Introduction to Information Retrieval

MYE003-ΠΛΕ70: Ανάκτηση Πληροφορίας
Διδάσκουσα: Ευαγγελία Πιτουρά
Διάλεξη 10: Βασικές Θέματα Αναζήτησης στον Παγκόσμιο Ιστό.
Ανάλυση Συνδέσμων (link analysis)

- Ανάλυση συνδέσμων
  - PageRank
  - HITS (Κομβικές σελίδες και σελίδες κύρους)
PageRank

Ποιοι είναι οι σημαντικοί κόμβοι σε ένα γράφο;

- Degree centrality \(\text{degree}(v)/|E|\)
- Υποθέστε ότι ο \(X\) και ο \(Y\) έχουν 3 φίλους, αλλά οι φίλοι του \(X\) είναι ο Barak Obama, Larry Page, the Pope
  - Είναι το ίδιο σημαντικό;
PageRank

_Eigenvector centrality_

While (not converged)
  for each vertex v
    for each incoming edge from node u
      rank(v) = rank(u)

Аллά:
▪ το ίδιο σημαντικό μια σελίδα να έχει link από μια σελίδα με εκατομμύρια outgoing links και από μια σελίδα με μόνο λίγα outgoing links?
PageRank

*Eigenvector centrality*

While (not converged)

for each vertex v

for each incoming edge from node u

rank(v) = + rank(u)/outdegree(u)
Παράδειγμα

\[
\begin{align*}
  w_1 &= \frac{1}{3} w_4 + \frac{1}{2} w_5 \\
  w_2 &= \frac{1}{2} w_1 + w_3 + \frac{1}{3} w_4 \\
  w_3 &= \frac{1}{2} w_1 + \frac{1}{3} w_4 \\
  w_4 &= \frac{1}{2} w_5 \\
  w_5 &= w_2
\end{align*}
\]
Παράδειγμα

\[ w_1 = \frac{1}{3} w_4 + \frac{1}{2} w_5 \]
\[ w_2 = \frac{1}{2} w_1 + w_3 + \frac{1}{3} w_4 \]
\[ w_3 = \frac{1}{2} w_1 + \frac{1}{3} w_4 \]
\[ w_4 = \frac{1}{2} w_5 \]
\[ w_5 = w_2 \]
PageRank: Διανυσματική αναπαράσταση

Stochastic Adjacency Matrix – Πίνακας Γειτνίασης $M$

Πίνακας $M$ – πίνακας γειτνίασης του web

Αν $j \rightarrow i$, τότε $M_{ij} = 1/\text{outdegree}(j)$

Αλλιώς, $M_{ij} = 0$

Η πιθανότητα να πάμε στη σελίδα $i$ αν είμαστε στη σελίδα $j$

Έστω ότι η σελίδα $j$ έχει links σε 3 σελίδες, συμπεριλαμβανομένη της $i$, αλλά όχι της $x$. 

![Stochastic Adjacency Matrix Diagram]
PageRank: Διανυσματική αναπαράσταση

Page Rank Vector $r$

Ένα διάνυσμα με μία τιμή για κάθε σελίδα (το PageRank της σελίδας)

$r = M \cdot r$

- Principal eigenvector του $M$
- Προσομοιώνει ένα τυχαίο περίπατο (random walks)
Random walk

- Question: what is the probability $p_{i}^{t}$ of being at node $i$ after $t$ steps?

\[
\begin{align*}
p_{1}^{0} &= \frac{1}{5} \\
p_{2}^{0} &= \frac{1}{5} \\
p_{3}^{0} &= \frac{1}{5} \\
p_{4}^{0} &= \frac{1}{5} \\
p_{5}^{0} &= \frac{1}{5}
\end{align*}
\]

\[
\begin{align*}
p_{1}^{t} &= \frac{1}{3} p_{4}^{t-1} + \frac{1}{2} p_{5}^{t-1} \\
p_{2}^{t} &= \frac{1}{2} p_{1}^{t-1} + p_{3}^{t-1} + \frac{1}{3} p_{4}^{t-1} \\
p_{3}^{t} &= \frac{1}{2} p_{1}^{t-1} + \frac{1}{3} p_{4}^{t-1} \\
p_{4}^{t} &= \frac{1}{2} p_{5}^{t-1} \\
p_{5}^{t} &= p_{2}^{t-1}
\end{align*}
\]
PageRank with restart

