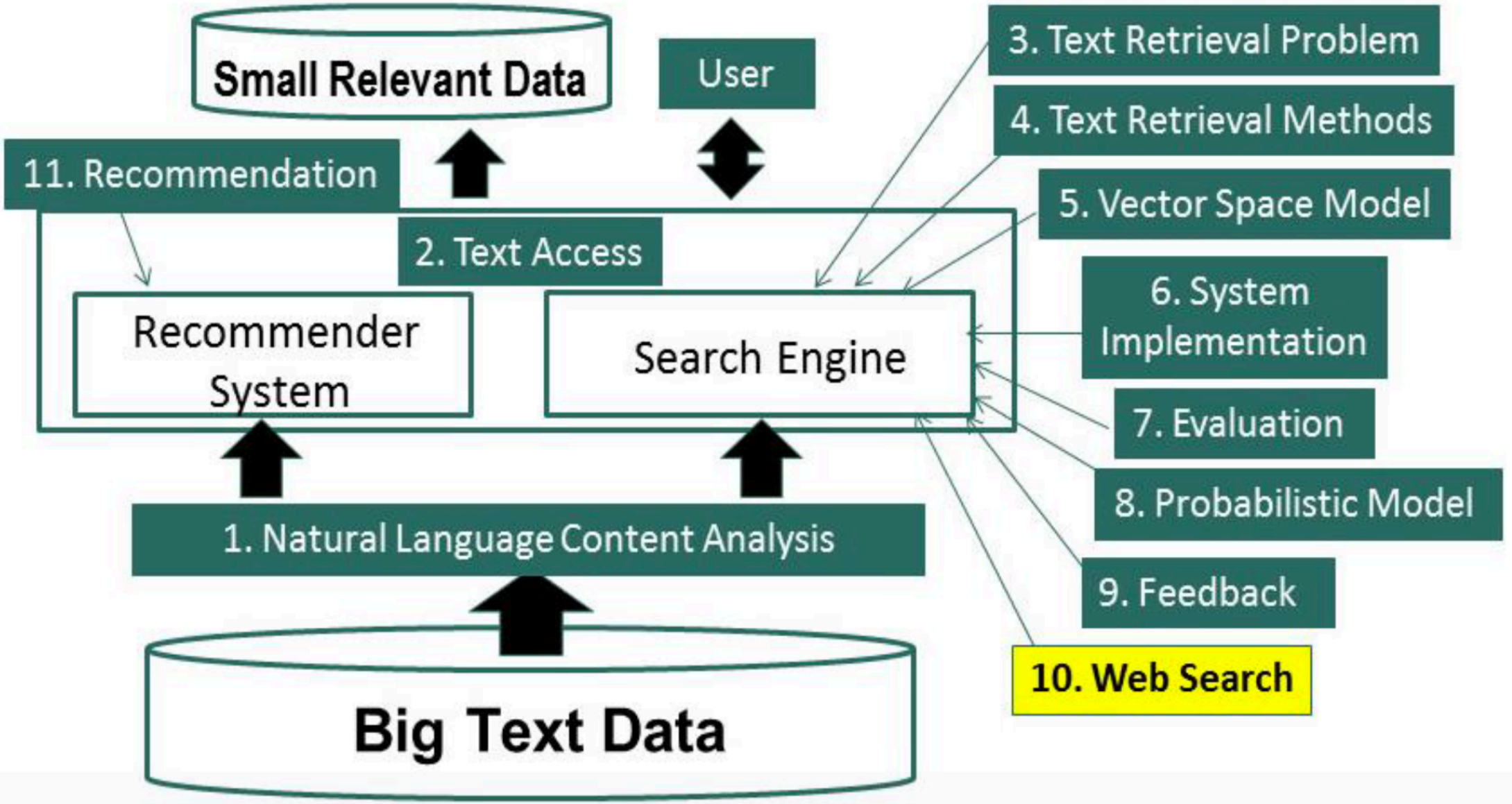
Information Retrieval & Text Mining

Web Search: Page Rank

Dr. Saeed Ul Hassan
Information Technology University

Course Schedule



Page Rank Algorithm

- PageRank is a “vote”, by all the other pages on the Web, about how important a page is.
- A link to a page counts as a vote of support
- The original PageRank algorithm was designed by Lawrence Page and Sergey Brin.

