Intelligent Browsing Application: Indexing, Organizing, and Querying Collections of User-Selected Web Pages

Steve McHenry <mchenry7@illinois.edu> (Captain)

For this project, I propose an intelligent browsing application in the form of a Google Chrome extension which allows users to store web pages to user-defined collections which are first indexed and then available to be queried using text retrieval techniques studied in this course. Throughout the course of modern academic and professional research, users will often collect online documents (i.e., web pages) from the Internet for further review and reference. These documents can be acquired in many ways – for example, by discovery through use of a search engine, through recommendation by a colleague, or from within professional journals and whitepapers. Once collected, it is typically the burden of the user to devise some method of organization of the documents – often by in-browser bookmarks – with content recall and retrieval typically relying on the user's recollection of each document. By providing the user with the ability to store online documents, organize them into non-exclusive, user-defined collections, and query a selected subset of collections for a ranked listing of relevant documents, the research process can be significantly streamlined by eliminating time spent manually reviewing documents for specific desired information. This application draws heavily on text retrieval techniques and search engine design which are central themes in this course.

The central components of this application are the text extractor and tokenizer, the indexer and inverted index, and the ranking model. As users add web pages to collections, the text of the web page is extracted, tokenized, and added to the index. Each document within the index will be tagged with the collections to which it belongs so that it may be filtered from the query results based upon the user's selected collections for the query. The ranking model will implement Okapi BM25 with smoothing to rank documents based on user-provided queries. A seed dataset will be created by manually selecting documents from a variety of publicly available online resources such as Wikipedia, university/scholarly web sites, and major news organization web sites. These documents will be manually categorized into collections of like documents.

The application's functionality will be demonstrated by issuing queries against various subsets of collections and observing that the returned ranked lists of results are empirically desirable based upon both the queries and documents currently stored within the index. For acceptance testing, users may freely add (or remove) documents and collections to observe that those documents are returned as results for sufficiently relevant queries.

The application will be built and deployed as a Google Chrome extension. JavaScript will be used as the programming language for application back end. JavaScript, HTML, and CSS will be used to build the interactive and structural elements of the user interface.

The design, implementation, and testing phases of this project are expected to satisfy the 20-hour effort requirement (for a single person team). The estimated time requirement of each major task is listed below:

1. Design phase (2 hours)

- Determine the requirements and design for the continuous document text extraction and indexing component (1 hour)
- Determine the requirements and design for the retrieval ranking/scoring function against the inverted index (1 hour)

2. Implementation and unit testing phase (19 hours)
   - Implement the text extraction and tokenizer functionality (3 hours)
   - Implement the inverted index management functionality (6 hours)
   - Implement the inverted index retrieval functionality (3 hours)
   - Implement the smoothed document ranking functionality (4 hours)
   - Implement the user interface (3 hours)
3. Build a sample document collection and perform system testing (2 hours)

Total estimated project time: 23 hours.