Embedding Tables in Text Context: NTCIR - 18 U4 Tasks

Hiroyuki Higa, Yuuki Maeyama, Kazuhiro Takeuchi

Osaka Electro-Communication University

Introduction

Research background

- Financial reports (e.g., statutory annual securities filings) contain extensive tables that list key corporate indicators.
  - Manual review is labor-intensive, costly, and prone to error.
- Conventional natural-language-processing (NLP) tools focus on narrative text.
  - They fail to capture the relational structure of rows and columns.
- New techniques must understand table structure and return exact figures in response to queries.

Research purpose

- Develop an error-detection and correction framework that improves large-language-model (LLM) accuracy on Table Question Answering (TQA) tasks for financial documents[1].
  - Use an LLM to parse tables and generate candidate answers.
  - Apply an auxiliary verifier to judge answer correctness, boosting accuracy without further fine-tuning of the LLM.

Proposed method

Integrated approach: coupling an LLM with a classification module

1. A query (Query), prompt (Prompt), and table (Table) are input into the LLM.
2. The LLM processes this information and generates an initial output.
3. This output is then converted into embedding vectors using a BERT model.
4. The embeddings are fed into a Gradient Boosted Decision Trees (GBDT) classification model, which evaluates whether the LLM's response is accurate.
5. If an error is detected, the model prompts the LLM to regenerate a more accurate response.

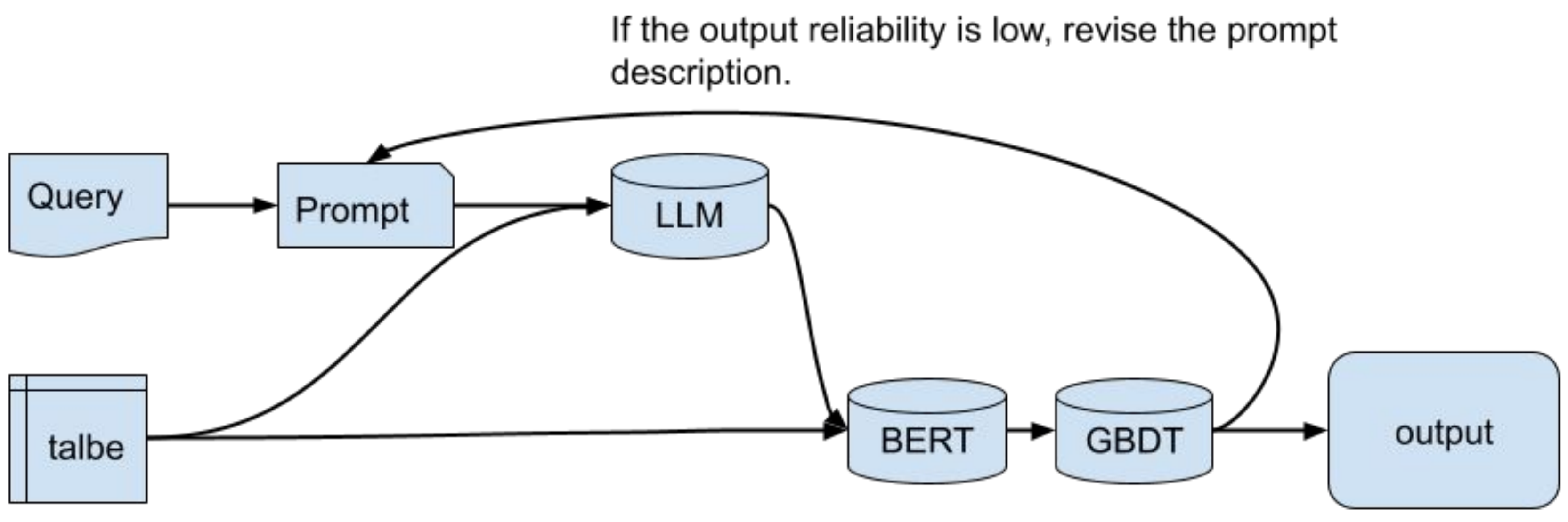

Figure1：Flowchart of the error-detection loop

Dynamic prompt injection to prevent repeated errors

- We reuse the prompt templates in Table 1.
- When the verifier flags an error, we insert an extra instruction (black box) that tells the LLM to identify and correct its previous mistake step by step.
- This targeted feedback prevents the repetition of identical errors and raises accuracy without additional fine-tuning.

Table1：template pronmpt

Type	Instruction prompt (excerpt)
System	*Task:* Return an answer based on the provided table and question. Follow these rules…(omitted for brevity)
User (1st pass)	Generate an answer using only the values in the table. Return `Response:{value}` on a single line.
User (regeneration)	The previous answer was incorrect. Analyse step by step and correct it. Generate an answer using only the values in the table. ### Previous incorrect answer: {prev_answer} Return `Response:{value}` on a single line.

