CS47300 Web Information Search and Management
HW3 Solutions

1. Web Crawling and Indexing

(a) Jaccard coefficient \( J(d_1, d_2) \):

\[
d_1 \cup d_2 = A, B, C, D, E, F
\]
\[
d_1 \cap d_2 = A, B, D
\]
\[
J(d_1, d_2) = \frac{3}{6} = \frac{1}{2}
\]

(b) Cosine similarity \( sim(d_1, d_2) \):

\[
d_1 = [1, 1, 1, 1, 1, 0]
\]
\[
d_2 = [1, 1, 0, 1, 0, 1]
\]
\[
|d_1| = \sqrt{5}, |d_2| = 2
\]
\[
sim(d_1, d_2) = \frac{d_1 \cdot d_2}{|d_1||d_2|} = \frac{3}{\sqrt{5} \times 2} \approx 0.671
\]

(c) Pros and cons of using Jaccard Coefficient.

Pros: Jaccard Coefficient is very efficient in finding mirror websites in duplicate detection. For example, if web page B simply copy web page A and repeat it for many times. Jaccard Coefficient of these two web pages is very high.

Cons: Jaccard Coefficient does not take term frequency into consideration. Rare terms are usually more valuable than frequent terms, but Jaccard Coefficient does not consider this information.

2. Web Crawling

(a) According to the question, we can get \( \lambda = 1/5, t = 10 \).

\[
Age(\lambda, t) = \int_{x=0}^{x=t} \lambda e^{-\lambda x} (t - x) dx
\]
\[
= \int_{x=0}^{x=10} 1/5 e^{-1/5 x} (10 - x) dx
\]
\[
= 2 \int_{x=10}^{x=0} e^{-1/5 x} dx - 1/5 \int_{x=10}^{x=0} e^{-1/5 x} x dx
\]
\[
= 5.677
\]

So the average age of the crawler is 5.677 days.

(b) 1). Age is continuous value. Freshness is binary value 0 and 1. When a web page is updated, freshness becomes 0 when it is crawled, it becomes 0. Freshness is the proportion of pages that are fresh. Some times freshness metric may lead to bad decisions of not crawling popular cites. But with age metric, it’s easier to decide when to crawl a website.

2). Sometimes freshness metric makes it crawling some web pages frequently, which violates politeness.
3. Mercator algorithm for balancing Politeness/Freshness

(a) Does one of the queues fill up?
   The queue won’t fill up, since the rate of crawling web pages and putting them into the system is the same.

(b) Do the F documents get crawled more often than S documents?
   Yes. Prioritizer decides the priority of urls. Urls are put into front queue $1 - k$. Suppose back queue uses round robin algorithms to select urls from front queue to process. The order will be $1 1 2 1 2 3 1 2 3 4$ and so on. And fresher urls will have higher priorities. So F can be crawled more often than S.

(c) Any other interesting behaviors? Whether fresher web pages can be crawled more frequently depends on the strategy that back queue selector. If it does not have priority to process urls from front queue with smaller index, all web pages may have same probability to be crawled.

4. PageRank

(a) Matrix Notation:
   $$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$
   $$B = \begin{bmatrix} 0/2 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$

(b) Initially, $R = \begin{bmatrix} 1/4, 1/4, 1/4, 1/4 \end{bmatrix}$
   $$R1 = B^T R = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 3/8 \\ 3/8 \\ 3/8 \\ 3/8 \end{bmatrix}$$
   $$R2 = B^T R1 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3/8 \\ 3/8 \\ 3/8 \\ 3/8 \end{bmatrix} = \begin{bmatrix} 3/8 \\ 3/8 \\ 3/8 \\ 3/8 \end{bmatrix}$$
   $R3 = R2$, because it converges.

5. PageRank and HITS

(a) Matrix Notation
   $$M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$
   $$B = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

(b) PageRank Coverage Eventually, it will become $PR(A) = 2PR(B) = 2PR(C)$. This is because A has a self-loop, and it’ll increase A’s
weight continuously. Actually if A has no out-link, A will become a sink and the Pagerank will converge as $PR(A) = 1$, the others as 0. We really need to remove self-loops in pagerank. Simply detecting and removing self-loop can be a method to solve this problem. Adding dampling factor can be another method.

(c) Matrix Notation

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Eventually, A will have the highest Hubness and Authority score. Because of the self-loop. C will also have a high hubness score, because it links to A. B will have a high authority score, because A links to B. To reduce the impact of self-loops, we need to detect and remove self-loops.

6. TF-IDF and cosine similarity

<table>
<thead>
<tr>
<th>TF</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>IDF</th>
<th>TF-IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>be</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>is</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>nobler</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>not</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>or</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>question</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>that</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>tis</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>to</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>whether</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

(a) $sim(Q,D1) = 0$

$sim(Q,D2) = \frac{2}{\sqrt{16}}$

$sim(Q,D3) = 0$

$sim(Q,D4) = 0$

(b) Rocchio formula: $\alpha = 1/3, \beta = 1/3, \gamma = 1/3$

$$q'_j = \alpha . q_j + \beta . \frac{1}{|Rel|} \sum_{D_i \in Rel} d_{ij} - \gamma . \frac{1}{|Nonrel|} \sum_{D_i \in Nonrel} d_{ij}$$

$$q' = 1/3q + 1/3D2 - 1/3(D1 + D3 + D4)$$

\[ \text{sim}(q, D1) = \frac{2/9 + 2/9}{\sqrt{17/3} \cdot \sqrt{2}} = 0.2287 \]
\[ \text{sim}(q, D2) = \frac{2/9 + 2 + 4/3 + 2/9}{\sqrt{17/3} \cdot \sqrt{2}} \]
\[ \text{sim}(q, D3) = \frac{-4/9 + 4}{\sqrt{17/3} \cdot \sqrt{16}} = -0.3233 \]
\[ \text{sim}(q, D4) = \frac{-4/9 + 3}{\sqrt{17/3} \cdot \sqrt{12}} = -0.2801 \]

You can also ignore negative values in \( q' \), or normalize \( q' \). These answers are all considered to be correct.

(c) \( \alpha = 0, \beta = 1, \gamma = 0 \)
\( q' = D2 = [1, 0, 0, 2, 2, 0, 0, 0, 1, 0] \)
\[ \text{sim}(q, D1) = \frac{1 + 1}{\sqrt{16} \cdot \sqrt{5}} = 1/\sqrt{5} \]
\[ \text{sim}(q, D2) = \frac{1 + 1 + 1 + 1}{\sqrt{16} \cdot \sqrt{10}} = 1 \]
\[ \text{sim}(q, D3) = \text{sim}(q, D4) = 0 \]

(d) Advantages: It is automatic method and it can work well. The new query is an expanded query, it increases the probability of finding documents about a topic instead of documents that only contain terms in a query.

(e) Disadvantages: Some times it will retrieves some non-relevant documents. Because some terms not in the original query will have positive value in the new query. This may cause some bad retrieved results.