notifAI

Note-taking Assistant mobile app

Tin Do

Marc Bernardino

Kolade Alabi

01

OVERVIEW

Our Background

As netizens, our team of Computer Science students formed a strong bond on Discord. We are interested in exploring new technology and its application in the real-world. Cohere hackathon, in collaboration with lablab.ai, presented an opportunity for us to do a fun and exciting project together.

All of us had something new to learn during the project: two teammates learned React Native for the first time, and one teammate got his first exposure to Python.

Our Problem Statement

Our target client is students, more specifically students at higher education institutions such as universities. We envision two use cases for our app.

1. The students’ notes taken during class are usually long and/or difficult to understand. Sometimes, handwritten notes in fast-paced lectures are barely legible. We face this experience first-hand as Computer Science students: notes for difficult classes such as data structure and algorithms are usually very involved.

2. Lecture slides and textbooks contains much information that are good for intuitive understanding but are not necessary for exams. As a result, students have the need to make the information more compact, which helps with long-term memory of the knowledge.

The Solution

To address this problem, we created a tool that automatically summarizes text. Given a selection of text, notifAI will return an easy-to-read summary and some main ideas.

To help make this process easier for our target audience, we also included image-to-text functionality. This way, all one needs to do is take a picture of a textbook page for summarization.

02

Methodology

Main Components

NLP

Summarization API powered by Cohere

Vision

Google Cloud Vision API’s document text detection feature

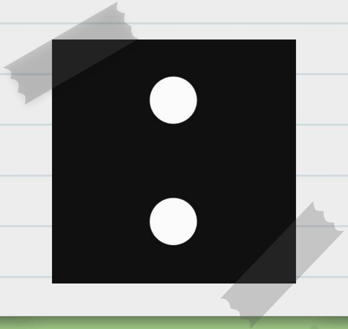
App

Main application layout

Summarization API

To obtain the summarizations that are the core function of notifAI, we utilized Cohere’s text generation model. Examples of textbook sections, as long as their summaries, were provided as prompts to the model. In turn, it returns accurate key points and summarizations.

The request to the API was then hosted in a Flask API on pythonanywhere, making it accessible by a simple API call.



Google Cloud Vision API

We employed the text_detection feature, which is part of Google's Optical Character Recognition service, to detect texts on an image. More specifically, we convert the image content taken by the client's camera into base64 encoding. We then send REST post request, with the encodings in the request's body, to the API and get a response back in the form of JSON. We then extract the specific information that we want to feed into the Cohere's summarization API, described in the below slide.

Main Application

We used React Native for our app architecture, which includes accessing the camera, displaying each page for the user, and making API requests to the other critical pieces of our application. With React, the application can deploy to IOS, Android, or as a Web application.

03

Demo

Thank you!