This document is my final documentation for the course project for Text Information Systems.

Team

The registered team name is “Wonder from Down Under”. This is a single person team with captain and sole member **Alan Anderson**, with netID **alansa3**

Recap on project scope

The topic I chose was to write a browser extension to allow the user to search the text in the page using a more sophisticated algorithm than simple keyword ‘exact match’.

Often, a google search will yield pages with lengthy text, and there is merit in being able to quickly locate the points in the text relevant to the query. For instance, if we wanted to learn about the economic impacts of the War of the Roses, we might well find that useful information is buried in a long document on the War of the Roses. We could just search for the keyword ‘economic’, but we might get a better ranking of relevant sections by performing a BM25 search for a larger set of relevant terms.

Implementation

I implemented my project as a Chrome extension using Javascript, and using the BM25 algorithm. Once installed and active, the extension reacts to the user navigating to any page in the English Wikipedia domain: https://en.wikipedia.org/*

Please see the content.js file for in-line comments explaining the code. However, I will set out a brief summary of the implementation here.

Files

- The manifest.json file catalogues the elements of the plug-in, including the icons that represent it once installed in the browser and the script file that it runs. It also specifies the match pattern that identifies the sites that the browser should insert this plug-in into. In my case, I have limited this to the Wikipedia domain, which was the use case I initially proposed. But the code is generic and could apply to other domains with small adjustment (I do use a unique element id from Wikipedia pages to anchor the results at present).
- The images directory contains the icons for the plug-in.
- The scripts directory contains the content.js file, which is where my algorithm is implemented.

Implementation of algorithm

- Once triggered by the browser visiting a Wikipedia page, the script brings up a prompt into which the user enters search terms if they wish.
- It then looks for a <body> tag in the web page, and then collects the <p> paragraph elements within that body.

- The script creates an array of paragraphs, *textArray*, including a numerical index (required to facilitate reference back to the paragraph elements after sorting), the text content of the paragraph, the word count of each paragraph and a zero-initialized field for the BM25 score.
- It then populates two additional arrays to supply the required variables for the BM25 formula: a word count of each search term for each paragraph, and a record of how many paragraphs each term appears in at least once (to supply the IDF term).
- Next the script cycles through the paragraphs to calculate BM25 scores and populate the field in *textArray* with the scores.
- The script then sorts the *textArray* by BM25 score.
- Finally, it makes use of this sorted array to present the results, as described below.

Presentation of results

- The script inserts duplicates of the top ranked five paragraphs at the top of the page. Each of these appears with its ranking and BM25 score.
- Each of these top paragraphs is attached to a hypertext link to where that paragraph appears in the text below, to enable easy navigation.
- It also converts the paragraphs in the body of the text to bold format, to catch the eye as one scrolls down the document.

How to use the software

This is a very simple plug-in to use: install it in Chrome by , navigate to a Wikipedia page, and the prompt will collect your search terms. Local installation is as described by the Chrome documentation on plug-ins:

- https://developer.chrome.com/docs/extensions/mv3/getstarted/development-basics/

In short: click on the “jigsaw puzzle” icon on chrome to bring up the extensions tab and click on “Manage extensions”. Enable Developer Mode by clicking the toggle switch next to Developer mode. Click the **Load unpacked** button and select the local directory in which the extension files are located. The plug-in will be installed and will work automatically when you navigate to an English Wikipedia page.

Reflections on user experience

I have tried this extension on the sorts of use cases I initially envisaged, with success. By way of background, I am quite interested in history, and often peruse Wikipedia pages on historical events. But

for major events, these are very long, and it can be difficult to find the most relevant paragraphs. Some examples that I tested with my plug-in:

- Go to Wikipedia's "War of the Roses" page and search for "princes in the tower". The top-ranked two paragraphs are indeed the ones relating to the infamous murder of the two young princes, even though the search terms appear in many other paragraphs.
- Go to Wikipedia's "Manhattan Project" page and search for "Oppenheimer". His name naturally occurs many times in the page, but the top 5 paragraphs by BM25 score are indeed the most relevant for someone seeking to learn about his role.
- For non-historical use case, go to the Wikipedia page for "Quantum Mechanics" and search for "communication". The first paragraph describes the use of quantum entanglement for communications. In this instance, there is only one 'hit', so to be fair, a simple text search would have sufficed.

Scope for improvement

If I were to spend more time developing this plug-in, I would attempt a few enhancements, such as:

- Testing and adjusting for a wider range of domains beyond Wikipedia.
- Allowing the use to toggle between paragraphs and sentences as the relevant unit that we are mining for with BM25.
- Adjusting the user interface so that you proactively bring up the plug-in prompt, rather than automatically triggering it by visiting a particular domain.