**Members**

**Yifan (Ivan) Zhang    yifan20**

**Jinyu Yang jinyuy2**

**Jialin He     jialinh4**

**Kaiwen Zhong   kaiwen4**

**Leader: Yifan Zhang**

**Introduction**

We are facing a world with an incredible amount of information, and the case is especially present for college students who need to read a massive number of documents and quickly determine whether they are useful. Therefore, the skill of skimming through a document and providing a brief summary and evaluation is essential. However, many people are not capable of this technique and reading over useless documents is quite a waste of time. In our course project, we endeavor to provide users with an automation tool of extracting keywords from a document and provide a quick summary with sentiment analysis via an extension on Google Chrome to optimize the reading experience of users. In addition, user will be allowed to ask questions on the document and the system will provide sentences or chunks that fits most with the query. We will also provide features for recommending similar documents if the user approves this material and provide other suggestions or let LLMs such as ChatGPT to generate user query / improvement on the current document.

**Estimated user impact**

The target user in this project is mainly college students. The average reading speed of a person is around 250 words per minute and research has shown that screen reading will decrease reading speed at around 20%, constituting an estimated value of 200 words per minute. Unfortunately, in college there are many international students whose reading speed is way below the above value and might take several times to understand a sentence. Take UofI for example, in most general education courses, students are required to read and reply to at least two discussion posts of 200-300 words. We assume that students read 3 posts of 250 words, which takes at least 3.75 minutes to read through. Given the tool provided and a heavily exaggerated estimate algorithm runtime of 10s, it only takes less than 20s to roughly a document by looking at generated keywords and summary. Therefore, reading 3 documents with our extension can be almost 4 times faster than usual.

**Technical Background**

**1. Document Preprocessing:**

* **Sentence Tokenization**: The document is broken down into individual sentences using natural language processing (NLP) techniques.
* **Handling Short Sentences**: Sentences with fewer than 15 words are concatenated with adjacent sentences to ensure meaningful context in subsequent analysis.

**2. Text Analysis and Indexing:**

* **Inverted Index Construction**: An inverted index is built to facilitate quick and efficient text retrieval. This data structure stores a mapping from terms to their locations in the document.
* **TF-IDF Calculation**: Term Frequency (TF) and Inverse Document Frequency (IDF) are calculated for each keyword to measure its importance within the document and across a collection of documents.
* **Keyword Extraction**: Significant terms are identified based on their TF-IDF scores, serving as a foundation for summary generation and sentiment analysis.

**3. Summary and Sentiment Analysis:**

* **Summary Generation**: OpenAI's API or other summarization techniques are employed to generate a concise one-sentence summary of the document.
* **Sentiment Analysis**: The sentiment or tone of the document is analyzed to provide additional context to users, helping them quickly gauge the document’s nature.

**4. Search and Query Processing:**

* **Search Functionality**: A search bar is incorporated to allow users to query the document.
* **BM25 Algorithm**: The BM25 algorithm is used to rank sub-documents or sentences based on their relevance to the user’s query, ensuring accurate and pertinent results.

**5. Recommendation System:**

* **Page Ranking for Suggestions**: Page Ranking or similar algorithms are used to recommend additional documents or resources based on user preferences and interaction with the current document.
* **Contradictory Document Suggestion**: In case the user dislikes the current document, the system is capable of suggesting documents with opposing viewpoints.

**6. User Interaction and Feedback:**

* **Question and Answer Feature**: Users are allowed to ask questions about the document, with the system providing relevant sentences or chunks in response.
* **Document Improvement with LLMs**: A section is dedicated to using Large Language Models (LLMs) like ChatGPT to generate user queries or suggestions for improving the current document.

**7. Chrome Extension Development:**

* **Browser Integration**: The tool is developed as a Google Chrome extension, ensuring easy accessibility and a seamless user experience.
* **User Interface**: A user-friendly interface is designed to display keywords, summaries, sentiment analysis results, and additional features like search and recommendations.