Random Surfer Model

PageRank as a model of user behaviour, where a surfer clicks on links at random with no regard towards content.

$$PR(A) = (1-d) + d(PR(Ti)/C(Ti) + \ldots + PR(Tn)/C(Tn))$$

PR(A) is the PageRank of page A,

PR(Ti) is the PageRank of pages Ti which link to page A,

C(Ti) is the number of outbound links on page Ti and

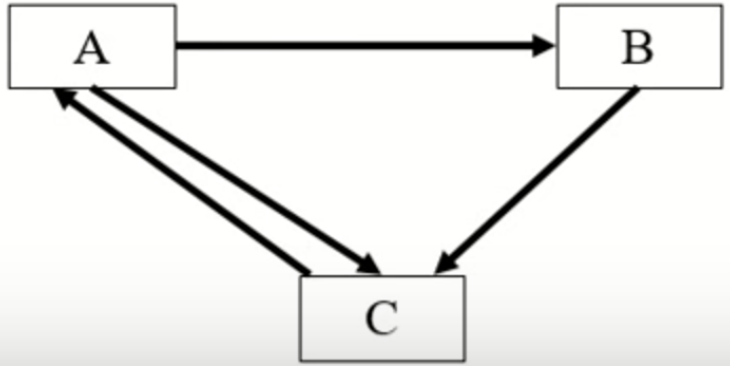
d is a damping factor which can be set between 0 and 1.

How Page Rank is Calculated

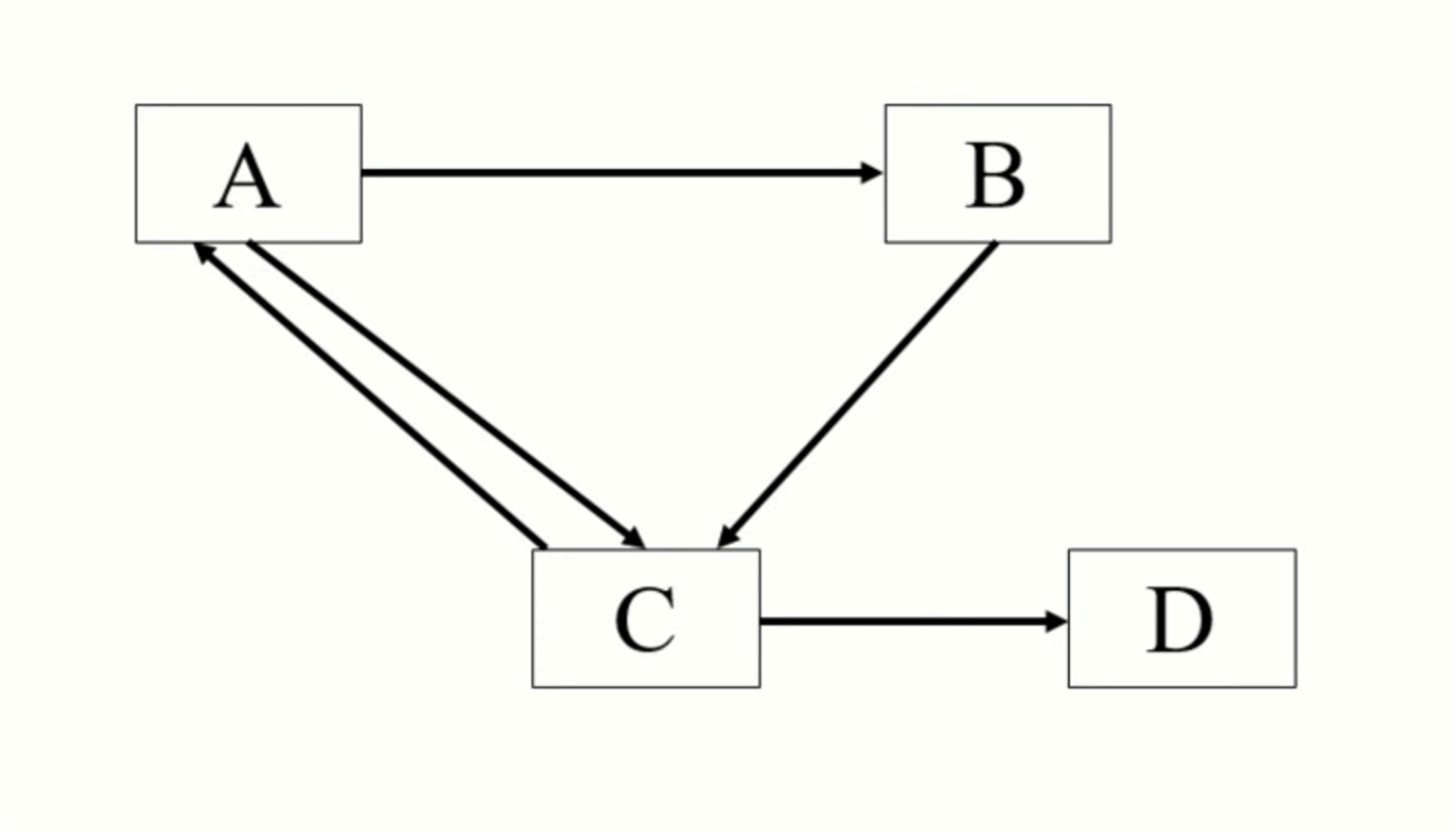
- The rank of a document is given by the rank of those documents which link to it.
- The PR of each page depends on the PR of the pages pointing to it.
- But we won't know what PR those pages have until the pages pointing to them have their PR calculated and so on.

