Matthew Bejtlich and Tong Zhang
2040
April 9, 2018
Part 3a
Report

1. How did your team keep on the same page? What problems did you face and what would you change in the future?

   We divided the project into sections, following the general flow described in the project instructions. We were individually responsible for specific sections, but often discussed together the approach on a whiteboard. We made sure to communicate frequently when we uploaded new versions of the code to GitHub. We faced no major problems. In the future, we would like to better use Github's branching functionality, as it may improve the workflow.

2. What sections of your Part 2 code did you need to change in order to add the PageRank scorer?

   We needed to modify the main() program in query.py. This contains our processing pipelines for each of the query types (e.g. one word, free text, Boolean). We set an optional flag using "argparse" and then use a set of "if" statements in the main() pipeline to determine which input was entered (PageRank or tf-idf). Depending on the desired scoring system, we would then compute and print the ranked page information for the given matched documents in the query. The goal of the new integration of PageRank in our main() for part 3a was to make use of the vector space title matching and printing functions. We did this using a set of conditional statements. We needed to write two new PageRank functions in query.py. One was to create a dictionary containing page and score, and the other was to rank just the subset of scores provided by the matched ids from the query. Both ranking systems utilize the matched ids (from Part 1).

3. Let PRi be the PageRank of the i-th page in the diagram below. Using the equation below, define PR1 and PR2 . Note: The only variables in your equation should be PageRank scores, PR, and the damping factor, d.

   $$\text{transition matrix } M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \end{bmatrix}$$

   because page1 and page2 are sinking pages, meaning there's no outbound links.
   so, we assume they are linking to every page.

   $$\text{transition matrix } M = \begin{bmatrix} 0 & 1/4 & 1/4 & 0 \\ 1/3 & 1/4 & 1/4 & 0 \\ 1/3 & 1/4 & 1/4 & 1 \\ 1/3 & 1/4 & 1/4 & 0 \end{bmatrix}$$

   PR1 is the initial state, which each page has equal probability to be clicked by an user.

```
PR1 = [1/4
           1/4
           1/4
           1/4]

According to the formula, PR2 = [(1-d)/4
                                                   (1-d)/4
                                                   (1-d)/4
                                                   (1-d)/4] + d*(M*PR1)
which is
PR2 = [(1-d)/4 + 0.125d,
          (1-d)/4 + 0.20833d,
          (1-d)/4 + 0.45833d,
          (1-d)/4 + 0.125d]
```

4. Include any known bugs or implementation details you'd like us to know about in your report

   - We let argparse throw an exception for cases when too few or too many input arguments are entered. Argparse naturally specifics the error. In a future version of the program, we could try to bypass argparser's messages to enter a custom error message for the user.