**8. BM25 for Document Retrieval and Relevance Scoring:**

* **Algorithm Overview**: BM25 is a ranking function used as part of a probabilistic information retrieval model. It extends the probabilistic model by introducing the concept of term frequency saturation.
* **Term Frequency Saturation**: Unlike in the vector space model, where term frequency is assumed to have a linear relationship with relevance, BM25 assumes that term frequency saturation occurs. This means that the more times a term appears in a document, the less additional relevance each occurrence adds.
* **Document Length Normalization**: BM25 introduces document length normalization, adjusting the score of each document based on its length. This prevents longer documents from dominating the results solely because they have more terms.
* **Implementation in the Project**: In the context of this project, BM25 is employed to rank sentences or chunks of text in response to a user query. When a user inputs a search term or phrase in the search bar, BM25 is used to calculate a relevance score for each segment of the document.

**Functionalities & Implementation**

First, we will break the document into sentences and each of which represents a document. If sentences are too short (for example, under 15 words), we will piece several sentences together. After that, we will build an inverted index and extract the TF and IDF of keywords and we may use OpenAI’s API or other measures to provide a one-sentence summary of the document. We may also provide a search bar and use BM25 to process the query and find corresponding sub-documents.

In addition, for the recommendation system, we might use some Page Ranking methods to filter out some possible suggestions for the document if the user likes it. We can use the same logic to come up with contradictory documents if the user does not like it and we also leave a section for ChatGPT to come up with a better solution.

**Milestone**

**Week 1: Project Initialization and Planning**

* **Objective**: Establish a clear understanding of the project goals, scope, and requirements.
* **Tasks**:
  + Define project scope and objectives.
  + Identify and allocate resources (team members, tools, etc.).
  + Develop a project timeline with specific milestones.
  + Set up communication channels and project management tools.
* **Deliverables**: Project plan document and communication plan.

**Week 2: Environment Setup and Document Preprocessing**

* **Objective**: Prepare the development environment and start working on document preprocessing.
* **Tasks**:
  + Set up the development environment (IDE, necessary libraries, etc.).
  + Implement sentence tokenization.
  + Develop logic for handling short sentences.
* **Deliverables**: Document preprocessing module (initial version).

**Week 3: Text Analysis and Inverted Index Construction**

* **Objective**: Analyze text data and build an inverted index for the document.
* **Tasks**:
  + Implement TF-IDF calculation.
  + Extract keywords based on their TF-IDF scores.
  + Build an inverted index for quick text retrieval.
* **Deliverables**: Text analysis module and inverted index.

**Week 4: Summary Generation and Sentiment Analysis**

* **Objective**: Develop functionality for generating document summaries and performing sentiment analysis.
* **Tasks**:
  + Integrate with OpenAI’s API or other summarization tools.
  + Implement sentiment analysis.
  + Test summary generation and sentiment analysis with sample documents.
* **Deliverables**: Summary and sentiment analysis module.

**Week 5: Search Functionality and Recommendation System**

* **Objective**: Implement search functionality and start working on the recommendation system.
* **Tasks**:
  + Implement BM25 algorithm for document retrieval.
  + Develop the search bar and query processing logic.
  + Start working on the recommendation system, using Page Ranking or similar methods.
* **Deliverables**: Search functionality and initial version of the recommendation system.

**Week 6: Finalization, Testing, and Documentation**

* **Objective**: Finalize all modules, conduct extensive testing, and prepare documentation.
* **Tasks**:
  + Finalize the recommendation system.
  + Conduct comprehensive testing of the entire system.
  + Prepare user manuals and technical documentation.
  + Fix any identified bugs and optimize performance.
* **Deliverables**: Final version of the project, comprehensive test reports, and documentation.

**Post-Project (Optional): User Feedback and Iteration**

* **Objective**: Gather user feedback and prepare for potential future iterations.
* **Tasks**:
  + Release the tool to a select group of users for beta testing.
  + Collect and analyze user feedback.
  + Identify areas for improvement.
* **Deliverables**: User feedback report and list of potential future enhancements.