Inbound, Outbound and Dangling Links

- Inbound link for a web page always increases that page's PageRank.
- An important aspect of outbound links is the lack of them on web pages.
- When a web page has no outbound links, its PageRank cannot be distributed to other pages.
- Lawrence Page and Sergey Brin characterise links to those pages as dangling links.

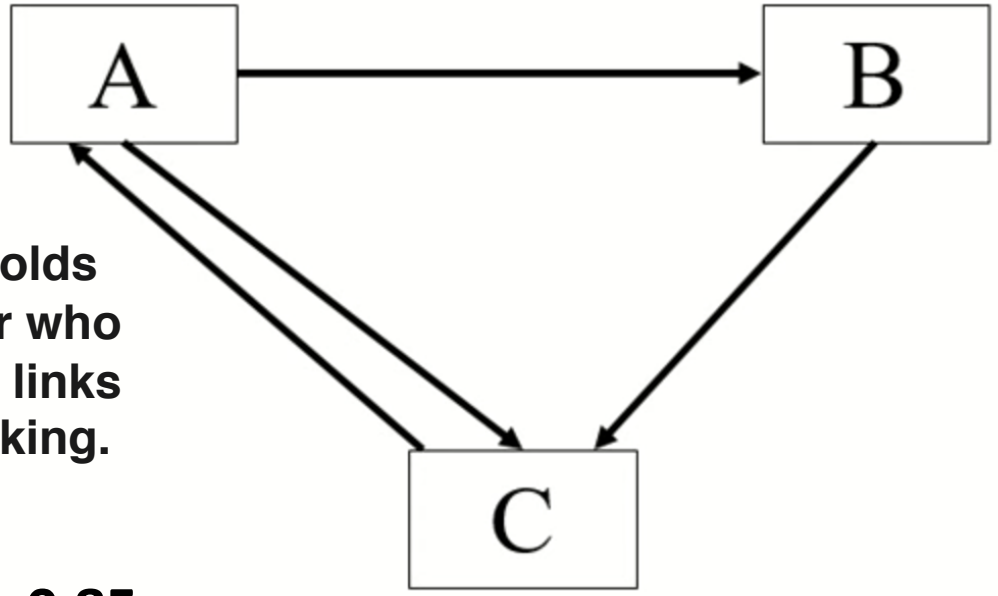


Dangling Links



Page Rank Initialization

- Initially Page Rank (PR) for all the web pages = 1



The PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking.

