HARDPOINT
CONSULTING

Hackathon Proposal

Internship Application Project

1. Define Scope and Requirements

Input: Academic articles (PDFs, HTML, etc.).
Output: Populated knowledge database.
Accuracy: >=90%

2. Metadata Extraction

- **Tools:** Use a metadata extraction tool to parse articles and extract basic information (Title, Year, Journal, Authors).
- **Process:** Automatically scan the document's metadata or use text recognition for scanned documents.

3. Text Processing and Analysis

- **Preprocessing:** Convert articles to a text format suitable for analysis (OCR for scanned documents, HTML/PDF parsing).
- **Segmentation:** Break down the article into sections (Abstract, Introduction, Methodology, etc.) for targeted analysis.

4. Large Language Model Integration

- **Model Selection:** Choose a model like GPT-4 or a specialized academic-focused model.
- **Custom Training:** Consider fine-tuning the model on a dataset of academic writings for better performance in this context.

Extraction Tasks: Use the model to fill in:

- Study Description
- Constructs
- Theoretical Perspective
- Study Context
- Method Description
- Level
- Core Findings
- Executive Summary

5. Data Validation and Quality Assurance

- **Automated Checks:** Implement algorithms to validate the extracted data (e.g., year format, author names).
- **Manual Review:** Establish a process for manual review, especially for critical or complex sections like 'Core Findings'.
- **Integrated Feedback Loop:** Allow users to provide a simple thumbs up or down. Also, please allow for unstructured text responses in two formats – comments and complete answers generated by humans.
    - This will likely need to be integrated at every extraction task labeled above.

6. Database Integration

- **Database Design:** Structure the database to reflect the Excel template.
- **Automation:** Develop scripts to populate the database with extracted and processed data.

7. User Interface

- **UI for Review/Editing:** Create a simple interface for users to review, edit, and confirm entries before final submission to the database.

8. Testing and Iteration

- **Prototype Testing:** Test the system with a small set of articles and refine the process.
- **Iterative Improvement:** Continuously improve the model and process based on feedback and performance.

9. Documentation and Training

- **User Guides:** Prepare detailed documentation for end-users.
- **Training Sessions:** Conduct training sessions for users who will interact with the system.

10. Deployment and Maintenance

- **Deployment Strategy:** Determine how the software will be deployed (cloud-based, local server, etc.).
- **Maintenance Plan:** Establish a maintenance plan for software updates, model retraining, and database management.

Considerations and Challenges:

- **Accuracy vs. Automation Trade-off:** More automation can lead to lower accuracy. Balance is crucial.
- **Model Bias and Limitations:** Language models might introduce biases or misunderstand complex academic concepts.
- **Keep The Project Python:** Keep as much of the project as possible python based so a large number of community members can benefit from it.

Notes:

- Remember that I am looking for your best effort, not perfection. Even if you are unable to complete all of this, please submit what are create.
- Consider different resources to help
  - You may (or may not) need to use agents
    - Consider using LangGraph if you do, it allows for great control
  - You may (or may not) need multiple LLM calls
  - You may (or may not) need multiple LLM models to handle different tasks
  - You may (or may not) need to use an LLM that can generate JSON output
- An example Excel file is also in the GitHub repo