

## CS410 FALL 2023 PROJECT PROPOSAL

**Team Members:** Hermann Rosch, Sudeshna Pal, Ahmad Barqawitz

**Team Captain:** Hermann Rosch

**Team Name:** Insightful Highlighters

**Project Name:** Textual Spotlight

**What topic have you chosen? Why is it a problem? How does it relate to the theme and to the class?**

Our group project aligns with the **Intelligent Browsing** topic, and we aim to create:

*“Textual Spotlight – A Chrome Extension for Contextual Text Understanding with GPT-4”*

The vast amount of text websites often leaves users confronted with complex explanations or jargon-filled content. With this Chrome extension, users will be able to gain deeper insights into the textual content they encounter on the web. By leveraging the capabilities of GPT-4 and traditional text retrieval techniques, users can highlight portions of text and request explanations, summaries, paraphrases, and more via buttons with prompts right within their browsers.

**Briefly describe any datasets, algorithms, or techniques you plan to use**

*Dataset:* The text web content highlighted by users.

*Algorithms:*

- **TF-IDF:** For evaluating the importance of words in the highlighted content. By understanding which words or terms are “most unique” or relevant in a section of the text, we can provide better context to GPT-4 when requesting explanations or summaries.
- **VSM:** Used to represent the highlighted text as vectors in a multi-dimensional space. By comparing the cosine similarity between the vector of the highlighted text and vectors of potential explanations or summaries, we can ensure that the most relevant and contextually appropriate responses are selected; this representation can aid in quantifying the similarity between the highlighted content and the generated summaries, ensuring that GPT-4's outputs are contextually aligned with the original text in the websites.

*Techniques:*

- Web scraping using JavaScript to capture the highlighted content.
- Using prompt engineering to optimize GPT-4 responses.
- Text preprocessing using Python (tokenization, stemming, etc.).

## How will you demonstrate that your approach will work as expected?

Members of our team will adopt the role of "expert users" to evaluate the functionality and efficacy of the Chrome extension. Specifically, we'll:

1. *Highlight and Analyze:*
  - Use the extension on various web articles to highlight text and request GPT-4's explanations, summaries, or paraphrases.
2. *Check Relevance and Clarity:*
  - Manually assess the relevance and clarity of the GPT-4 generated content in relation to the highlighted text.
3. *Functionality Testing:*
  - Confirm that the user interface (buttons with prompts) responds as expected.
  - Ensure the extension's interaction with the OpenAI API is smooth and error-free.
4. *Documentation:*
  - Record instances where the extension functioned optimally, and note any areas needing refinement or improvement.

## Which programming languages do you plan to use?

- *JavaScript* (for Chrome extension development)
- *Python* (for backend text processing and retrieval operations)

## Workload Justification – Estimated total time: 60 hours

### 1. Chrome Extension Development – 20 hours

- 1.1. Feature design and scoping (6hrs)
- 1.2. Frontend UI design and implementation (6hrs)
- 1.3. Integration with the Python backend server for text processing (2hrs)
- 1.4. Integration with OpenAI API for GPT-4 responses (2hrs)
- 1.5. User interaction handling (frontend logic, including buttons for highlights and prompts) (4hrs)

### 2. Python Backend Development – 18 hours

- 2.1. Setup and development of the Python backend server (5hrs)
- 2.2. Text preprocessing techniques (tokenization, stemming, etc.) in Python (7hrs)
- 2.3. Integration and application of algorithms like TF-IDF and VSM (7hrs)

### 3. Prompt Engineering – 10 hours

- 3.1 Designing and testing prompts for GPT-4 to optimize responses (6hrs)
- 3.2 Integrating the prompts to the frontend JS code (4hrs)

### 4. Testing and refinement tasks – 12 hours

- 4.1 Testing backend functionalities (3hrs)
- 4.2 Improving backend text processing functionalities (3hrs)
- 4.3 Testing buttons and frontend functionalities (3 hrs)
- 4.2 Improving the UI/frontend (2hrs)

## Workload Distribution

Member	Task	Estimated Hours
Hermann Rosch	Setup and development of the Python backend server, Designing and testing prompts for GPT-4, Integration with OpenAI API for GPT-4 responses Integrating the prompts to the frontend JS code, Improving backend text processing functionalities	20
Sudeshna Pal	Text preprocessing techniques in Python, Integration of Algorithms like TF-IDF and VSM, Testing backend functionalities Testing frontend functionalities	20
Ahmad Barqawitz	Chrome extension feature design, Frontend UI design/display, Integration with the backend server for text processing, Integration with OpenAI API for GPT-4 responses, Developing buttons for highlighting texts and prompts Improving the UI/frontend	20