Damping Factor = d= 0.85

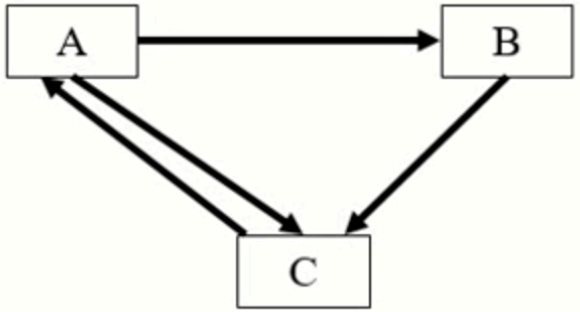
$$PR(A) = (1\text{-}d) + d(PR(Ti)/C(Ti) + \ldots + PR(Tn)/C(Tn))$$

Page Rank of A & B

- Initially Page Rank (PR) for all the web pages = 1

$$PR(A) = (1-d) + d(PR(Ti)/C(Ti) + \ldots + PR(Tn)/C(Tn))$$

$$PR(A) = (1-d) + d\,[\,PR(C) / C(C)\,]$$
$$= (1-0.85) + 0.85\,[\,1/1\,]$$
$$= 0.15 + 0.85[\,1\,]$$
$$= 0.15 + 0.85$$
$$= 1$$

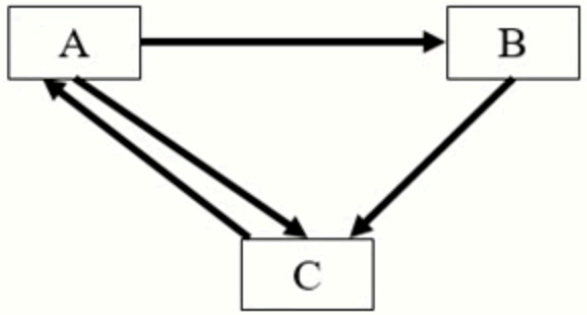


$$PR(B) = (1-d) + d\,[\,PR(A) / C(A)]$$
$$= (1-0.85) + 0.85\,[\,(1) / 2]$$
$$= 0.15 + 0.85\,[0.5]$$
$$= 0.15 + 0.425$$
$$= 0.575$$

Page Rank of C

- Initially Page Rank (PR) for all the web pages = 1

$$PR(A) = (1-d) + d(PR(Ti)/C(Ti) + \ldots + PR(Tn)/C(Tn))$$

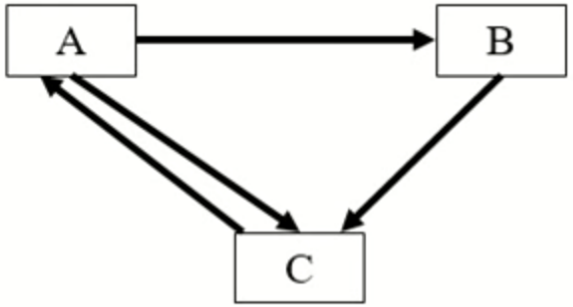


$$PR(C) = (1-d) + d\,[\,PR(A) / C(A) + PR(B) / C(B)\,]$$
$$= (1-0.85) + 0.85\,[\,(1/2) + (0.575 / 1)]$$
$$= 0.15 + 0.85[\,0.5 + 0.575]$$
$$= 0.15 + 0.85\,[\,1.075]$$
$$= 0.15 + 0.91375$$
$$= 1.06375$$