Results and Discussion

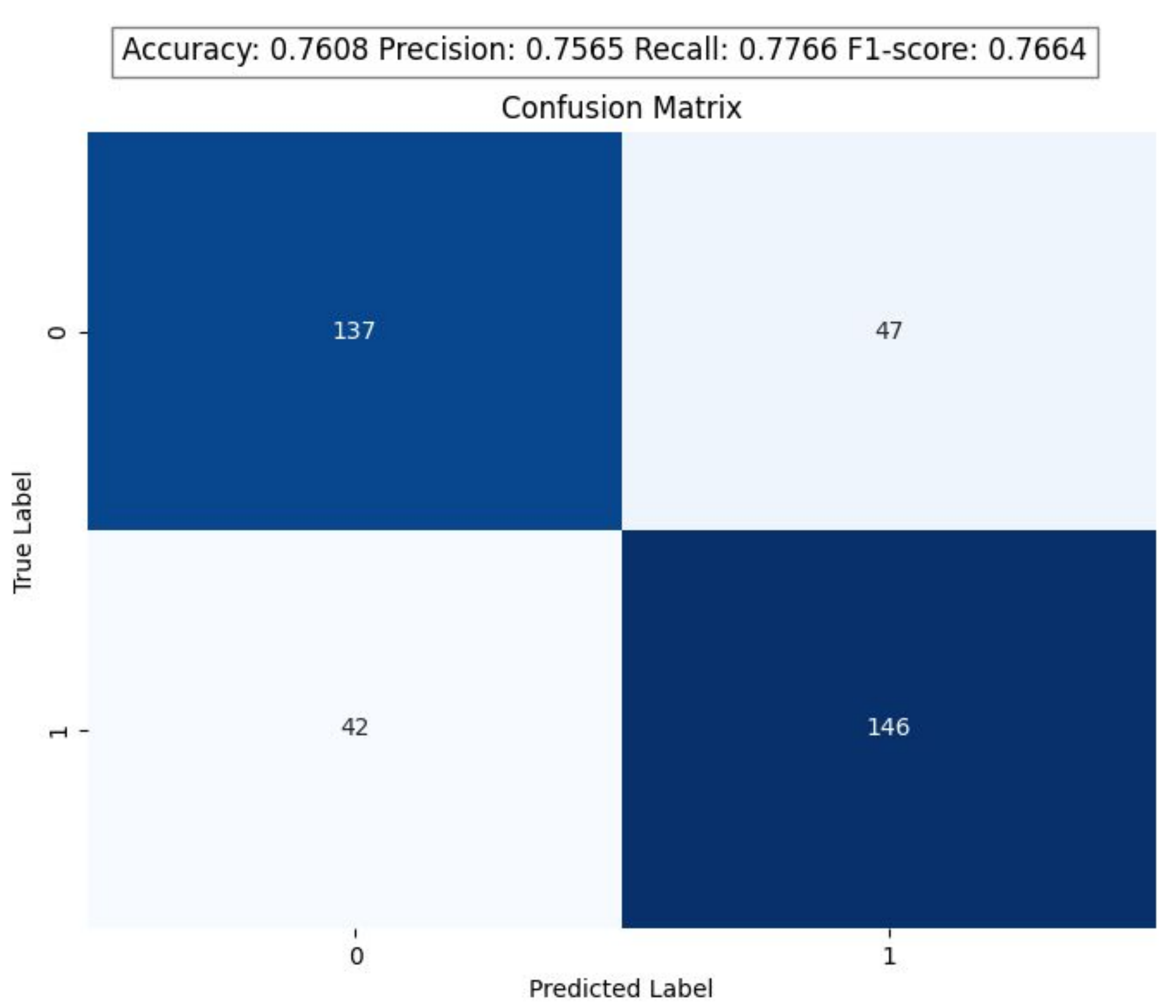

Figure2：Predictive accuracy of the verifier in distinguishing correct and incorrect LLM answers.

- **Accuracy on the Table QA benchmark: 0.4959.**
- **Current limitation**
  - T The verifier detects wrong answers, but simply stating “your answer is incorrect” seldom changes the output; accuracy gains are minimal.
- **Planned improvement**
  - Build a taxonomy of common error types (off-by-row, off-by-column, magnitude error, unit error, etc.).
  - Extend the verifier to output both correctness and **error category**.
  - Inject the specific error category into the follow-up prompt so the LLM can focus on the exact revision needed.

Reference：[1] Kimura, Y., Sato, E., Kadowaki, K., & Ototake, H. (2025). *Overview of the NTCIR-18 U4 Task*. Proceedings of the 18th NTCIR Conference on Evaluation of Information Access Technologies, June 2025.