMs for the order of inputs in MCQs and the permutations of answer options, to form the queries to an LLM and to analyze the responses.

Use of GenAI:

Code Snippets Generation: During the development process, I was using GenAI to write small and relatively independent sections of code including:

Combining all the options of the MCQ using permutations module of Python’s itertools.

Example API calls for the purpose of asking an LLM (for instance, the OpenAI’s GPT-4).

JSON and CSV output planning for Reporting LLM Response Error:

Using such prompts as ‘Produce a Python function to create all permutations of an array of options’, I got brief and utilitarian code snippets that helped in the developmental period.

Error Handling and Debugging: At the moments when I had to firefight with errors or optimize the multifaceted logic in the code, I employed GenAI to show me how to deal with the problems. For instance, in dealing with numerous API calls at once, or the creation of the Streamlit UI interface.

API Documentation Assistance: By using GenAI I was able to make OpenAI API call examples to query the model easily without again browsing through the documentation.

3. Report Writing and Documentation

Objective:

In order to write adequate documentation involving the README file as well as writing a report on observations and findings concerning order dependencies.

Use of GenAI:

README and Comments: GenAI was utilized to explain the structure of the README file, provide guidance on how to execute the application and possible sections to add: Installation, Usage and Evaluation Methods.

I made specific requests of GenAI such as “Write instructions for running a Streamlit application” and this yielded well formatted text that I customized to creating the final README.

Explaining Findings: While looking into results of the application, I utilized GenAI for the purpose of summarizing certain aspects of order dependency. For example, I provided the model with the instruction to write explanations concerning dissimilarities in LLM responses depending on the ordering of MCQ options.

Evaluation Metrics: GenAI helped in recommending possible evaluation metrics such as the “Discrepancy Rate” and “Bias Detection.” It offered ideas of how best to approach the quantification of the LLM responses when considering order sensitivity, although the ideas were refined and used in the project.

4. Final Report

Objective:

To deliver a well-coordinated and understandable impression regarding the application’s concept of solving the order dependency problem and outcomes of the assessment.

Use of GenAI:

Generating Structured Responses: When writing the final report, I employed GenAI in the sense where the tool helped me to structure my thinking properly. For example, writing prompts like, Explain how this application demonstrates the order dependency problem in LLMs helped to generate a draft of what I wrote as the final report.

Patterns and Insights: In the course of the analysis, I asked GenAI to assist in looking for patterns that may arise from the differences in the MCQ responses (for example, whether or not the LLM had a preference for specific locations in the option list). It helped me in analyzing the results to come up with the final report thus the curriculum played a vital role in my preparation.

5. Reflection on the Use of GenAI

GenAI is usually defined as a system which uses Artificial Intelligence approaches to improve genetics and help humanity live better.

GenAI tools acted as an invaluable companion throughout the process, helping to:

Quickly brainstorm the problem while giving easy to understand illustrations.

Write, learn errors, and optimise real or hypothetical code fragments, which would take less time to develop.

Give need an writing help in creating documentation writing, instructions as well as the content of final report.

This was followed by some manual fine tuning of the generated code and text where fineness had to be fine tuned to the particular need of this project despite the excellent work done by GenAI in automating the process.

Incorporating of GenAI in the particular course of the work enabled me demonstrate upper level of thinking in aspects of the design and analysis phases of the project, as the major chunk of the coding and drafting of the reports and several other documents could be generated by GenAI automatically. Equally blending human innovation and Artificial Intelligence when developing an efficient and effective solution to the order dependency issue will remain a key strength of the programme.