Development Track - Group 13

Names, email: Aayush Agarwal (aayush8@illinois.edu), Mukta Jaiswal (muktaj2@illinois.edu), Rudrik Patel(rudrikp2@illinois.edu), Palveet Kaur Saluja(psaluja2@illinois.edu), Suvodeep Saibal Sinha(sssinha2@illinois.edu)

Co-ordinator of the project: Suvodeep Saibal Sinha(sssinha2@illinois.edu)

Title: Context-Aware Bookmarking Tool (CABT)

Functions and Users:

The Context-Aware Bookmarking Tool (CABT) 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. CABT uses natural language processing (NLP) to analyze the content of saved web pages and the user's notes or highlights, facilitating automatic categorization 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.
- **Automatic Categorization:** Leveraging BERTopic to dynamically categorize bookmarks into topics or research areas.
- **Search by Concept**: Implementing semantic search techniques to allow retrieval of bookmarks based on related concepts, questions, or themes.

The primary users of CABT 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.

Significance

CABT 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, CABT improves the efficiency of managing and accessing digital resources. This tool has the potential to transform academic research by saving time, enhancing organization, and facilitating easier access to information.

Approach

To build CABT, 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 NLTK or spaCy will be integrated for content analysis and semantic search capabilities. We would like to compare and benchmark techniques like BERTopic, LDA, and other related topic-modeling methods
- **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 will use certain evaluation metrics with the most significant being:

Coherence Score: Coherence measures how interpretable and coherent the topics are. It evaluates the semantic similarity between high-scoring words within the same topic. Popular coherence measures include UMass, and Cosine Similarity

Perplexity: Perplexity measures how well the model predicts a sample. Lower perplexity indicates better performance.

Human Judgment: Human evaluation involves having human annotators assess the quality and interpretability of the topics generated by the model. This can be done through qualitative analysis or using metrics like topic uniqueness or relevance.

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, and launch.

Task Division

- Aayush & Suvodeep: Topic Modeling - NLP processing
- Mukta, Palveet & Rudrik - Frontend & Backend for Browser Extension