Content

III. Text Transformation

- Text Statistics
- Parsing Documents
- Information Extraction
- Link Analysis

Link Analysis
Hyperlinks

The web is a network of documents induced by hyperlinks:

> This web page `is perhaps the most famous example there ever was.`

Hyperlinks refer readers of a web page to another. There can be but one reason for adding a hyperlink to a web page:

> The author believes the linked page important to be reachable.

A hyperlink is usually attached to, or in the vicinity of a text or an image found on a web page that explains the linked page's relevance, e.g., by summarizing it.

These properties of hyperlinks can be exploited for web search.
Never trust user input:

- Omit hyperlinks that can be created by users of a web page.
- Omit hyperlinks that originate from malicious pages.
- Omit hyperlinks that are added by default to a web page.

Link Analysis

Anchor Text

The web is a network of documents induced by hyperlinks (HTML source code):

```
<a href="http://www.example.com">This web page</a> is perhaps
the most famous example there ever was.
```

The text enclosed by an HTML anchor element is called anchor text. It forms the clickable part of a hyperlink, redirecting to the URL given in the `href` attribute.

Anchor texts, and optionally their surrounding passages (e.g., sentence or paragraph) are used as a source of index terms for the linked page.

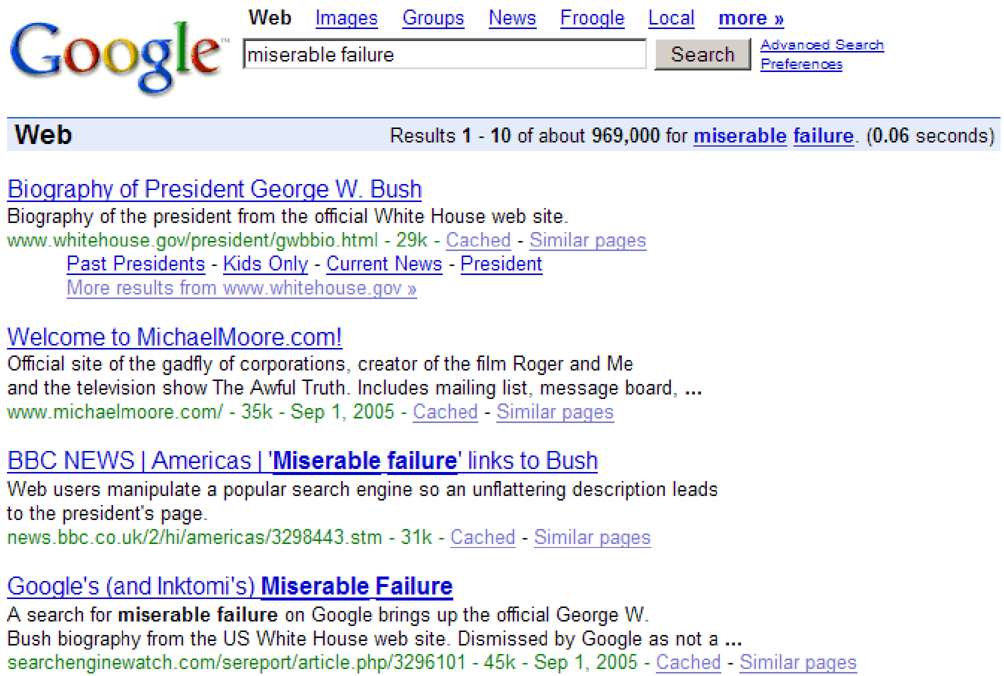
Anchor texts provide for index terms not necessarily found on the linked web page, severely improving retrieval performance.

Never trust user input: This may be misused, e.g., to give web pages a bad name.

An anchor text processing pipeline will include a customized stop word list, including words such as `page`, `here`, `click`.

Remarks:

- The term Google bomb refers to the practice of causing a website to rank highly in web search engine results for irrelevant, unrelated or off-topic search terms by linking heavily.



