Luoqi Zhao
Daniel Jones
Ryan Vera

CS 6320
5/6/2025

NLP Bookkeeping Project

1 Summary

The NLP Bookkeeping Project is a Natural Language Processing (NLP) application developed for the CS6320 course at UT Dallas in Spring 2025. It is designed to mimic a personal finance application that allows users to insert, update and delete transactions into a CSV-based database. A key feature of this project is the ability for Google's Gemini model to analyze the transaction data based on a user's prompt. This feature will be discussed further in the Key Feature section.

2 Key Features

2.1 Google Gemini API

A key feature of the project is the AI analyze functionality, which allows users to harness the power of AI for advanced transaction analysis. By passing transaction history and a user-defined question to the Gemini API, the system provides detailed insights and answers, enabling users to perform complex queries on their financial data.

Additionally, Gemini is responsible for taking user input and selecting the appropriate function that is used to manipulate the database. It transforms the user's natural language into a proper function call that can easily interact with the database in a more controlled manner.

The result of how the specified function interacted with the database is returned to the frontend so that the user can easily understand how their prompt affected the database.

3 Implementation

3.1 Database

The database operations, such as reading, writing, and updating, are performed on a CSV file using the Pandas library, keeping the project lightweight and focused on NLP functionality.

The example database we use details whether the entry is an expense or payment which indicates whether it is an increase/decrease to the user's account. It also has 14 unique categories for what type of transaction occurred. Finally, it tracks the date of each transaction to model overall transactions over time.

3.2 Python FastAPI Framework

The backend of the project was implemented using the FastAPI framework written in Python. Users can interact with the system through a SwaggerUI-like interface, which simplifies API interaction. While the project focuses on NLP and function calling, it also includes basic API endpoints for health checks and generating AI responses. The system is designed to be intuitive and user-

friendly, making it a practical application of NLP in personal finance management.

3.3 React GUI

The frontend of the project is built using the React-based Next.js framework, featuring a component-based architecture and responsive design. Key functionalities include transaction display and filtering, as well as transaction management. The system communicates with the FastAPI backend in real time via RESTful APIs, utilizing a local-first loading strategy and optimistic updates to enhance loading speed and interaction responsiveness. Additionally, it integrates an AI chat component that enables natural language interaction with the ledger through Google's Gemini model, delivering an AI-powered smart bookkeeping experience.

4 Contributions

4.1 Contributions – Luoqi

- Complete design and implementation of the frontend GUI

- Partial design and implementation of backend API functionalities

- Several adjustments to the database schema

- Several adjustments to the prompt for Gemini

4.2 Contributions – Daniel

- Built initial basic model for CRUD.

- I explored using ChatGPT for function calling, we later settled on Gemini.

- Filled out database to provide more realistic query interactions.

4.3 Contributions – Ryan

- Explored different Python API frameworks, LLM models and versions, and database implementations.

- Implemented and maintained backend API, database, and Gemini configurations and function calling.

4.4 Self-Scoring – Luoqi

75 points – Significant Exploration Beyond Baseline:

Identified the challenge of importing accounts recorded in natural language into bookkeeping apps. To address this, I proposed the project concept: leveraging LLMs' function calling capabilities to convert natural language descriptions into structured database entries.

25 points – Innovation or Creativity:

Introduced innovative modifications to the traditional bookkeeping app UI, enabling it to support natural language input while preserving the conventional manual input method, striking a balance between aesthetics and user experience.

10 points – Discussion of Lessons Learned and Potential Improvements:

On the frontend side, this was my first time independently designing UI/UX, during which I learned a lot about user experience design principles. On the NLP side, I gained a deeper understanding of LLM function calling and personally designed several function tools for the LLM to use.

As for potential improvements, I believe enabling LLM to chain multiple function tool calls would allow it to better handle user

commands—for example, when a user wants to "add a transaction for yesterday," LLM could first call a time tool to retrieve the current date, then call the insert tool to log the transaction.

10 points – Exceptional Visualization/Diagrams/Repo:

Responsible for the design and implementation of the entire project's UI/UX, including visualizations for the transaction records and the LLM input interface.

4.5 Self-Scoring – Daniel

65 - significant exploration beyond baseline
30 - Innovation or Creativity: Function Calling to make DB querying easy for anybody.
10 – Improvements: Use a Relational DB to generalize & expand/generalize functions to allow for more general queries.
10 - exceptional visualization/repo

4.6 Self-Scoring – Ryan

75 points – significant exploration beyond baseline: Evaluated and implemented various Python API frameworks to select best one for project. Explored various LLMs and versions to find the most lightweight one for project while completing necessary tasks.

25 points – Innovation or Creativity: Implemented simplified approach to function calling with Gemini.

10 points – Exceptional repository: Created and maintained clear and readable repository and README file. Additionally, added and maintained easy-to-follow instructions for running the program.

5   Repositories

The link to the GitHub repository for the backend API can be found here. A copy of this report in PDF format can be found in the repository as well.

The link to the GitHub repository for the frontend GUI can be found here and a link to the website can be found here.