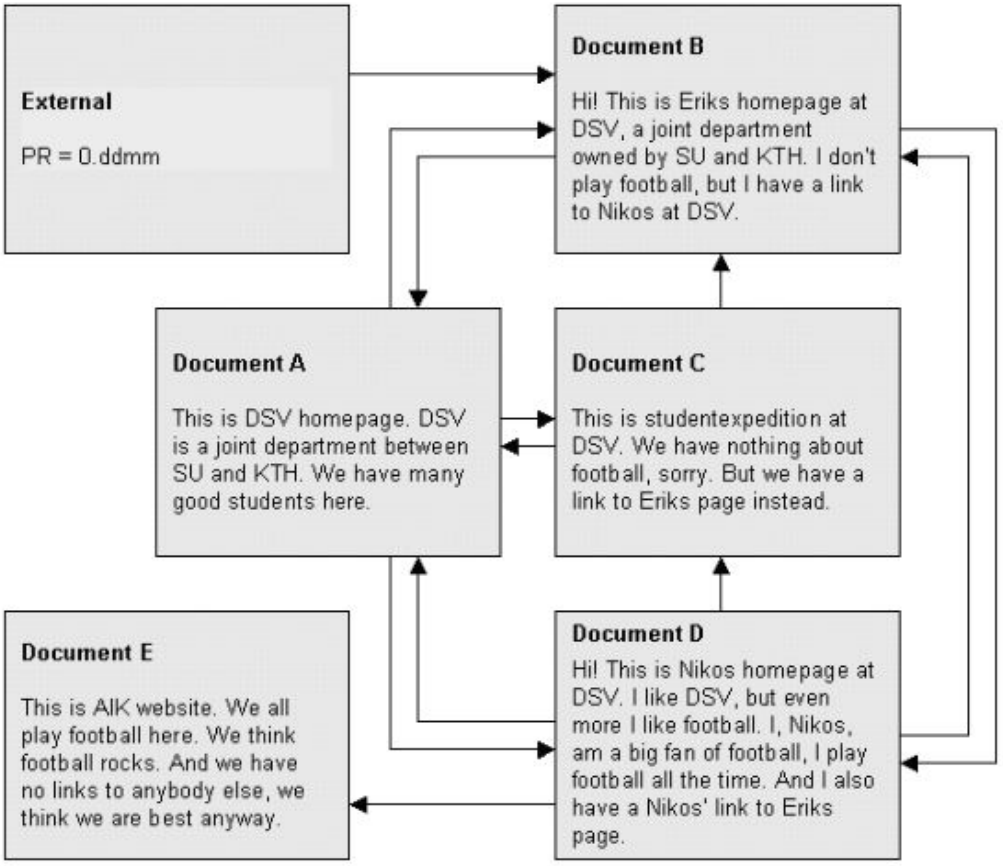
Davide Lagano
Georgios Patrikis
Björkeroth, Didrik
Eliasson, Robin

Assignment Group 03

Assignment 2: Impact of Page Rank

In this assignment we are going to use Text similarity and PageRank in order to rank the documents below.

Documents



Text Similarity

The query that we decided to use for this part was “DSV football”. By using this query we could extract text similarity for all documents. In the table below, we applied the formula for similarity (See Apendix for formula) in order to rank the documents based on their similarity score:

