CS 410 Project Documentation
Flexible Find Browser Extension
Kajetan Haas

Overview of the function of the code:

The project is a Firefox browser extension that can be used to search for text in webpages in a more flexible way. Traditional methods for searching for text on a webpage in a browser only search for exact matches of your query text, this can be very limiting.

This extension offers two search modes. The first is "Fuzzy Find", this searches the webpage using approximate string matching, for example if your query is "bread" it might match with "broad" since there is only a one-character difference. Another more useful example is if you search for "CS-410" it might match with "CS410" (no hyphen), which can be very useful since you do not always know exactly how text is written on the page.

The other search mode is "Relational Search", this searches the page for words that are related to your query word, like synonyms. For example, if you search for "exam" you will also get results matching "test", or if you search for "speech" you will get results for "language" and "lecture". This is helpful for searching a page for specific content when you are not sure how that content might be represented.

When you search for a query using one of these two modes, all matching terms are highlighted on the webpage and results are listed in the search window. The results are ranked first by their similarity to the query string and then by their order on the page. If you click on a result in the search window it will scroll to that result in the webpage and highlight it with a different color.

Documentation of how the software is implemented

The extension is implemented using HTML, JavaScript, and CSS. The code for the pop-up window is implemented in `search_bar.js` and `search_bar.html`. These files represent the UI and send messages to the file that initiates the searches. The main logic is in `find.js`, this is a content script that is loaded into any webpage that the extension is run on. It implements functions to gather all the text on the page, search the text using fuzzy or relational search, highlight matching text on the page, and receive commands from and send results to the pop-up window.

Two other projects are included in this project. The first is FuzzySet, implemented in `content_scripts/fuzzyset.js`, available here: https://github.com/Glench/fuzzyset.js. This includes the functionality for comparing two strings and finding how similar their structure is, it is used in the fuzzy find algorithm.

The second project is wordnet.js, available here: https://github.com/nlp-compromise/wordnet.js. This is a thin wrapper for querying a version of WordNet that has been converted to JSON. Wordnet.js is used for the relational search to find synonyms. The original wordnet.js dataset was overly large and contained data unnecessary to this application. The size of the dataset was slowing down the application significantly, so I stripped out all unnecessary data, reducing it to 13% of its original size, which resulted in great performance improvements.

Since the application uses Node modules it cannot be directly loaded as a browser extension. All module dependencies must be resolved first, so browserify is used to build a file that has all the required code bundled together. If you make changes to `find.js`, or any of the modules, you must rebundle the code with the following command at the root of the project directory: "browserify content_scripts/find.js -o content_scripts/bundle.js". This bundle.js file is what is actually loaded into the browser extension. Browserify is available here: https://browserify.org/.

Documentation of the usage of the software and instructions on how to install and run a software

To install the browser extension, first download the source code from GitHub by selecting Code -> Download ZIP. Unzip the contents of the ZIP file. Then enter "about:debugging" in the search bar of Firefox. Press the "This Firefox" button which should then show all installed extensions. Press the "Load Temporary Add-on…" button and navigate to and select the "manifest.json" file of this extension when prompted. The extension is now temporarily installed in the browser.

To use the extension, navigate to a webpage that you would like to search on. Click on the Flexible Find logo that should have appeared in the upper right corner of the browser, this should open a pop-up window allowing you to search. To use fuzzy find make sure the "Fuzzy Find" button is selected and then type a query in the search box, then press "Search". Matching text on the webpage should now be highlighted yellow and several buttons should appear in the pop-up below the search bar. Clicking on the buttons will highlight the matched term in magenta and scroll the matched term into view. To clear the highlights from the page press the "Clear Results" button.

To use relational search, click on the "Relational Search" button. Now enter a single word into the search bar and press "Search". If there are terms on the page that are related to the query, then they will be highlighted. Buttons will also appear in the pop-up that allow you to focus on a certain matched term.

Contribution of each team member

Only one member on the team.