Web Document Models for Web Information Retrieval

Michel Beigbeder
Web page model

URL

HTTP header

TITLE 1

Section

<P>text

anchor text

anchor text

HTML:

(Internal) Structure

Links (External structure)
Flat Web page model

No structure
Links for gathering only
Vector Web page model (1)

No structure
No links
No text: bag of words
Vector Web page model (2)

No structure
No links
No text: bag of words

Improvements:
Tag dependant weight (vector) or occurrence (boolean)
No structure
No links
No text: bag of words

Improvements:
Tag dependant weight (vector) or occurrence (boolean)

Indexation with anchor text of inlinks
Vector Web page model (4)

- **PageRank**: 0.8
- **Hubbiness**: 0.2
- **Authority**: 0.7

No structure
No links
No text: bag of words

Improvements:
Tag dependant weight (vector) or occurrence (boolean)

Indexation with anchor text of inlinks

PageRank (Google) & HITS
Vector Web page model (5)

No structure
No links
No text: bag of words

Improvements:
Tag dependant weigth (vector) or occurrence (boolean)

Indexation with anchor text of inlinks

PageRank (Google) & HITS

Occurrence proximity (Google)
What features to use?

<table>
<thead>
<tr>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP header</td>
</tr>
<tr>
<td>n-grams</td>
</tr>
<tr>
<td>stems</td>
</tr>
<tr>
<td>words</td>
</tr>
<tr>
<td>text</td>
</tr>
<tr>
<td>internal structure</td>
</tr>
<tr>
<td>external inlinks anchor text</td>
</tr>
</tbody>
</table>
Web graph usage

Index enhancement and relevance propagation
Page ranking

Topic page gathering
Page categorization
Page classification
Similar page discovery
Replica discovery
Logical Units Discovery
Communities discovery
Web graph usage

Index enhancement and relevance propagation
Page ranking

Topic page gathering
Page categorization
Page classification
Similar page discovery
Replica discovery
Logical Units Discovery
Communities discovery

→ clustering
Integrate clusterization in search engines

Relevance, popularity, etc. produce ranked lists thus the user as an $O(n)$ access
⇒
Organize results in trees (clusters) to have an $O(\log n)$ access.

How:

• clustering
  – before or after the query
  – in a site, in the Web
  – based on different similarities according to the user needs

• categorizing
  Using (open) categorizations (www.dmoz.org)