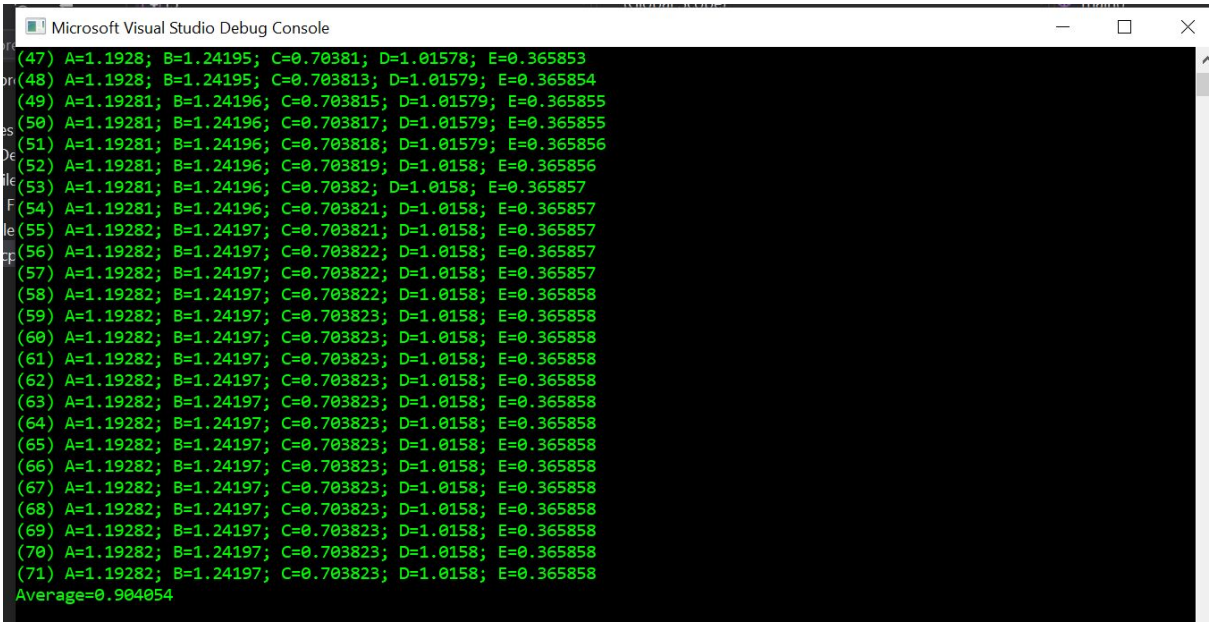
	Documents				
	A	B	C	D	E
DSV_q	1	1	1	1	1
DSV_d	2	2	1	2	0
$football_q$	1	1	1	1	1
$football_d$	0	1	1	3	2
N_d	19	29	22	40	27
sim(q,d)	**0,1053**	**0,1034**	**0,091**	**0,125**	**0,074**

PageRank

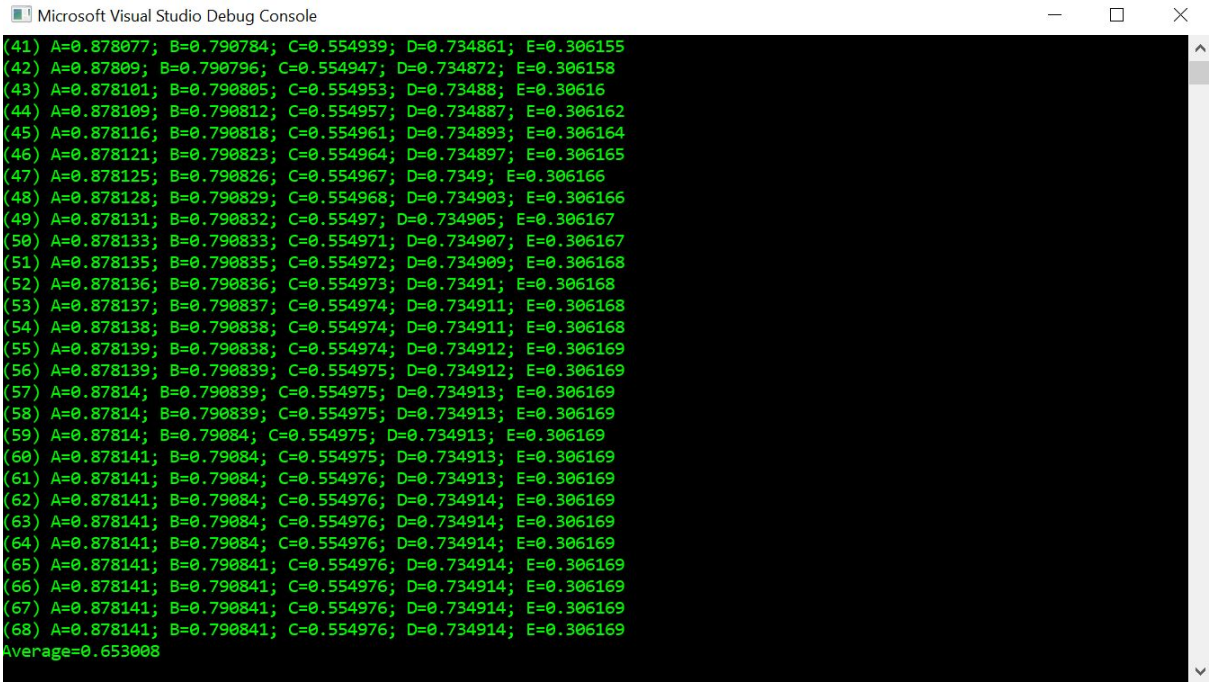
For each of the documents the PageRank was calculated, furthermore the external document was given the PageRank of the oldest person in the group according to the following value 0.ddmm, which resulted in the value 0.2812. The PageRank for each document is shown below, and the process from start to finish.