Iterations

- Initially Page Rank (PR) for all the web pages = 1

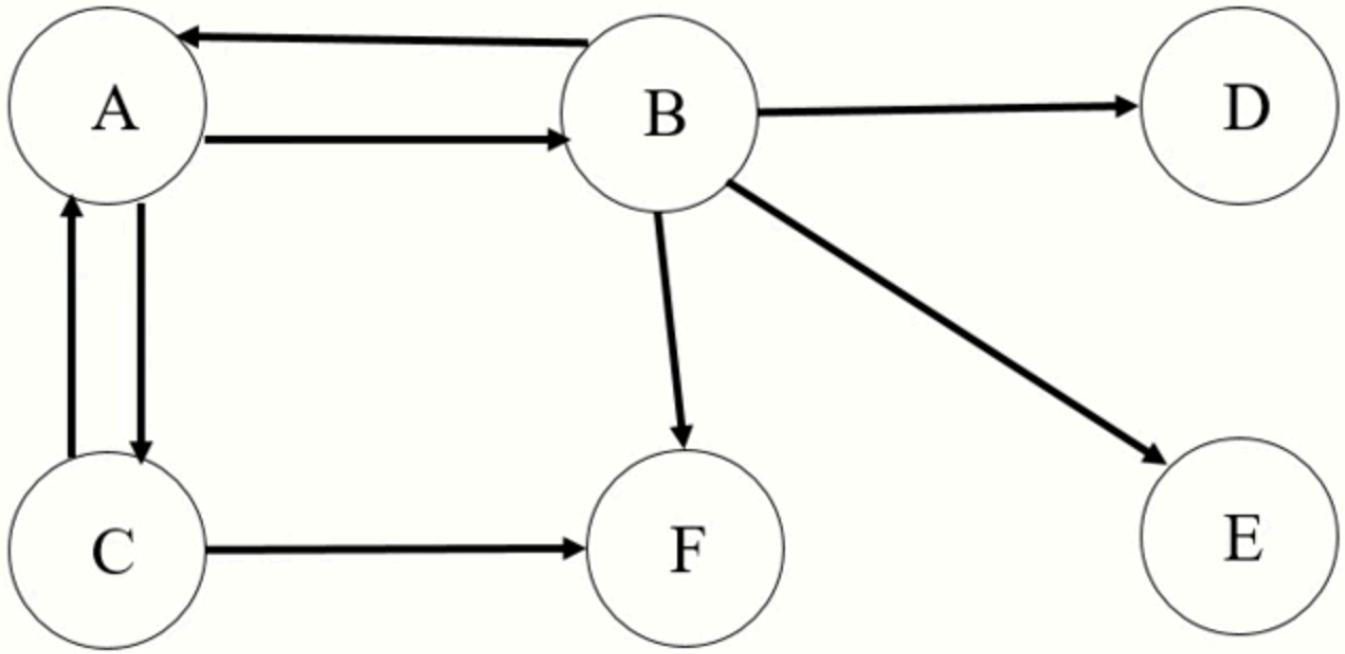
$$PR(A) = (1-d) + d(PR(Ti)/C(Ti) + \ldots + PR(Tn)/C(Tn))$$



Iteration	A	B	C
0	1	1	1
1	1	0.575	1.06375
2	1.0541875	0.5980296875	1.06354922

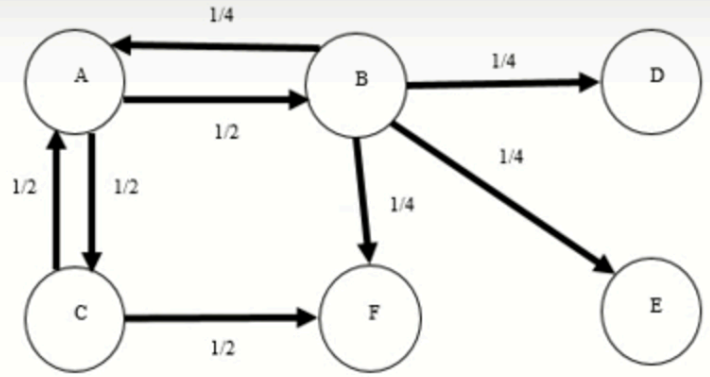
Page Rank using Matrix

- Initially Page Rank (PR) for all the web pages = 1



Matrix Representation

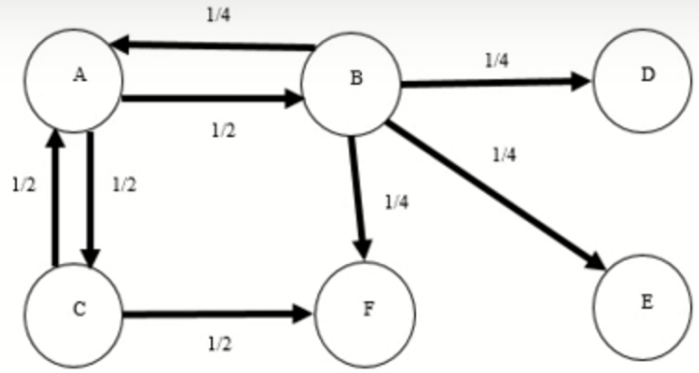
- Teleport Factor = 0.8



	A	B	C	D	E	F
A	0	1/2	1/2	0	0	0
B	1/4	0	0	1/4	1/4	1/4
C	1/2	0	0	0	0	1/2
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