**Work Justification**

**Ivan Zhang: Project Manager and Document Preprocessing**

* **Responsibilities**:
  + Oversee the entire project, ensuring that all milestones are met.
  + Lead the document preprocessing efforts, implementing sentence tokenization and handling short sentences.
* **Justification**:
  + Ivan has strong organizational and leadership skills, making them well-suited for the role of Project Manager.
  + Their knowledge in natural language processing (NLP) is crucial for the initial stages of the project where document preprocessing takes place.

**Jialin He: Text Analyst and Inverted Index Constructor**

* **Responsibilities**:
  + Implement TF-IDF calculations and keyword extraction.
  + Construct the inverted index for quick text retrieval.
* **Justification**:
  + Student B has a solid background in text analysis and information retrieval, making them the ideal candidate for this role.
  + Their attention to detail ensures that the inverted index is built accurately, which is vital for the success of the search functionality.

**Jinyu Yang: Summary and Sentiment Analyst**

* **Responsibilities**:
  + Integrate with OpenAI’s API or other summarization tools.
  + Implement and optimize the sentiment analysis module.
* **Justification**:
  + Jinyu has experience in working with APIs and external tools, as well as a strong understanding of sentiment analysis.
  + Their technical expertise is necessary for ensuring that the summary generation and sentiment analysis are both efficient and accurate.

**Kaiwen Zhong: Search and Recommendation System Developer**

* **Responsibilities**:
  + Implement the BM25 algorithm and develop the search functionality.
  + Work on the recommendation system, using Page Ranking or similar methods.
* **Justification**:
  + Student D has a strong foundation in search algorithms and recommendation systems, making them well-suited for this role.
  + Their ability to optimize algorithms ensures that the search and recommendation functionalities are both fast and relevant.

**Collaborative Responsibilities:**

* All students will collaborate on:
  + Final testing and debugging of the system.
  + Preparation of user manuals and technical documentation.
  + Collecting and analyzing user feedback for future improvements.

**User impact Validation**

**1. User Surveys and Feedback:**

* **Description**: Distribute surveys to a group of college students, asking them to use the tool and provide feedback on its usability, efficiency, and overall impact on their reading and research process.
* **Validation Metrics**: Ease of use, time saved, perceived effectiveness, and overall satisfaction.
* **Impact Validation**: High scores in ease of use and time saved would validate the project’s impact in enhancing the reading experience.

**2. Comparative Analysis:**

* **Description**: Have users perform reading and information retrieval tasks with and without the tool, measuring the time taken and accuracy of information retrieved in both scenarios.
* **Validation Metrics**: Time taken to complete tasks, accuracy, and completeness of information retrieved.
* **Impact Validation**: Shorter completion times and higher accuracy with the tool would validate its effectiveness.

**3. Focus Group Discussions:**

* **Description**: Conduct focus group sessions with college students to discuss their experience using the tool, gathering qualitative feedback on its strengths and areas for improvement.
* **Validation Metrics**: Qualitative feedback on user experience, perceived value, and suggestions for improvement.
* **Impact Validation**: Positive feedback and constructive suggestions would validate the tool’s utility and highlight areas for future enhancements.

**4. Usage Analytics:**

* **Description**: Implement usage analytics within the tool to gather data on how often features are used, which features are most popular, and any potential issues users are encountering.
* **Validation Metrics**: Usage frequency of different features, user engagement duration, and error logs.
* **Impact Validation**: High usage of key features and prolonged engagement would indicate that the tool is providing value to its users.

**5. Academic Performance Correlation:**

* **Description**: With proper permissions and privacy considerations, analyze any correlations between the use of the tool and academic performance or efficiency in coursework.
* **Validation Metrics**: Correlation between tool usage and grades or assignment completion times.
* **Impact Validation**: A positive correlation would suggest that the tool is positively impacting academic performance.