

Optimizing Student Schedule Coordination through GenAI

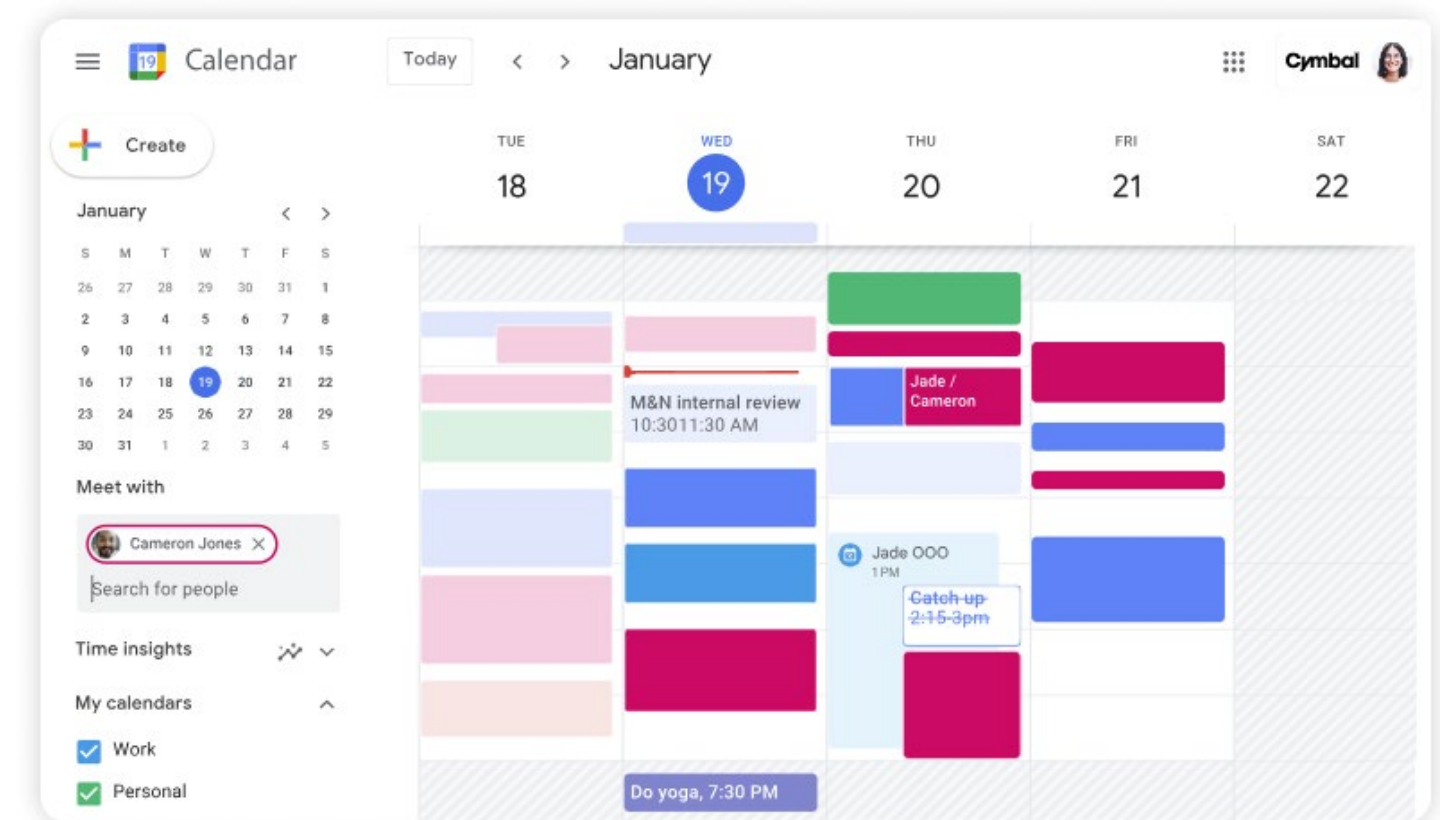
Student Schedule Coordination

Task:

- Challenge of coordinating 10 students' schedules
- Traditional manual scheduling inefficiencies

Goal:

- Find timeslots with 8+ students available
- Test GenAI capability in automated schedule optimization
- Use Ground Truth, to measure accuracy of the different models



