AI-Powered Task Distribution and Optimization System

By: Anushka Sethi, Laura Elshaer, Anipreet Chowdhury and Prarthona Paul

Executive Overview of the Project

Key Objective: Promote prompt delivery of project components and reduce stress levels to enhance task distribution within team structures

Strategy: Divide large deliverables into smaller, manageable parts and maximize current available time resources towards task progress.

Benefits:

- More efficient workflow.
- Streamlined integration into daily work routines.
- Optimized work schedules for each team member.

Result: Consistent progress towards project objectives.

Mechanism: Ensures group stays on track, fosters shared responsibility, and ensures timely project completion.

Primary Functionality

To achieve our goal of optimizing workflows and dividing tasks into smaller, manageable chunks, we wish to develop a web-based application that will do the following tasks:

- Information Extraction from PDF Documents
- Task Classification and Timeline Assignment
- Sub-task Suggestion Feature Development
- Integration with Team Members' Calendars

Each of these tasks acts as a milestone, with each utilizing a different technique to maximize its potential. We wish to build this application in phases, with each phase focusing on one task and building on the previous task.

Objective 1: Information extraction from PDFs

This task consists of two components: converting the PDF into a format that can be parsed and extracting the necessary information from the converted format. Let's discuss a few methods for accomplishing both, as we'll need to justify our choice of method for each task.

- **Converting PDF into Parsable format:**
  - PDFs are commonly used sources of information but are challenging to automatically parse.
  - Efficient extraction methods are crucial for extracting contextual data swiftly.
  - The chosen method must be efficient with low error rates.
- **Applying NLP to converted PDF to extract information:**
  - With the converted file in a parseable format, NLP techniques can be applied.
  - Compare various NLP methods to extract relevant information efficiently.
  - The goal is to assess each method's suitability based on its advantages and limitations, selecting the most appropriate one for our use case.

Objective 2: Task Classification of extracted data

This task involves three main subtasks: classifying the main deliverables extracted from the PDF into predetermined brackets, projecting a suggested timeline for task completion based on factors such as category and task description, and reinforcing correct projections to increase prediction accuracy.

- **Main Task Classification:**
  - Classify main deliverables into predetermined categories.
  - Train a pre-existing transformer model like Llama 2 with a high-quality dataset of over 1000 PDFs.
- **Task Duration Deduction:**
  - Adjust task durations based on various criteria like task description, category, and mark weightage.
  - Utilize regression techniques to calculate task durations and train models with existing engines.
  - Design a UI to display project timelines resembling a Gantt chart.
- **Reinforcement Learning:**
  - Allow users to correct predicted tasks and durations, improving predictions over time.
  - Integrate reinforcement learning into the tool to incorporate user adjustments and enhance prediction accuracy.

Objective 3: Task Assignment and allocation

This task involves gathering when individuals of the team would be free, and assigning them existing tasks on the basis of how much time they have. This task consists of the following subtasks:

- **Task Allocation Algorithm:**
  - Design the algorithm to maximize the utilization of available hours while minimizing conflicts and overlaps.
  - Consider factors such as task dependencies, priority, and individual skills when allocating tasks.
  - Implement the algorithm to efficiently assign tasks to team members within their available time slots.
- **Data collection and pattern creation:**
  - Design a user-friendly interface similar to when2meet for team members to input their weekly available hours.
  - Develop a backend system to collect and store team members' available hours.
  - Implement logic to handle different availability patterns (e.g., recurring availability, one-time exceptions).

Objective 4: Sub-task Suggestion Feature

The task here is to to divide larger deliverables into smaller, more manageable tasks based on the task description. This will allow us to keep track of each task in accordance with its timeline and urgency, thus ensuring that no task or subtask is overlooked or delayed.

- **Contextual Understanding and divisions:**
    - Divide a task into subtasks on the basis of task description.
    - Use of context and other data such as time allocated for main task.
    - Use of NLP and tokenization mechanism to break tasks into smaller tasks
- **Reinforcement Learning:**
    - Allow user to provide feedback on suggested subtasks.
    - Use feedback to improve future suggestions and predictions

Plan of action

We plan to build this project in stages, with each stage relating to an objective and building it further. Our current plan of action is as follows:

- **Milestone 0**: (Current Stage)
  - In depth research into each objective.
  - Gathering sources to ensure we choose the best method to address each problem.
  - Creating starting database of course syllabi to train models.
- **Milestone 1:**
  - Create information extraction tool and starting user interface.
  - Application should allow user to upload a PDF and extract all key tasks from it.
- **Milestone 2**:
  - Create and implement task categorizer.
  - Application should use extracted data to classify tasks and project completion timelines.
- **Milestone 3**:
  - Implement task allocation algorithm.
  - User should be able to put in available hours, application should automatically assign tasks.
- **Milestone 4**:
  - Implement subtask generation and allocation.
  - Implement into working application