Δύο προβλήματα

1. Dead ends: σελίδες χωρίς εξερχόμενες ακμές
   Έχουν ως αποτέλεσμα να ξεφεύγει (leak out) to PageRank

2. Spider traps: Ομάδα σελίδων που όλες οι εξερχόμενες ακμές είναι μεταξύ τους
   Τελικά απορροφούν όλο to PageRank
Dead end (αδιέξοδα)

\[ y = \begin{array}{cccc}
1/3 & 1/3 & 3/12 & 5/24 & 8/48 & 0 \\
1/3 & 1/6 & 1/6 & 3/24 & 5/48 & \ldots & 0 \\
1/3 & 1/6 & 1/12 & 1/12 & 3/48 & 0
\end{array} \]
### Spider trap

<table>
<thead>
<tr>
<th>y</th>
<th>a</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>m</td>
<td>0</td>
<td>1/2</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
\text{y} & 1/3 & 1/3 & 3/12 & 5/24 & 8/48 & 0 \\
\text{a} & 1/3 & 1/6 & 1/6 & 3/24 & 5/48 & 0 \\
\text{m} & 1/3 & 1/2 & 8/12 & 9/12 & 39/48 & 1 \\
\end{array}
\]

...
PageRank with restart

Dumping factor:
Random jump (teleport) to any node in the graph

Add a random jump to any node in the network

(reduce the effect of distant nodes in the PageRank)
Επεκτάσεις

Topic specific PageRank
Personalized PageRank
Η ιντερνετική πλατφόρμα HITS είναι μια βασική τεχνική που χρησιμοποιείται για την αξιολόγηση των σελίδων του ιντερνετ. Η τεχνική αυτή χρησιμοποιεί δύο βαθμούς:

- Ένα βαθμό κύρους (authority rank)
- Ένα κομβικό βαθμό (hub rank)
HITS

- **Authorities**: pages containing useful information (the prominent, highly endorsed answers to the queries)
  - Newspaper home pages
  - Course home pages
  - Home pages of auto manufacturers

- **Hubs**: pages that link to authorities (highly value lists)
  - List of newspapers
  - Course bulletin
  - List of US auto manufacturers

✓ A good hub links to many good authorities
✓ A good authority is linked from many good hubs
HITS: Algorithm

Each page $p_i$ has two scores:
- A **hub score** $(h)$ quality as an expert
  - Total sum of authority scores that it points to
- An **authority score** $(a)$ quality as content
  - Total sum of hub scores that point to it

\[
h_i = \sum_{j \neq i} a_j
\]

\[
a_i = \sum_{j \rightarrow i} h_j
\]
Iterative update

- Repeat the following updates, for all $x$:

**I operation**

$$h(x) \leftarrow \sum_{y \leftarrow x} a(y)$$

**O operation**

$$a(x) \leftarrow \sum_{y \rightarrow x} h(y)$$

Normalize (scale down)
Example

- hubs
- authorities
Example

Initialize

hubs  authorities
Example

Step 1: O operation

1  2  3  2  1
hubs  authorities
Example

Step 1: I operation

1 hubs → 6 authorities
2 hubs → 5 authorities
3 hubs → 5 authorities
2 authorities → 2 authorities
1 authorities → 1 authorities
Example

Step 1: Normalization (Max norm)

1/3  2/3  1  2/3  1/3
hubs

1
5/6
5/6
2/6
1/6

authorities
Example

Step 2: O step

hubs    authorities

1    1
11/6    5/6
16/6    5/6
7/6    2/6
1/6    1/6
Example

Step 2: 1 step

1 \rightarrow 33/6
11/6 \rightarrow 27/6
16/6 \rightarrow 23/6
7/6 \rightarrow 7/6
1/6 \rightarrow 1/6

hubs \quad \text{authorities}
Example

Step 2: Normalization

hubs  authorities

6/16  1
11/16  27/33
1  23/33
7/16  7/33
1/16  1/33
Example

Convergence

hubs | authorities
--- | ---
0.4 | 1
0.75 | 0.8
1 | 0.6
0.3 | 0.14
0 | 0
Πίνακας γειτνίασης

- \( n \times n \) adjacency matrix \( A \):
  - each of the \( n \) pages in the base set has a row and column in the matrix.
  - Entry \( A_{ij} = 1 \) if page \( i \) links to page \( j \), else = 0.