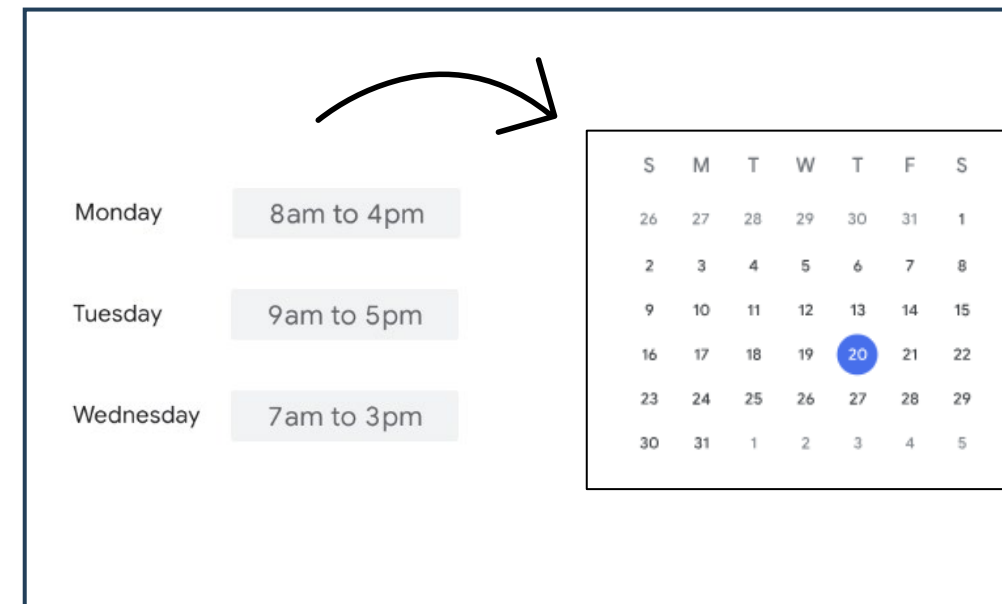
Methodology

Data preparation & GenAI implementation



Prompt engineering for models [1] to filter out all unavailable timeslots

Timetable generation



Timeslots to timetable generation

Evaluation framework

Metrics:
Accuracy
Precision
Recall
F1 Score

Ground truth comparison, evaluate model quantitatively

GenAI Prompt

Prompt (ChatGPT as an example)

Question

✎ solve the problem step by step
Ten Students are working on a group project. They want to find some time to meet.
They have the following constraints (morning means from 8 AM to 12 PM, and afternoon means from 1 PM to 5PM):
Student A: I have classes on Monday and Wednesday from 2 PM to 4 PM, club activities on Tuesday and Thursday evenings from 7 PM to 9 PM, and I'm available on Friday afternoons.
Student B: I have a lab on Monday from 9 AM to 11 AM, classes on Tuesday and Thursday afternoons from 1 PM to 3 PM, and I'm free all day on Friday, but I might have to work late in the evening after 10 PM.
Student C: I have classes every weekday morning from 8 AM to 10 AM. I may have to work on weekends from 3 PM to 5 PM. If A calls me on Friday morning, we'll go play tennis in the afternoon 6 PM – 8 PM. I'm free otherwise.
Student D: I have classes on Monday and Friday from 10 AM to 12 PM

Answer

To solve this problem, let's break it down step by step. We'll go through each student's schedule to find the times they are unavailable, then cross-reference these times to identify any overlapping free times where at least 8 students are available.

Step 1: Summarize Each Student's Availability and Unavailability

We'll outline the unavailable times first, noting these in a weekly schedule format (Monday through Sunday, 8 AM - 10 PM).

1. Student A:

- Unavailable: Monday, Wednesday (2 PM - 4 PM), Tuesday, Thursday (7 PM - 9 PM).
- Available: Friday afternoon.

2. Student B:

- Unavailable: Monday (9 AM - 11 AM), Tuesday, Thursday (1 PM - 3 PM), possibly Friday evening after 10 PM.
- Available: All day Friday until 10 PM.

3. Student C:

- Unavailable: Weekday mornings (8 AM - 10 AM), weekends (3 PM - 5 PM), Friday evening (6 PM - 8 PM, if A calls for tennis).

Filtering & Timetable Generation

Python libraries:

Libraries	Uses
openai	GPT-4o API
re & ast [1]	Extract each timeslots data from string Convert string to dictionary format
pandas	Create table, can export as excel file Import excel file, calculate model's performance

[1] re: Regular Expression, ast: Abstract Syntax Trees

Evaluation Framework

Ground Truth: The baseline benchmark for comparison

Metrics:

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Results & Evaluation

Model	Accuracy	Precision	Recall	F1-score
GPT-4o	0.62	0.98	0.63	0.77
Claude-3.5-Sonnet	0.74	1.00	0.74	0.85
Gemeni-1.5-Pro	0.3	0.47	0.45	0.46
Llama3.1-405B	0.387	0.59	0.53	0.56

Table 1: Model performance vs Ground Truth

Best Performing Model: Claude-3.5-Sonnet

Conclusion

Contributions:

- We proposed a method, to streamline the process of time scheduling via GenAI
- Created a standardized prompt engineering approach for timetabling

Future work:

- Integration with calendar systems (eg: Google Calendar)
- Expanding to handle more complex scheduling constraints