[Wikipedia]

Link Analysis

PageRank [Brin 1998]

Links between web pages may be used to gauge web page importance: The more links point to a web page, the more important it must be.

Naive importance measure for a web page A:

$$\textit{importance}(A) = |\{B \mid B \text{ is a web page} \wedge B \rightarrow A\}|,$$

where $B \rightarrow A$ indicates that B links to A.

Problems:

- every link counts equally much
- every web page can have an arbitrary number of links to other web pages

Desirable properties:

- the importance of A should depend on that of pages linking to it
- the importance of B should be shared by the pages it links to, not multiplied

➜ Meet the random surfer model

Link Analysis

PageRank: Random Surfer Model [Brin 1998]

Link Analysis

PageRank: Random Surfer Model [Brin 1998]

The PageRank of web page A is the probability that a random surfer will look at A.

Random surfing:

1. Open a random web page
2. Choose $\alpha \in [0, 1]$ at random
3. If $\alpha < \lambda$: go to Step 1
4. If the current page has no links: go to Step 1
5. Else: follow a random link, then go to Step 2

Observations:

- Random surfing has the Markov property.
- Steps 2-4 ensure the surfer does not get stuck, and that every page has a non-zero chance of being visited.
- Empirically, $\lambda = 0.15$.

PageRank: Definition [Brin 1998]

Given a page u, its PageRank is computed as follows:

$$PR(u) = \lambda \cdot \frac{1}{n} + (1-\lambda) \cdot \sum_{v \in B_u} \frac{PR(v)}{L_v},$$

where n is the number of web pages, B_u is the set of pages linking to u, and L_v the number of outgoing links on page v.

Algebraic formulation: Let $\mathbf{T}$ denote the matrix of page transition probabilities, so that the probability of transitioning from page i to j is given by:

$$\mathbf{T}_{ij} = \lambda \cdot \frac{1}{n} + (1-\lambda)\frac{1}{L_i}.$$

Then $\mathbf{r}$ is the vector of page probabilities at time t of executing the random surfing process when repeatedly multiplying it with $\mathbf{T}$:

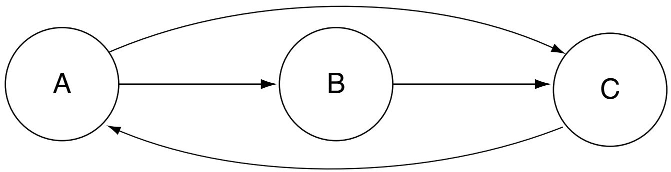
$$\mathbf{r} \cdot \mathbf{T}^t$$

As $t \to \infty$, $\mathbf{r}$ yields the PageRanks for all pages, which corresponds to the principal eigenvector of $\mathbf{T}$.

Since T is stochastic, irreducible, and aperidodic, this process converges.

Link Analysis

PageRank: Example



$$\mathbf{T} = \begin{bmatrix} 0.05 & 0.475 & 0.475 \\ 0.05 & 0.05 & 0.9 \\ 0.9 & 0.05 & 0.05 \end{bmatrix}$$

$t = 0$: $\quad \mathbf{r} \cdot \mathbf{T}^t = [1, 0, 0]$

$t = 1$: $\quad \mathbf{r} \cdot \mathbf{T}^t = [0.05, 0.475, 0.475]$

$t = 2$: $\quad \mathbf{r} \cdot \mathbf{T}^t = [0.454, 0.071, 0.475]$

$t = 3$: $\quad \mathbf{r} \cdot \mathbf{T}^t = [0.454, 0.243, 0.303]$

$t = 5$: $\quad \mathbf{r} \cdot \mathbf{T}^t = [0.432, 0.181, 0.387]$

$t = 10$: $\quad \mathbf{r} \cdot \mathbf{T}^t = [0.389, 0.212, 0.399]$

The initialization of $\mathbf{r}$ can also be chosen uniformly distributed, or based on previously computed PageRanks.