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0 & 1 & 0 \\
2 & 1 & 1 & 1 \\
3 & 1 & 0 & 0 \\
\end{array}
\]
Hub/authority vectors

- View the hub scores \( h() \) and the authority scores \( a() \) as vectors with \( n \) components.
- Recall the iterative updates

\[
h(x) \leftarrow \sum_{y \rightarrow x} a(y)
\]

\[
a(x) \leftarrow \sum_{x \rightarrow y} h(y)
\]
Rewrite in matrix form

- \( h = Aa \).
- \( a = A^t h \).

Recall \( A^t \) is the transpose of \( A \).

Substituting, \( h = AA^t h \) and \( a = A^t Aa \).

Thus, \( h \) is an eigenvector of \( AA^t \) and \( a \) is an eigenvector of \( A^t A \).

Further, our algorithm is a particular, known algorithm for computing eigenvectors: the *power iteration* method.

Guaranteed to converge.
Query dependent link analysis

- Given text query (say *browser*), use a text index to get all pages containing *browser*.
  - Call this the **root set** of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set.
- Call this the **base set**.
Query dependent input

Root set obtained from a text-only search engine

Root Set
Query dependent input

IN → Root Set → OUT
Query dependent input
Query dependent input

Base Set

IN

Root Set

OUT
Things to note

- Pulled together good pages regardless of *language of page content*.
- Use *only* link analysis *after* base set assembled
  - iterative scoring is query-independent.
- Iterative computation *after* text index retrieval - significant overhead.
Τι άλλο θα δούμε σήμερα;

- Τι ψάχνουν οι χρήστες
- Spam
- Πόσο μεγάλος είναι ο Ιστός;
ΟΙ ΧΡΗΣΤΕΣ
Ανάγκες Χρηστών

- Ποιοι είναι οι χρήστες;
- Μέσος αριθμός λέξεων ανά αναζήτηση 2-3
- Σπάνια χρησιμοποιούν τελεστές
Ανάγκες Χρηστών

Need [Brod02, RL04]

- **Informational** (πληροφοριακά ερωτήματα) – θέλουν να μάθουν (learn) για κάτι (~40% / 65%)
  - Συνήθως, όχι μια μοναδική ιστοσελίδα, συνδυασμός πληροφορίας από πολλές ιστοσελίδες

- **Navigational** (ερωτήματα πλοήγησης) – θέλουν να πάνε (go) σε μια συγκεκριμένη ιστοσελίδα (~25% / 15%)
  - Μια μοναδική ιστοσελίδα, το καλύτερο μέτρο = ακρίβεια στο 1 (δεν ενδιαφέρονται γενικά για ιστοσελίδες που περιέχουν τους όρους United Airlines)
Ανάγκες Χρηστών

**Transactional** (ερωτήματα συναλλαγής) – θέλουν να κάνουν *(do)* κάτι (σχετιζόμενο με το web) (~35% / 20%)

- Προσπελάσουν μια υπηρεσία *(Access a service)*
- Να κατεβάσουν ένα αρχείο *(Downloads)*
- Να αγοράσουν κάτι
- Να κάνουν κράτηση

*Γκρι περιοχές* *(Gray areas)*

- Find a good hub
- Exploratory search “see what’s there”
Examples of Typing Queries

Calculation: 5+4
Unit conversion: 1 kg in pounds
Currency conversion: 1 euro in kronor
Tracking number: 8167 2278 6764
Flight info: LH 454
Area code: 650
Map: columbus oh
Stock price: msft
Albums/movies etc: coldplay
Τι ψάχνουν;

Δημοφιλή ερωτήματα

- http://www.google.com/trends/hottrends

Και ανά χώρα

Τα ερωτήματα ακολουθούν επίσης power law κατανομή
Ανάγκες Χρηστών

Επηρεάζει (ανάμεσα σε όλα)

- την καταλληλότητα του ερωτήματος για την παρουσίαση διαφημίσεων

- τον αλγόριθμο/αξιολόγηση, για παράδειγμα για ερωτήματα πλοήγησης ένα αποτέλεσμα ίσως αρκεί, για τα όλα (και κυρίως πληροφοριακά) ενδιαφερόμαστε για την περιεκτικότητα/ανάκληση
Πόσα αποτελέσματα βλέπουν οι χρήστες

“When you perform a search on a search engine and don’t find what you are looking for, at what point do you typically either revise your search, or move on to another search engine? (Select one)"

