Intelligent Browsing: Hacker News Jobs Super-Search

Overview

Quality of information retrieval and search continues to increase in importance as the internet becomes more and more integral to daily life. Although many websites contain and list valuable information, much of it is irrelevant to specific users because of significantly divergent interests. The aim of this application is to ameliorate this shortcoming on a specific website, "Hacker News" (https://news.ycombinator.com/), although the approach may be ported to other websites in future work.

Hacker News hosts a web page updated on a monthly basis that serves as an up-to-date repository of job openings in the software and technology industries. Users interested in such jobs may browse and apply for jobs they find through the page. Although it is technology-focused, the current ranking system for job postings is based on voting and does not take into account the preferences of the viewer. For example, a front-end software engineer may be shown back-end software engineer jobs at the top of the page.

Users may currently search the job postings page using a simple exact search function from the browser (control-f or ⌘-f key combinations), but this approach requires exact matching and that the user scroll through the page to find results.

The application introduced here is a chrome extension that affords the ability to intelligently browse the job postings via search. When the user clicks on the extension icon, they are taken to the most recent jobs page and prompted to enter search terms. After entering the search terms, the postings are re-ranked so that more relevant information is presented first.

Implementation

The application is written in TypeScript, JavaScript, HTML, and CSS with React as the main framework. Upon activating the extension, a search field is presented to the user and the browser is redirected to the most recent jobs postings page on Hacker News. Requests for the top level comments (consisting of job opportunities) are made through the HackerNews API (https://github.com/HackerNews/API) and the results stored in an array of HackerNews items. The array is then submitted to the WinkJS natural language processing library to form a set of documents.

Once the user enters and submits a search, the query is submitted to the WinkJS engine for processing through a BM25 ranking function, the documents are filtered and ranked according to the results, and the web page is re-sorted accordingly.

The source file structure is as follows:

- src > chrome > content.ts: Main entry point.
- src > chrome > hn-api.ts: HackerNews API implementation.
- src > chrome > job-postings.ts: Front-end UI sorting and re-sorting.
- src > chrome > bm25-search.js: Search implementation.

Installation

The extension can be installed via the following steps:

1. Download and install the chrome web browser from: https://www.google.com/chrome/
2. Download and unzip the build.zip file from the github repository: https://github.com/RahulSinghalChicago/CourseProject/blob/main/build.zip
3. Open the chrome browser and go to the extensions management interface.
4. Enable the developer mode toggle in the top right corner.
5. Click on the “load unpacked” button in the top left corner and select the unzipped build folder.
6. The extension is now available for use and should be listed in the “jigsaw piece” extensions dropdown in the top right corner of the chrome browser.

Usage

Once the extension icon is clicked and activated, the user need only enter a query into the search field and click the search button. The extension will then open a new browser tab, navigate to the latest jobs postings page on Hacker News, and re-rank and filter the postings according to the query. To see the original page, the jobs postings page can be reloaded and the postings will be listed as ranked by the website as opposed to the extension.

Team Member Contribution

Our team consists of Charlie Truong (ctruong4), David Newman (davidn4), and Rahul Singhal (rahuls11). Not including time spent on documentation, each team member spent more than 20 hours working on this project. The following is an approximate overview of the contributions of each team member:

Charlie Truong: Browser extension scaffolding, performance optimization
David Newman: HackerNews API
Rahul Singhal: WinkJS (BM25 search)

Additionally, all team members contributed to:

- Module integrations into the browser extension
- Testing (program correctness and evaluation)
- Research, strategy, and planning