**Financial Q&A System Report**

1. **Method Overview**

## **Original thinking**

With the assignment's requirement to design an LLM-driven prototype that can answer questions based on financial documents, my initial approach was to use a Retrieval-Augmented Generation (RAG) system. In this approach, the LLM would examine the train.json file, perform a semantic search for the most contextually relevant Q&A pair, and respond with the most appropriate answer.

However, the task's core purpose was for the LLM to calculate the correct answer, not simply retrieve it. This led me to consider creating an agentive loop with tools and an appropriate formula library to address various question types. A key objective was to develop a feedback mechanism where the LLM could assess its own working and validate the sensibility of its generated answer.

Initially, I contemplated adding validation to ensure the LLM only generates and utilises mathematically valid formulas. The primary challenge was the lack of a comprehensive list of question types (such as percentage change, difference, Return on Investment, etc.). Without a pre-defined list of formulas, there was a risk of being unable to answer questions that fell outside of this.

## **Methodology**

I landed on creating a more general agent with the following approach:

1. Extract context information of table and information before and after the table
2. Pass contextual information to the LLM along with the question
3. Request a detailed answer with:
   * Reasoning steps
   * Relevant data points
   * Calculation formula
   * Potential validation checks
   * Calculated answer
   * Confidence level
4. Implement a self-validation mechanism for the LLM to assess the answer’s reliability

*NOTE: I used the first 10 entries of the dataset due to time constraints of running the model.*

1. **Accuracy Metrics**

## **Rationale for Metric Selection**

When thinking about what accuracy metrics to use, I kept in mind that this is a mathematical financial question-answering system therefore the metrics used would have to account for this.

## **Exact Match Accuracy**

This metric calculates the percentage of calculated answers that matched the actual answer exactly, representing the LLM’s precision in calculating correct answers. This is useful for scenarios where we need extreme precision, such as financial reporting. The limitations with this are that it is highly sensitive to minor numerical differences and penalises answers that are mathematically correct but numerically different.

## **Fuzzy Match Accuracy**

Here I introduced a tolerance to account for reasonable numerical variations, since we must recognise that small differences may be acceptable in some financial contexts. I chose to implement a 10% tolerance to balance precision and flexibility. This metric is particularly useful for percentage calculations and estimates where high levels of precision are less important.

## **Validation Confidence**

I leveraged the LLMs capability to perform a self-assessment and provide insight into the models own internal reasoning ability. This could help to identify some potential biases as well as offer an assessment of answer reliability. Since we have the capability of an LLM, it made sense to utilise this in addition to the more traditional accuracy metrics.

1. **Performance Metrics**

## **Accuracy Metrics Summary *(out of sample of 12)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Mean** | **Success Count** | **Success Percentage** |
| Exact Match Accuracy | 0.0833 | 1 | 8% |
| Fuzzy Match Accuracy | 0.5833 | 7 | 58% |
| Validation Confidence | 0.4167 | 5 | 42% |

1. **Detailed Metric Analysis**

## **Exact Match Accuracy**

In a sample of 12 questions, we can see that only 1 was an exact match. The limitation with this method is the extreme sensitivity to precision, for example if the actual answer recorded in the dataset was 1.3% and the answer that the LLM calculated was 1.27%, this would count as a false exact match, despite being mathematically close.

## **Fuzzy Match Accuracy**

This method is more forgiving than exact match as shown by the success percentage which has increased from 8% for exact match accuracy to 58% for fuzzy match accuracy. This provides a more realistic assessment of the accuracy of the LLM.

## **Validation Confidence**

The final metric I calculated was the proportion of calculated answers that were marked as ‘High’ validity by the self-validation, where it deemed that 42% of answers were highly reliable. This represents the LLMs internal confidence in its reasoning and answer reliability when assessing the question again.

1. **Limitations of Current Approach**

* The same LLM that is calculating the answer is also performing the validation which can create a closed loop of reasoning that can potentially introduce bias. This means that the LLM might inadvertently agree to its own incorrect reasoning which makes it difficult to detect and misunderstandings or errors in the calculation.
* For the self-validation assessment, we ask the LLM to categorise into ‘High/Medium/Low’ which is very subjective. There are no standard criteria, and the outcome will vary depending on the LLM and it’s training.

1. **Proposed Improvements**

* We could develop a comprehensive library of standard financial calculation formulas that we can use to validate the formulas being used by the LLM. We can cross reference this library with the LLM generated formula, creating a rule-based validation system so that we are not only checking the final answer, but the methodology that the LLM has used.
* I have used OpenAI GPT 4o however if we switched to a financial domain specific model it may provide extra contextual understanding that could improve the understanding of the questions being asked and the calculations that are needed to perform.
* To increase reliability, we could introduce more validation mechanisms that are independent to the LLMs reasoning and self-validation. We could use symbolic mathematics libraries (such as sympy which allows you to parse and analyse mathematical expressions) to verify formula structure and check and validate the mathematical logic. This would provide a more objective assessment of the LLM’s calculations.

1. **Conclusion**

The financial QA prototype demonstrates promising capabilities in understanding and answering numerical financial questions, with the accuracy metrics providing a view of performance, highlighting the strengths and areas for improvement. We can see that fuzzy matching improves the perceived accuracy of the system, and self-validation offers insights into the reasoning of the LLMs decisions. In future iterations, we should focus on improving the contextual understanding of the LLM (done by fine tuning the model or switching to a domain specific model) and developing more robust validation mechanisms to test the formula and calculations used by the LLM.