Matrix Transpose

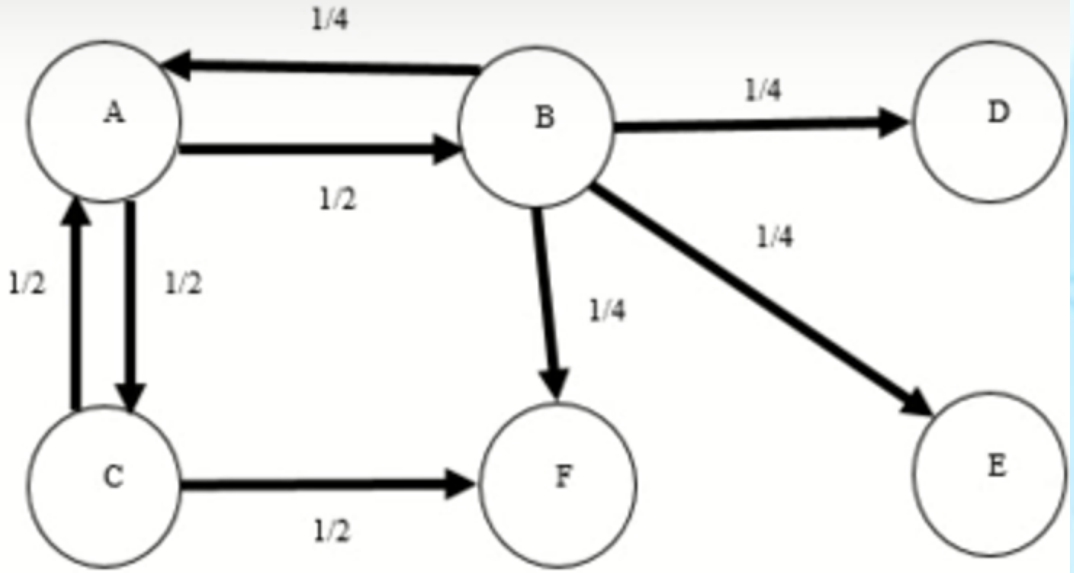
- Teleport Factor = 0.8



M^T

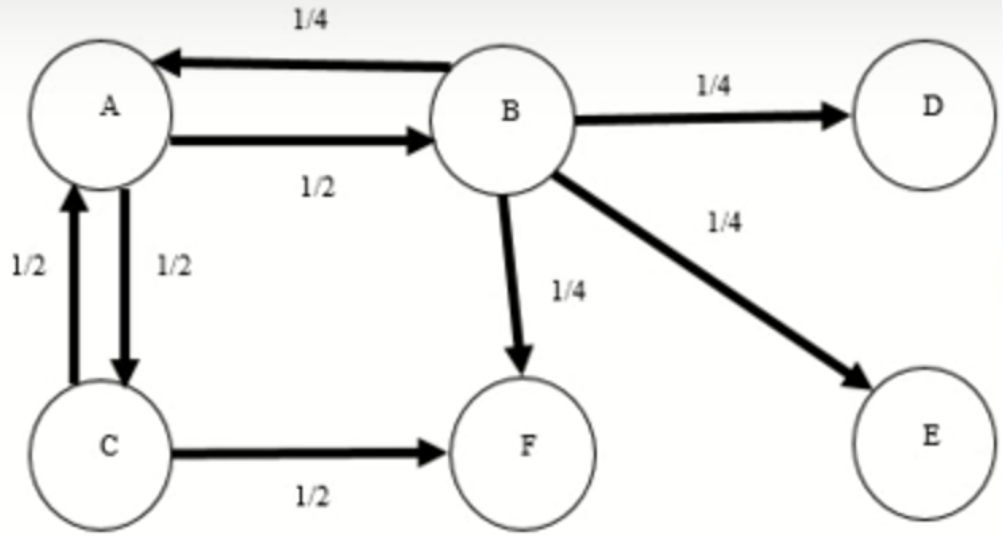
	A	B	C	D	E	F
A	0	1/4	1/2	0	0	0
B	1/2	0	0	0	0	0
C	1/2	0	0	0	0	0
D	0	1/4	0	0	0	0
E	0	1/4	0	0	0	0
F	0	1/4	1/2	0	0	0

