Development Track - Group 13

Names, email: Aayush Agarwal (aayush8@illinois.edu), Suvodeep Saibal Sinha(sssinha2@illinois.edu)

Co-ordinator of the project: Aayush Agarwal (aayush8@illinois.edu)

Title: Context-Aware Bookmark Organization Tool (CABOT)

Functions and Users:

The Context-Aware Bookmark Organization Tool (CABOT) is envisioned as a hybrid software tool comprising a browser extension. This tool is designed to change the way researchers, students, and academics manage and retrieve web-based resources. CABOT uses natural language processing (NLP) to analyze the content of saved web pages facilitating automatic organization in a context-aware manner. Its major functions include:

- **Content and Context Analysis:** Understanding the textual content and user annotations to recognize the context and relevance of each bookmark using Key Phrase extraction.
- **Automatic Categorization:** Leveraging TF-IDF weighting along with other methods to dynamically categorize bookmarks into topics or research areas.
- **Organization:** The saved URLs are automatically organized in the existing bookmark manager on Chrome directly without the need for any other external interface for interaction.
- **Pop-up notification for categorization:** Whenever the bookmark icon is clicked for a specific URL, our extension will automatically assign it to a category for the organization, and it will also notify the user of the category it will be assigning the URL to through a pop-up notification.

The primary users of CABOT are researchers, academics, and students engaged in extensive web-based research, looking for a solution to efficiently manage and retrieve web resources relevant to their studies or projects directly on the bookmark manager on Chrome.

Significance

CABOT addresses a significant "pain point" for its users: the difficulty in organizing and retrieving a vast number of bookmarks and web resources using traditional, folder-based systems.

By leveraging an intelligent, context-aware system that understands the content and significance of each bookmark, CABOT improves the efficiency of managing and accessing the bookmarks through the bookmark manager by automating the bookmark organization. This tool has the potential to transform academic research by saving time, automating organization, and facilitating easier access to information.

Approach

To build CABOT, the following technologies and strategies will be leveraged:

- **Frontend Development**: The browser extension will be developed using HTML, CSS, and JavaScript.
- **Backend Processing**: Python, along with Flask, will be used for backend API development to manage data processing, storage, and NLP tasks.
- **Natural Language Processing**: Libraries such as Gensim or spaCy will be integrated for content analysis and semantic search capabilities.
- **Database Management**: SQLite will be used for local data storage, with options for cloud synchronization for backup and cross-device access.

Potential barriers include the complexity of accurately categorizing and retrieving bookmarks based on context and the challenge of ensuring user privacy and data security. These risks will be managed by adopting best practices in software development, focusing on robust NLP models, and implementing strict data handling and privacy protocols. Finding the length of text appropriate for topic modeling, which shouldn't be too short, that it is not informative, or too long that it is taking time to process.

Evaluation

We plan to achieve the "Keyword phase extraction" task using models like Gensim/Spacy. This method is to be benchmarked and evaluated with other Techniques, including fine-tuning with Llama2, BERT extractor, etc.

The common metrics seen for this task are :

Precision at K: Measures the top-K extracted keyword phrases' precision, evaluating the top-ranked phrases' quality.

Other common measures would be Precision, Recall, F1 Score, Accuracy, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG)

Timeline

- Day 1-5: Requirement gathering and planning.
- Day 5-10: Development of the browser extension.
- Day 10-13: Backend development and integration of NLP functionalities.
- Day 14: Initial testing.
- Day 15: Refinement and deployment of the browser extension.
- Day 15-20: Final testing, bug fixing.

Task Division

- Suvodeep: Topic Modeling - NLP processing
- Aayush - Frontend & Backend for Browser Extension