Link Analysis

Algorithm: IterativePageRank

Input: $G = (P, L)$. Web graph with pages P and links L.
λ. Random jump probability.

Output: I. Approximate PageRanks for all pages in P.

```
# Initialization of I
I, R = vectors of length |P|
FOREACH i ∈ [0, |P|] DO
    I[i] = 1/|P|
ENDDO

# Update loop
WHILE NOT converged(I, R) DO
|
ENDDO

RETURN(I)
```


Link Analysis

Algorithm: IterativePageRank

Input: $G = (P, L)$. Web graph with pages P and links L.
λ. Random jump probability.

Output: I. Approximate PageRanks for all pages in P.

```
# Update loop
WHILE NOT converged(I, R) DO
  # Reinitialization of R
  FOREACH i ∈ [0, |P|] DO
    R[i] = 1/|P|
  ENDDO

  # Update step
  FOREACH p ∈ P DO
  |
  ENDDO

  I = R
ENDDO
```

Link Analysis

Algorithm: IterativePageRank

Input: $G = (P, L)$. Web graph with pages P and links L.
λ. Random jump probability.

Output: I. Approximate PageRanks for all pages in P.

Update step
FOREACH $p \in P$ **DO**
    $Q = \{\ q \mid q \in P$ and $(p, q) \in L\ \}$
    IF $|Q| > 0$ **THEN**
        FOREACH $q \in Q$ **DO**
            $R[q] = R[q] + (1 - \lambda) \cdot I[p]/|Q|$
        ENDDO
    ELSE
        FOREACH $p \in P$ **DO**
            $R[p] = R[p] + (1 - \lambda) \cdot I[p]/|P|$
        ENDDO
    ENDIF
ENDDO

PageRank: Convergence

Convergence is typically checked with

$$||R - I|| < \tau,$$

where $||\cdot||$ denotes the L_1 or L_2 norm, and τ is a threshold.

The choice of τ depends on the number of documents n, since $||R - I||$ (for a fixed numerical precision) increases with n. The larger τ, the faster convergence is reached. Optionally, $||R - I||/n$ can be used instead.

The number of iterations required to converge is roughly in $\mathcal{O}(\log n)$. [Page 1999]

Counterintuitively, the PageRank algorithm does not converge faster when initialized with the PageRanks from a previously converged run compared to a uniform initialization. This is partly due to the rapid pace at which the web evolves.

[Meyer 2004]

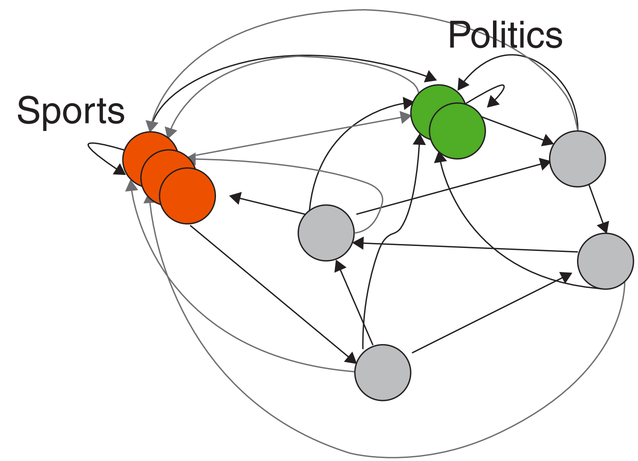
Link Analysis
PageRank: Variants

The PageRank algorithm can be applied to web graphs at different levels of granularity:

- Web pages

- **Websites**
  Combining all pages hosted under a domain allows for computing the importance of websites as a whole.

- **Topic-specific clusters**
  Categorizing web page by topic, or clustering them induces a web graph between categories / clusters. This allows for computing PageRanks within and across categories / clusters.



- **Personalized PageRank**
  Based on topic-specific PageRanks, a user may provide personal interests which can be applied as normalized weights onto each topic's PageRank vector.