- 25% After reviewing the first 2 pages
- 27% After reviewing more than 3 pages
- 20% After reviewing the first page
- 16% After reviewing the first few entries
- 12% Before leaving the first page

(Source: iprospect.com WhitePaper_2006_SearchEngineUserBehavior.pdf)
Πως μπορούμε να καταλάβουμε τις προθέσεις (intent) του χρήστη;

Guess user intent *independent of context*:
- Spell correction
- Precomputed “typing” of queries

Better: Guess user intent *based on context*:
- Geographic context (slide after next)
- Context of user in this session (e.g., previous query)
- Context provided by personal profile (Yahoo/MSN do this, Google claims it doesn’t)
Geographical Context

Three relevant locations
1. Server (nytimes.com → New York)
2. Web page (nytimes.com article about Albania)
3. User (located in Palo Alto)

Locating the user
- IP address
- Information provided by user (e.g., in user profile)
- Mobile phone

**Geo-tagging**: Parse text and identify the coordinates of the geographic entities
Example: East Palo Alto CA → Latitude: 37.47 N, Longitude: 122.14 W
✓ Important NLP problem
Geographical Context

How to use context to modify query results:

- **Result restriction**: Don’t consider inappropriate results
  - For user on google.fr only show .fr results
- **Ranking** modulation: use a rough generic ranking, rerank based on personal context

Contextualization / personalization is an area of search with a lot of potential for improvement.
Relevance and validity of results
- Precision at 1? Precision above the fold?
- Comprehensiveness – must be able to deal with obscure queries
  - Recall matters when the number of matches is very small

UI (User Interface) – Simple, no clutter, error tolerant
- No annoyances: pop-ups, etc.
- Trust – Results are objective
- Coverage of topics for polysemic queries
  - Diversity, duplicate elimination
Αξιολόγηση από τους χρήστες

- Pre/Post process tools provided
  - Mitigate user errors (auto spell check, search assist,...)
  - Explicit: Search within results, more like this, refine ...
  - Anticipative: related searches

- Deal with idiosyncrasies
  - Web specific vocabulary
    - Impact on stemming, spell-check, etc.
  - Web addresses typed in the search box
Navigational
Typo: Ioanina

Transactional query: adds
SPAM
(SEARCH ENGINE OPTIMIZATION)
The trouble with paid search ads

- It costs money. What’s the alternative?

*Search Engine Optimization (SEO):*
  - “Tuning” your web page to rank highly in the algorithmic search results for select keywords
  - Alternative to paying for placement
  - Thus, intrinsically a marketing function

- Performed by companies, webmasters and consultants (“Search engine optimizers”) for their clients
- Some perfectly legitimate, some very shady
Η απλούστερη μορφή

- Οι μηχανές πρώτης γενιάς βασίζονταν πολύ στο \( \text{tf/idf} \)
  - Οι πρώτες στην κατάταξη ιστοσελίδας για το ερώτημα \textit{maui resort} ήταν αυτές που περιείχαν τα περισσότερα \textit{maui} και \textit{resort}
- SEOs απάντησαν με πυκνή επανάληψη των επιλεγμένων όρων
  - π.χ., \textit{maui resort maui resort maui resort}
  - Συχνά, οι επαναλήψεις στο ίδιο χρώμα με \text{background} της ιστοσελίδα
    - Οι επαναλαμβανόμενοι όροι έμπαιναν στο ευρετήριο από crawlers
    - Αλλά δεν ήταν ορατοί από τους ανθρώπους στους browsers

Απλή πυκνότητα όρων δεν είναι αξιόπιστο ΑΠ σήμα
Παραλλαγές «keyword stuffing»

a web page loaded with keywords in the meta tags or in content of a web page (outdated)

- Παραπλανητικά meta-tags, υπερβολική επανάληψη
- Hidden text with colors, position text behind the image, style sheet tricks, etc.

**Meta-Tags =**
“... London hotels, hotel, holiday inn, hilton, discount, booking, reservation, sex, mp3, britney spears, viagra, ...”
Cloaking (Απόκρυψη)

- Παρέχει διαφορετικό περιεχόμενο ανάλογα αν είναι ο μηχανισμός σταχυολόγησης (search engine spider) ή ο browser κάποιου χρήστη
- DNS cloaking: Switch IP address. Impersonate

![Diagram showing cloaking process]
Άλλες τεχνικές παραπλάνησης (spam)

- **Doorway pages**
  - Pages optimized for a single keyword that re-direct to the real target page.
  - If a visitor clicks through to a typical doorway page from a search engine results page, redirected with a fast *Meta refresh* command to another page.

