Extractive Text Summary Project

**Main software: Python + NLTK**

**Web application: vanilla JS + HTML/CSS**

**Input: a body of text**

**Output: the sentence of the text which is most important**

**Main software**

1. Pre-processing
   1. Split into sentences
   2. Remove stop words
   3. Remove punctuation
2. Generating weighted frequencies of each sentence
   1. Calculate frequency of each word, and total number of words
   2. Calculate total weighted frequencies of each sentence (total score/no. words)
3. Producing output
   1. Return sentence with largest weighted frequency

**Wikipedia scraping**

* Make an algorithm which scrapes text from a Wikipedia article and summarises it using the above algorithm

**Web application**

* Create a UI using JavaScript + HTML/CSS

1. Create the basic page layout using HTML
   1. Header
   2. Text box to insert Wikipedia link
   3. Button to generate summary from Wikipedia link
   4. Text area to insert a block of text
   5. Button to generate summary from text
2. Style using CSS
3. Create functionality using JavaScript
   1. On clicking button, run the Python function
      1. Use the on click JavaScript function and link the JavaScript file to the HTML file
      2. Import the Python function to JavaScript
      3. Figure out how to display the resulting text summary on the web page

**To do:**

* Python: Make the no. sentences parameter optional, with a default value of a fifth (?) of the total number of sentences, to the nearest integer.
* HTML/CSS: Add text fields for the user to input no. sentences?
* Import python file to JS file so that I can generate summary
* Create web scraping algorithm

**Problems with this approach**

* Shorter words are more likely to appear in greater frequency, yet longer words may be more likely to carry more meaning (?)
* Frequency of a word’s appearance in text is not perfectly indicative of its importance