- Teleport Factor = 0.8



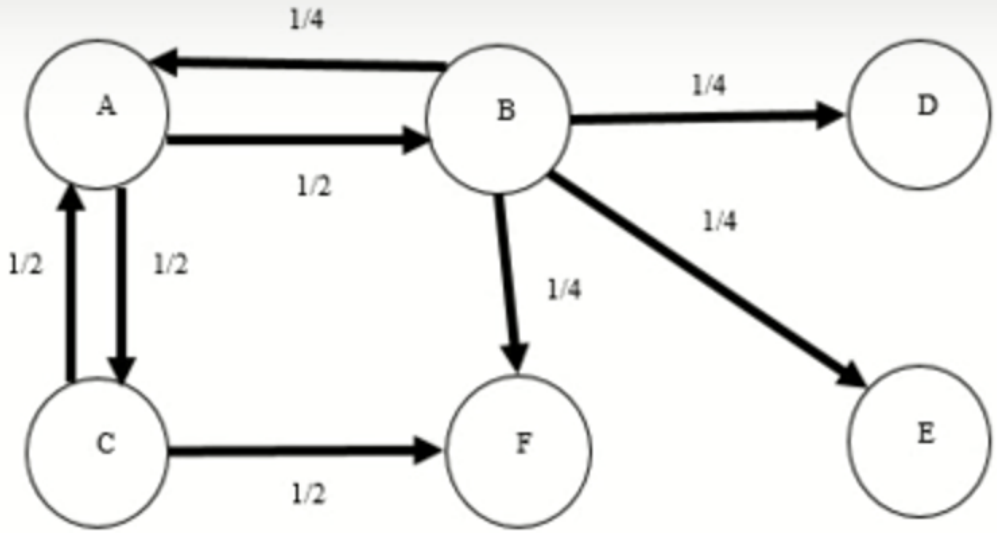
$M^1 =$

$$\begin{pmatrix} 0 & 1/4 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.4 \\ 0.4 \\ 0.2 \\ 0.2 \\ 0.6 \end{pmatrix}$$



$M^2 =$

$$\begin{pmatrix} 0 & 1/4 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0.6 \\ 0.4 \\ 0.4 \\ 0.2 \\ 0.2 \\ 0.6 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.1 \\ 0.1 \\ 0.3 \end{pmatrix}$$



$M^3 =$

$$\begin{pmatrix} 0 & 1/4 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.1 \\ 0.1 \\ 0.3 \end{pmatrix} = \begin{pmatrix} 0.225 \\ 0.15 \\ 0.15 \\ 0.075 \\ 0.075 \\ 0.225 \end{pmatrix}$$

Next Generation Search Engines

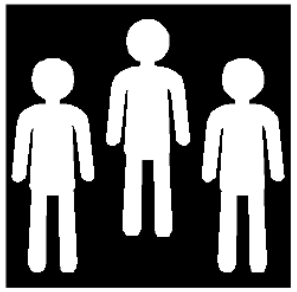
- More specialized/customized (vertical search engines)
  - Special group of users (community engines, e.g., Citeseer)
  - Personalized (better understanding of users)
  - Special genre/domain (better understanding of documents)
- Learning over time (evolving)
- Integration of search, navigation, and recommendation/filtering (full-fledged information management)
- Beyond search to support tasks (e.g., shopping)
- Many opportunities for innovations!

The Data-User-Service (DUS) Triangle

Data

Web pages
News articles
Blog articles
Literature
Email
…

Users



Lawyers
Scientists
Online shoppers
…
…

Services

Search
Browsing
Mining
Task support
…

Future Intelligent Information Systems