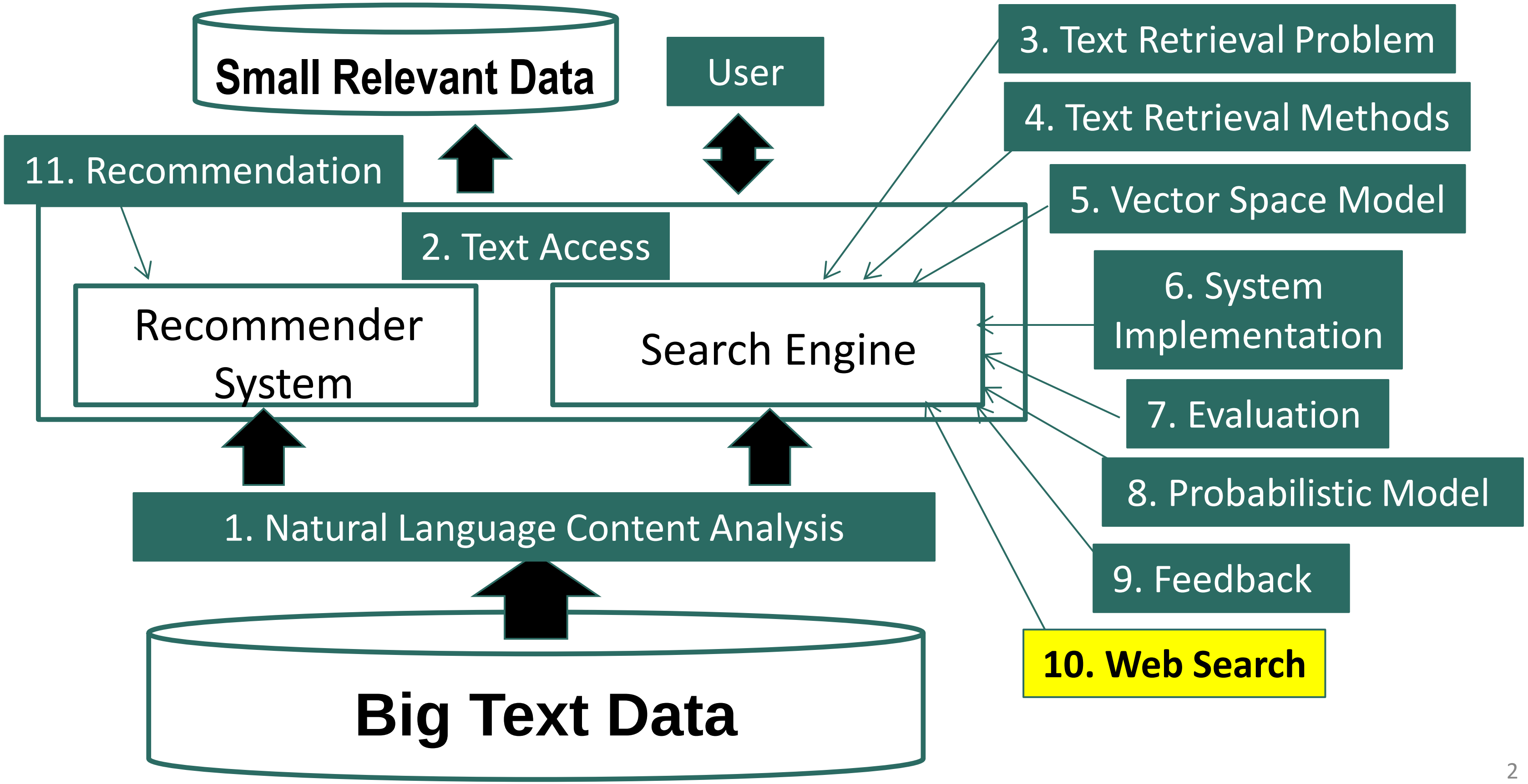
Text Retrieval and Search Engines

Web Search: Link Analysis - Part 1 - 3

ChengXiang "Cheng" Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

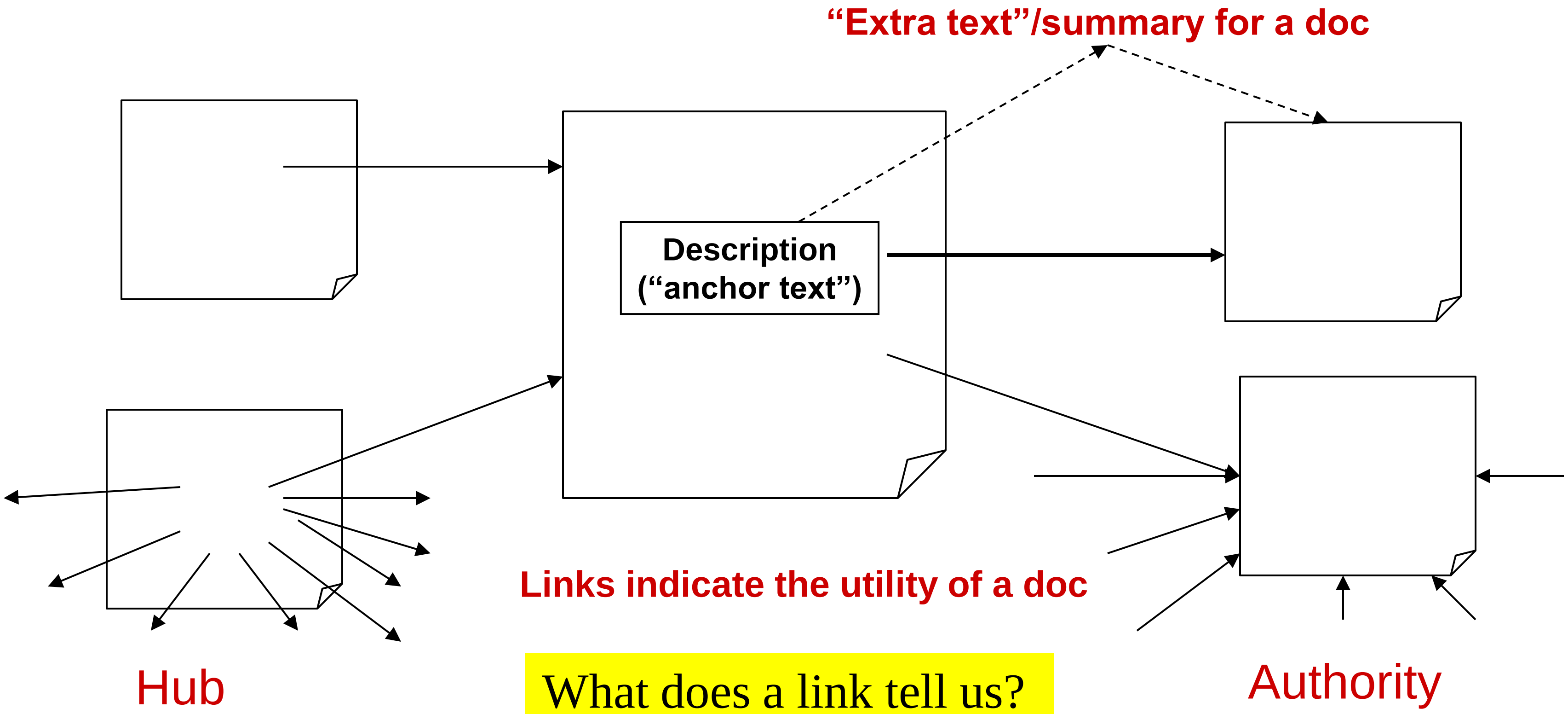
Web Search: Link Analysis



Ranking Algorithms for Web Search

- Standard IR models apply but aren't sufficient
  – Different information needs
  – Documents have additional information
  – Information quality varies a lot
- Major extensions
  – Exploiting links to improve scoring
  – Exploiting clickthroughs for massive implicit feedback
  – In general, rely on machine learning to combine all kinds of features

Exploiting Inter-Document Links



What does a link tell us?

PageRank: Capturing Page "Popularity"

- Intuitions
  - Links are like citations in literature
  - A page that is cited often can be expected to be more useful in general
- PageRank is essentially "citation counting", but improves over simple counting
  - Consider "indirect citations" (being cited by a highly cited paper counts a lot…)
  - Smoothing of citations (every page is assumed to have a non-zero pseudo citation count)
- PageRank can also be interpreted as random surfing (thus capturing popularity)

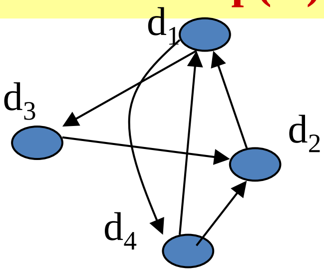
The PageRank Algorithm

Random surfing model: At any page,

With prob. α, randomly jumping to another page

With prob. $(1-\alpha)$, randomly picking a link to follow.