- **Lander page**:
  - optimized for a single keyword or a misspelled domain name, designed to attract surfers who will then click on ads.
Άλλες τεχνικές παραπλάνησης (spam)

- **Link spamming**
  - Mutual admiration societies, hidden links, awards
  - *Domain flooding*: numerous domains that point or redirect to a target page
  - Pay somebody to put your link on their highly ranked page
  - Leave comments that include the link on blogs

- **Robots (bots)**
  - Fake query stream – rank checking programs
    - “Curve-fit” ranking programs of search engines
  - Millions of submissions via Add-Url
The war against spam

- Quality signals - Prefer authoritative pages based on:
  - Votes from authors (linkage signals)
  - Votes from users (usage signals)
- Policing of URL submissions
  - Anti robot test
- Limits on meta-keywords
- Robust link analysis
  - Ignore statistically implausible linkage (or text)
  - Use link analysis to detect spammers (guilt by association)

- Spam recognition by machine learning
  - Training set based on known spam
- Family friendly filters
  - Linguistic analysis, general classification techniques, etc.
  - For images: flesh tone detectors, source text analysis, etc.
- Editorial intervention
  - Blacklists
  - Top queries audited
  - Complaints addressed
  - Suspect pattern detection
More on spam

- Web search engines have policies on SEO practices they tolerate/block

- Adversarial IR (Ανταγωνιστική ανάκτηση πληροφορίας): the unending (technical) battle between SEO’s and web search engines

Check out: Webmaster Tools (Google)
SIZE OF THE WEB
Ποιο είναι το μέγεθος του web?

Θέματα

- Στην πραγματικότητα, ο web είναι άπειρος
  - Dynamic content, e.g., calendars
  - Soft 404: [www.yahoo.com/<anything>](http://www.yahoo.com/) is a valid page
- Static web contains syntactic duplication, mostly due to mirroring (~30%)
- Some servers are seldom connected

Ποιο νοιάζει;

- Media, and consequently the user
- Σχεδιαστές μηχανών
- Την πολιτική crawl - αντίκτυπο στην ανάκληση.
Τι μπορούμε να μετρήσουμε;

Το σχετικό μέγεθος των μηχανών αναζήτησης

- The notion of a page being indexed is still *reasonably* well defined.
- Already there are problems
  - Document extension: e.g., engines index pages not yet crawled, by indexing anchortext.
  - Document restriction: All engines restrict what is indexed (first $n$ words, only relevant words, etc.)
  - Multi-tier indexes (access only top-levels)
New definition?

- The statically indexable web is whatever search engines index.
  - IQ is whatever the IQ tests measure.

- Different engines have different preferences
  - max url depth, max count/host, anti-spam rules, priority rules, etc.

- Different engines index different things under the same URL:
  - frames, meta-keywords, document restrictions, document extensions, ...
Relative Size from Overlap

Given two engines A and B

1. **Sample** URLs randomly from A
2. **Check** if contained in B and vice versa

\[
A \cap B = \frac{1}{2} \times \text{Size A}
\]

\[
A \cap B = \frac{1}{6} \times \text{Size B}
\]

\[
\frac{1}{2} \times \text{Size A} = \frac{1}{6} \times \text{Size B}
\]

\[
\therefore \frac{\text{Size A}}{\text{Size B}} = \frac{1}{3}
\]

**Each test involves:** (i) **Sampling** (ii) **Checking**
Δειγματοληψία (Sampling) URLs

Ideal strategy: Generate a *random URL*

- Problem: Random URLs are hard to find (and sampling distribution should reflect “user interest”)
- Approach 1: Random walks / IP addresses
  - In theory: might give us a true estimate of the size of the web (as opposed to just relative sizes of indexes)
- Approach 2: Generate a random URL contained in a given engine
  - Suffices for accurate estimation of relative size
Statistical methods

Approach 2
1. Random queries
2. Random searches

Approach 1
1. Random IP addresses
2. Random walks
Random URLs from random queries

1. Generate random query: how?
   - Lexicon: 400,000+ words from a web crawl
   - Conjunctive Queries: $w_1$ and $w_2$
     - e.g., vocalists AND rsi

2. Get 100 result URLs from engine A

3. Choose a random URL as the candidate to check for presence in engine B
   - This distribution induces a probability weight $W(p)$ for each page.
Query Based Checking

