NLP Project Report

Group members:
Maunika Achanta
Mir Patel
Tushar Wani

Project Description

Overview

Many students utilize Notion for organizing academic notes and tasks. While powerful on desktop, Notion's mobile application can be slow and cumbersome for quick note-taking or accessing information on the go. This limits its utility for capturing thoughts or appending tasks spontaneously.

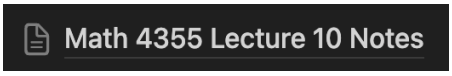
To address these limitations, we developed Voice-to-Notion, a web application providing a streamlined, voice-first interface for interacting with a user's Notion workspace. It allows users to use natural language commands via voice or text to perform actions within Notion, significantly simplifying mobile usage.

GitHub Link:

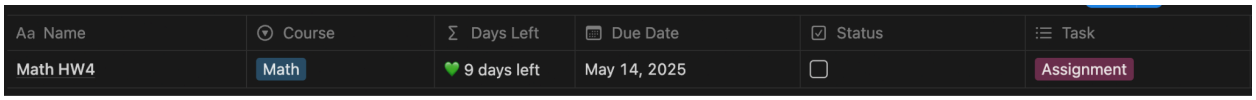
https://github.com/TusharW4ni/cs6320-project

Features

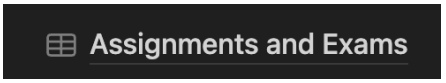
- Natural-language commands to execute CRUD (create, read, update, and delete) operations on the user's Notion account
    - Through either voice or text input create new pages and add content inside of them

    - Adding assignments with different parameters (Name, Course, Due Date, Days Left, Task tag type, Status)

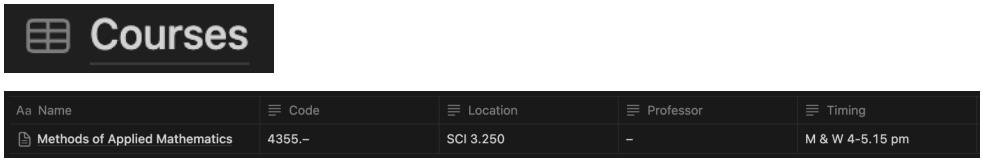
      

        - Create a database if one doesn't previously exist

        - Update different parameters of a page in the database (such as changing the due date, adding a course or task tag, or checking off an assignment

- Extract relevant information from syllabi documents and collate in a Notion database

    - Thanks to the large context window of Gemini models (about a million tokens) the user can upload all the syllabi for classes they have ever taken

    - The website accepts .doc/.docx/.pdf files. However, Gemini APIs do not accept .doc or .docx files. To solve that we convert .doc and .docx files using the ILovePDF APIs into .pdf files.

- We then send the files to the gemini-2.0-flash model with a JSON object representation of the output we expect

- Along with the expected output shape, the files, and system prompt, we also use Gemini's function calling feature
  - We declare the structure and I/O of the API's available on our backend
  - Gemini returns the function signature it believes will help in satisfying the request
  - The API is run on our backend using Gemini's JSON structured output

- Our APIs call the Notion APIs to satisfy user's requests

Lessons Learned

- Making the prompt handling more robust to satisfy ambiguous queries
    - Robustness against various forms of querying due dates when adding assignments
    - Resilience against different syllabus formats
- Utilizing prompt engineering in an efficient way to get Gemini to perform various tasks, whether that be function calling or code generation.

Contributions

Details in the project description section

Maunika Achanta

- Implemented the update assignments feature to change different parameters of an assignment.
- Created vector embeddings of the Notion API documentation using Google Generative AI's embedding function
- Implemented a RAG system to interact with the Notion API via the vector embeddings
    - Allows Gemini to generate and execute code based on user queries to perform different tasks in Notion

Mir Patel

- Implemented assignment creation
    - Includes Course, Assignment name, Due date, Days until due, tags, and status
- Implemented database creation
    - Calls on database creation if not already present when assignment creation is called
- Adjusted Gemini function calling logic to handle page, assignment, and database creation

Tushar Wani

- Implemented the voice querying feature
- Implemented the new page and adding contents inside of the page feature
- Implemented the the syllabi data extraction feature
- Scraped Notion API documentation website to build the vector store to implement RAG

Self-Scoring

Mir Patel

60/80 points	significant exploration beyond baseline (couple issues detected and partially solved)
20/30 points	Innovation or creativity: demonstrated unique approaches such as using novel techniques or creative data augmentations
5/10 points	Highlighted complexity - could be data gathering, error analysis, architecture, optimization, etc.
10/10 points	Discussion of lessons learned and potential improvements
0/10 points	Exception visualization/digrams/repo
0/10 points	Discussion of testing outside of the team, on 5 people
0/10 points	Earned money with the project

Maunika Achanta

60/80 points	significant exploration beyond baseline (couple issues detected and partially solved)
25/30 points	Innovation or creativity: demonstrated unique approaches such as using novel techniques or creative data augmentations - Implemented automatic executable code generation based on user queries to perform different Notion tasks

10/10 points	Highlighted complexity - could be data gathering, error analysis, architecture, optimization, etc.
10/10 points	Discussion of lessons learned and potential improvements
0/10 points	Exception visualization/digrams/repo
0/10 points	Discussion of testing outside of the team, on 5 people
0/10 points	Earned money with the project

Tushar Wani

65/80 points	significant exploration beyond baseline (couple issues detected and partially solved)
20/30 points	Innovation or creativity: demonstrated unique approaches such as using novel techniques or creative data augmentations
0/10 points	Highlighted complexity - could be data gathering, error analysis, architecture, optimization, etc.
10/10 points	Discussion of lessons learned and potential improvements
5/10 points	Exception visualization/digrams/repo
0/10 points	Discussion of testing outside of the team, on 5 people
0/10 points	Earned money with the project