$p(d_i)$: PageRank score of d_i = average probability of visiting page d_i



Transition matrix

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

M_{ij} = probability of going from d_i to d_j

$$\sum_{j=1}^{N} M_{ij} = 1$$

probability of visiting page dj at time t+1

probability of at page di at time t

"Equilibrium Equation":

$$p_{t+1}(d_j) = \underbrace{(1-\alpha)\sum_{i=1}^{N} M_{ij}\, p_t(d_i)}_{\text{Reach dj via following a link}} + \underbrace{\alpha \sum_{i=1}^{N} \tfrac{1}{N}\, p_t(d_i)}_{\text{Reach dj via random jumping}}$$

N= # pages

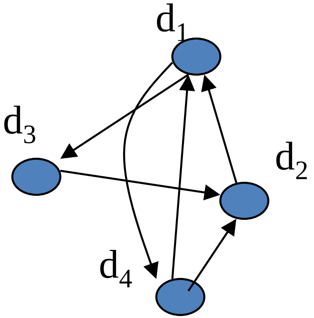
dropping the time index

$$p(d_j) = \sum_{i=1}^{N} [\tfrac{1}{N}\alpha + (1-\alpha) M_{ij}]\, p(d_i) \quad \Rightarrow \quad \vec{p} = (\alpha I + (1-\alpha)M)^T \vec{p}$$

I_{ij} = 1/N

We can solve the equation with an iterative algorithm

PageRank: Example



$$p(d_j) = \sum_{i=1}^{N} [\tfrac{1}{N}\alpha + (1-\alpha)M_{ij}]\, p(d_i)$$

$$\vec{p} = (\alpha I + (1-\alpha)M)^T\, \vec{p}$$

$$A = (1-0.2)M + 0.2I = 0.8 \times \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix} + 0.2 \times \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

$$\begin{bmatrix} p^{n+1}(d_1) \\ p^{n+1}(d_2) \\ p^{n+1}(d_3) \\ p^{n+1}(d_4) \end{bmatrix} = A^T \begin{bmatrix} p^n(d_1) \\ p^n(d_2) \\ p^n(d_3) \\ p^n(d_4) \end{bmatrix} = \begin{bmatrix} 0.05 & 0.85 & 0.05 & 0.45 \\ 0.05 & 0.05 & 0.85 & 0.45 \\ 0.45 & 0.05 & 0.05 & 0.05 \\ 0.45 & 0.05 & 0.05 & 0.05 \end{bmatrix} \times \begin{bmatrix} p^n(d_1) \\ p^n(d_2) \\ p^n(d_3) \\ p^n(d_4) \end{bmatrix}$$

$$p^{n+1}(d_1) = 0.05 * p^n(d_1) + 0.85 * p^n(d_2) + 0.05 * p^n(d_3) + 0.45 * p^n(d_4)$$

Initial value p(d)=1/N, iterate until converge

Do you see how scores are propagated over the graph?

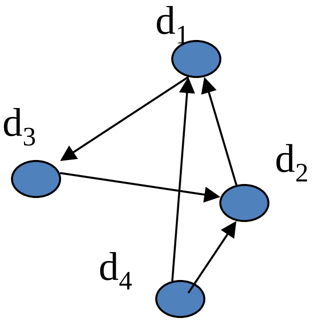
PageRank in Practice

- Computation can be quite efficient since M is usually sparse
- Normalization doesn’t affect ranking, leading to some variants of the formula
- The zero-outlink problem: p(di)’s don’t sum to 1
  - One possible solution = page-specific damping factor (α=1.0 for a page with no outlink)
- Many extensions (e.g., topic-specific PageRank)
- Many other applications (e.g., social network analysis)

HITS: Capturing Authorities & Hubs

- Intuitions
  - Pages that are widely cited are good authorities
  - Pages that cite many other pages are good hubs
- The key idea of HITS (Hypertext-Induced Topic Search)
  - Good authorities are cited by good hubs
  - Good hubs point to good authorities
  - Iterative reinforcement…
- Many applications in graph/network analysis

The HITS Algorithm



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

“Adjacency matrix”

Initial values: $a(d_i)=h(d_i)=1$

$$h(d_i) = \sum_{d_j \in OUT(d_i)} a(d_j)$$

$$a(d_i) = \sum_{d_j \in IN(d_i)} h(d_j)$$

Iterate

Normalize:

$$\sum_i a(d_i)^2 = \sum_i h(d_i)^2 = 1$$

$$\vec{h} = A\vec{a}; \qquad \vec{a} = A^T \vec{h}$$

$$\vec{h} = AA^T \vec{h}; \quad \vec{a} = A^T A\vec{a}$$

Summary

- Link information is very useful
  - Anchor text
  - PageRank
  - HITS
- Both PageRank and HITS have many applications in analyzing other graphs or networks