**New LLM Use Cases and Techniques for Broader Automation Scope**

*By: Michael Larsen*

I recently had the privilege of presenting at SDFM’s DA/DS 2024 conference where my talk, *Beyond Chatbots*, sought to advance the conversation around Large Language Models (LLMs). Using budget justification drafting as an example, I highlighted the emerging approaches to LLM development that can solve complex challenges beyond the usual applications like ChatGPT or auto-summarization in search engines.

At cBEYONData, I’ve been spearheading the development of *Guru*, a retrieval-augmented generation (RAG) tool designed to perform question-answering tasks based on an organization's documents. While tools like ChatGPT and Guru represent transformative deployments of LLMs, they only scratch the surface of what’s possible. Budget justification drafting provided an opportunity to expand the boundaries of automation and show how bespoke LLM solutions can be created to meet our client’s most pressing challenges.

**Budget Justification Use Case**

These reports often include:

* Editorial Content: Explaining changes in prioritization or unforeseen challenges.
* Factual Content: Mission statements, financial figures, charts and textual summaries of those charts and tables.

By creating a custom LLM solution, it is possible to automate a portion of that factual content and generate high-quality drafts. Let’s review the RAG method and explore how advanced reasoning techniques like Chain of Thought (CoT) and Question Decomposition play a role in development.

**Retrieval Augmented Generation (RAG)**

RAG combines LLMs with a retriever that selects relevant document segments to answer specific questions. This is the dominant approach for LLM development as it greatly increases accuracy and reduces hallucination by relying on a specific document library, instead of a model’s training data. The process starts by using all the relevant financial tables and documents that will need to be referenced to write the new draft.

**1. Chain of Thought (CoT) Reasoning**

COT reasoning breaks down complex tasks into a step-by-step process, mimicking how humans approach problems. By replicating these steps in an LLM pipeline, a model can be leveraged at each step with the results used as inputs for the next phase. This enables a tool to:

* Follow logical or branching sequences.
* Handle predictive analytics tasks.

This structured approach allows LLMs to tackle more complex use cases, from narrative drafting to data-driven insights.

**2. Question Decomposition**

Question decomposition involves breaking a complex paragraph into discrete, answerable questions. Consider this sample:

**2024 Budget Text:** “Under current law, USDA’s total outlays for 2024 are estimated at $228.3 billion. Outlays for mandatory programs are $191.5 billion, 83.9 percent of total outlays. Mandatory programs provide services …”

By decomposing this paragraph, we generate targeted questions:

* What are the total outlays for USDA in 2024?
* What portion of outlays is for mandatory programs?

Building question decomposition into RAG deployments allows LLMs to handle complex queries requiring multi-step reasoning or comparisons. Answering in isolation, the right documents can be found to answer each question. Structured responses (e.g., lists in Python) can further enhance automation pipelines.

**Overcoming Challenges with Tabular Data – Table Serialization**

Tabular data, critical for financial reports, poses unique challenges for LLMs due to its low semantic value. Retrievers in RAG pipelines may struggle to reason against and match such data. To address this:

Serialization involves converting tabular data into descriptive text. By asking an LLM to generate textual representations during the document indexing step, we:

* Add semantic value for better matching and reasoning.
* Prepare data for anticipated queries, reducing errors at runtime.

For example, a table showing budget allocations could be serialized as: “Table showing USDA’s 2024 budget breakdown by program type, with mandatory programs accounting for $191.5 billion (83.9%).” This is partly possible because the solution is tailor-made to the use case. Knowledge of the type of documents and the questions or challenges the tool will face informs development and improves the results above and beyond what an off the shelf solution would provide.

**Integrating CoT Reasoning and Tool Use**

Advanced reasoning techniques can also enhance LLM capabilities by incorporating external tools. For instance:

* Using an LLM to extract the arguments and operators for the equation.
* Pipeline passes data to a programmed function to perform the calculations.

This approach, often referred to as “agent-based LLMs,” combines CoT reasoning with tool integration to address LLM limitations, enabling sophisticated solutions for automation workflows.

**The Path Ahead**

LLMs are poised to transform automation across industries, from streamlining budget justifications to tackling other high-value tasks. By combining RAG with advanced techniques like CoT reasoning, question decomposition, and table serialization, we can unlock new possibilities for solving complex challenges. For the budget justification example, these approaches allowed for the creation of draft portions that matched what was ultimately published.

I encourage my colleagues and clients to think beyond the surface-level applications of LLMs. With the right techniques, these models can drive meaningful innovation and deliver tailored solutions for even the most intricate problems. Let’s continue pushing the boundaries of what’s possible in automation!

What other use cases do you envision for LLMs in automation?