PR(A) = 0.15 + 0.85 * (PR(B) / 2 + PR(C) / 2 + PR(D) / 4)
PR(B) = 0.15 + 0.85 * (PR(A) / 3 + PR(C) / 2 + PR(D) / 4 + 0.2812)
PR(C) = 0.15 + 0.85 * (PR(A) / 3 + PR(D) / 4)
PR(D) = 0.15 + 0.85 * (PR(A) / 3 + PR(B) / 3)
PR(E) = 0.15 + 0.85 * (PR(D) / 4)

PageRank values



Verification of PageRank



Combining Text similarity with PageRank

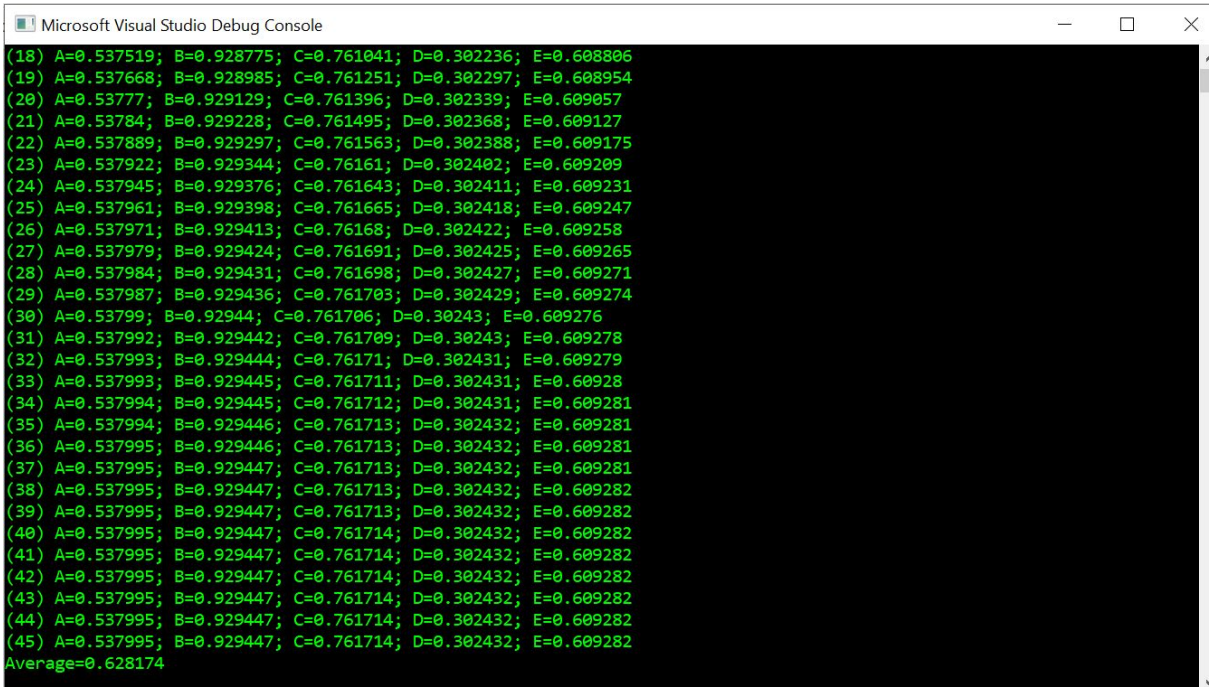
The documents were here reranked with a formula (see apendix) based on the initial Page Rank and the similarity.

Doc A: 0,105 + 0,5 * 1,19 = 0,7
Doc B: 0,103 + 0,5 * 1,24 = 0,723
Doc C: 0,091 + 0,5 * 0,7 = 0,441
Doc D: 0,125 + 0,5 * 1,02 = 0,635
Doc E: 0,074 + 0,5 * 0,37 = 0,259

Re-linking the documents

Since the PageRank of one document is dependant on the PageRank of the incoming links, we redirected the outgoing links of document with the highest PageRank to the ones we wanted boost. The following redirections was made, which was sufficient in order to increase the values of documents E and C.

