Michelin Project Discussion with Dr. Ehrett

By Troy Butler

On October 17, 2024, fellow team member Jacob Cox and I met with Dr. Carl Ehrett of the Watt AI Creative Inquiry program to discuss LLM implementation.

Our conversation began with a high-level overview of the Capstone-Michelin project and definition of the project goal, which is to extract business rules from legacy COBOL code and create a generative AI interface capable of answering natural-language queries about those business rules. Our current choices of LLaMA 3 for our model due to its open-source nature and our inclination toward using a RAG pipeline were expressed to Dr. Ehrett.

Dr. Ehrett stressed upfront that retrieving the correct chunks of COBOL code in response to a natural language query would be the hardest part. Since the datastore used in the RAG pipeline will involve code and the queries are taking in human language, there is an inherent difference. Dr. Ehrett suggested that we introduce an intermediate step – ask the LLM to give examples of code that could implement a particular business rule, and then use those results to form the underlying query for the final output. Dr. Ehrett also mentioned that there are libraries that handle a lot of the components of RAG in an automated way, although those libraries are usually aimed more at natural language datastores rather than code-based ones. With that said, he didn't know that it would make a noticeable difference given how we plan to implement RAG.

Dr. Ehrett also stressed that we should take the time to set up some automated test suites to evaluate how well the LLM succeeds when queried, i.e., "for this query, here are the chunks of code that should be retrieved." We ultimately want to be able to measure how well it succeeds against human-generated cases.

When asked about evaluating retrieval of individual tokens, Dr. Ehrett mentioned that the most straightforward way is human evaluation, but also suggested that we use a smarter LLM to grade the output, i.e., "here's the user query – does this response answer the user's question?" We could even give the smarter model the query along with big chunks of COBOL code and ask if the response looks correct. This would obviously be a less reliable way of evaluating, but easier in other respects.

On the subject of incorporating more than one LLM into our pipeline, Dr. Ehrett explained that it mostly boils down to computational (and potentially financial) expenses. We ideally want to use the smartest LLM we can – there is no virtue in having diversity in the LLM's we use.

With respect to our choice of LLaMa 3 as our model, Dr. Ehrett mentioned that some models are better than others at code generation. He said that in the case of this project, we don't intend to produce perfect code, so it may not matter if the LLM is a little buggy in that regard. It likely writes and reads code well enough for what we need, but this is something we will need to evaluate.

When asked about the potential use of guardrailing in our pipeline, Dr. Ehrett stated that in past projects, he has mostly relied on model selection as a way of dictating the quality of output as different models have different strengths and weaknesses. Since we are ultimately considering an AI tool for employee use (i.e. not customer-facing), some safety concerns are not as relevant. One safety measure that could be easily implemented, however, is to add an additional LLM step that queries the output to make sure it is "safe", i.e., does not contain false or discriminatory information. If any red flags are detected, the output would be intercepted and not delivered.

On the subject of tailoring the complexity of generated output based on the original query (i.e. to generate different complexities of response based on the technical knowledge of the user), Dr. Ehrett suggested that we consider making the response complexity an explicit part of the prompt. He described “Few-shot” learning (FSL), and explained that when we query the LLM, in addition to the prompt, we show it a series of prompts with correct responses as this would improve the quality of the model’s output. He suggested writing a number of cases of “here’s a prompt and a really good answer” while also specifying the concern of the technical level in the prompt, e.g., low-technical users vs high-technical users.

At this point, our meeting ended with Jacob and I thanking Dr. Ehrett for his time, and Dr. Ehrett expressing his willingness to provide more help as needed as we delve deeper into the project.