1. Redirect link B → D to B → E
2. Redirect link B → A to B → C



Furthermore another formula had to be used (see Apendix), in order to calculate the new rank.

Doc A: 0,105 + 0,5 * 0,54 = 0,375
Doc B: 0,103 + 0,5 * 0,93 = 0,568
Doc C: 0,091 + 0,5 * 0,76 = 0,471
Doc D: 0,125 + 0,5 * 0,3 = 0,275
Doc E: 0,074 + 0,5 * 0,61 = 0,379

Rank	Initial page rank		New page rank		sim(q,d)		SIM1(q,d)		SIM2(q,d)	
	doc.id	PR value	doc.id	PR value	doc.id	sim value	doc.id	sim value	doc.id	sim value
1	B	1.24	B	0.93	D	0,125	B	0,72	B	0,57
2	A	1.19	C	0.76	A	0,105	A	0,7	C	0,47
3	D	1.02	E	0.61	B	0,103	D	0,64	E	0,379
4	C	0.70	A	0.54	C	0,091	C	0,44	A	0,375
5	E	0,37	D	0.30	E	0,074	E	0,26	D	0,28

Appendix:

Code for initial PageRank:

```
int main() {

	float a1, b1, c1, d1, e1;
	a1 = b1 = c1 = d1 = e1 = 0;
	cout << "A=" << a1 << "; B=" << b1 << "; C="
		<< c1 << "; D=" << d1 << "; E=" << e1 << "\r\n";

	for (int i = 0; i < 100; i++) {
		float a2 = 0.15 + 0.85 * (b1/2 + c1/2 + d1/4);
		float b2 = 0.15 + 0.85 * (a1/3 + c1/2 + d1/4 + 0.2812) ;
		float c2 = 0.15 + 0.85 * (a1/3 + d1/4);
		float d2 = 0.15 + 0.85 * (a1 / 3 + b1/2);
		float e2 = 0.15 + 0.85 * (d1/4);
		if (a1 == a2 && b1 == b2 && c1 == c2 && d1 == d2 && e1 == e2) { /*when convergence --> Break*/
			break;
		}
		a1 = a2;
		b1 = b2;
		c1 = c2;
		d1 = d2;
		e1 = e2;
		cout << "(" << i << ") " << "A=" << a1
			<< "; B=" << b1 << "; C=" << c1
			<< "; D=" << d1 << "; E=" << e1 << "\r\n";
	}
	cout << "Average=" << (a1 + b1 + c1 + d1 + e1) / 5
		<< "\r\n";

	return 0;
}
```

Code for re-linked PageRank:

```
int main() {

	float a1, b1, c1, d1, e1;
	a1 = b1 = c1 = d1 = e1 = 0;
	cout << "A=" << a1 << "; B=" << b1 << "; C="
		<< c1 << "; D=" << d1 << "; E=" << e1 << "\r\n";

	for (int i = 0; i < 100; i++) {
		float a2 = 0.15 + 0.85 * (c1/2 + d1/4);
		float b2 = 0.15 + 0.85 * (a1/3 + c1/2 + d1/4 + 0.2812/1) ;
		float c2 = 0.15 + 0.85 * (a1/3 + d1/4 + b1 / 2);
		float d2 = 0.15 + 0.85 * (a1 / 3);
		float e2 = 0.15 + 0.85 * (d1/4 + b1 / 2);
		if (a1 == a2 && b1 == b2 && c1 == c2 && d1 == d2 && e1 == e2) { /*when convergence --> Break*/
			break;
		}
```

```
                a1 = a2;
                b1 = b2;
                c1 = c2;
                d1 = d2;
                e1 = e2;
                cout << "(" << i << ") " << "A=" << a1
                        << "; B=" << b1 << "; C=" << c1
                        << "; D=" << d1 << "; E=" << e1 << "\r\n";
        }
        cout << "Average=" << (a1 + b1 + c1 + d1 + e1) / 5
                << "\r\n";

        return 0;
}
```

Formula for text similarity:

$$sim(q,d) = \frac{\sum_{i=1}^{n} q_i \cdot d_i}{N_d}$$

Formula for combined Text Similarity and Page Rank

$SIM_1(q, d) = sim(q, d) + 0.5 \cdot PR_{\text{initial}}(d)$

Formula for the final combined Text Similarity and Page Rank

$SIM_2(q, d) = sim(q, d) + 0.5 \cdot PR_{\text{new}}(d)$