- Either *search for the URL* if the engine B support this or

- *Generate a Strong Query* to check whether an engine B has a document D:
  - Download D. Get list of words.
  - Use 8 low frequency words as AND query to B
  - Check if D is present in result set.
Random searches

- Choose random searches extracted from a local query log [Lawrence & Giles 97] or build “random searches” [Notess]
- Use only queries with small result sets.
- For each random query: compute ratio $size(r_1)/size(r_2)$ of the two result sets
- Average over random searches
Random searches

- 575 & 1050 queries from the NEC RI employee logs
- 6 Engines in 1998, 11 in 1999
- Implementation:
  - Restricted to queries with < 600 results in total
  - Counted URLs from each engine after verifying query match
  - Computed size ratio & overlap for individual queries
  - Estimated index size ratio & overlap by averaging over all queries
Queries from Lawrence and Giles study

- adaptive access control
- neighborhood preservation topographic
- hamiltonian structures
- right linear grammar
- pulse width modulation neural
- unbalanced prior probabilities
- ranked assignment method
- internet explorer favourites importing
- karvel thornber
- zili liu

- softmax activation function
- bose multidimensional system theory
- gamma mlp
- dvi2pdf
- john oliensis
- rieke spikes exploring neural
- video watermarking
- counterpropagation network
- fat shattering dimension
- abelson amorphous computing
Random IP addresses

- Generate random IP addresses
- Find a web server at the given address
  - If there’s one
- Collect all pages from server
  - From this, choose a page at random
Random IP addresses

- HTTP requests to random IP addresses
  - Ignored: empty or authorization required or excluded
  - [Lawr99] Estimated 2.8 million IP addresses running crawlable web servers (16 million total) from observing 2500 servers.
  - OCLC using IP sampling found 8.7 M hosts in 2001
    - Netcraft [Netc02] accessed 37.2 million hosts in July 2002
- [Lawr99] exhaustively crawled 2500 servers and extrapolated
  - Estimated size of the web to be 800 million pages
  - Estimated use of metadata descriptors:
    - Meta tags (keywords, description) in 34% of home pages, Dublin core metadata in 0.3%
Τυχαίοι Περίπατοι (Random walks)

Το διαδίκτυο ως ένας κατευθυνόμενος

- Ένας τυχαίος περίπατος σε αυτό το γράφο
  - Includes various “jump” rules back to visited sites
    - Does not get stuck in spider traps!
    - Can follow all links!
  - Συγκλίνει σε μια κατανομή σταθερής κατάστασης (stationary distribution)
    - Must assume graph is finite and independent of the walk.
    - Conditions are not satisfied (cookie crumbs, flooding)
    - Time to convergence not really known
  - Sample from stationary distribution of walk
  - Use the “strong query” method to check coverage by SE
Size of the web

Check out
http://www.worldwidewebsize.com/

The Indexed Web contains at least 3.57 billion pages (Tuesday, 20 May, 2014). The Indexed Web contains at least 4.58 billion pages (Thursday, 19 May, 2016).
Size of the web

Based on the **number of pages indexed** by search engines (Google, Bing, Yahoo, Ask) (minus their overlap)

*Size of the index of a search engine* based on a method that combines

- *word frequencies* obtained from a large offline text collection (corpus), and
- *search counts* returned by the engines.
Size of index

- Each day 50 words are sent to all four search engines.
- Record number of webpages found for these words
- Compare their relative frequencies in the background corpus
- Make multiple extrapolated estimations of the size of the engine's index which are subsequently averaged.

Example
Say word 'the' is present in 67,61% of all documents within the corpus
Google says that it found 'the' in 14.100.000.000 webpages
Estimated size of the Google's total index would be 23.633.010.000.

Background corpus contains more than 1 million webpages from DMOZ
50 words selected evenly across logarithmic frequency intervals (Zipf's Law)
Size of the web

- Overlap between the indices of two search engines is estimated by daily overlap counts of URLs returned in the top-10 by the engines.
- Words randomly drawn from the DMOZ background corpus.
Introduction to Information Retrieval

Size Bing
(Number of webpages)

Size Google
(Number of webpages)
Τι άλλο θα δούμε

Web crawlers or spiders (κεφ 20)
**Spiders (σταχυολόγηση ιστού)**

1. Page1
2. Page2
3. Page3

**Web**

**Spider**

**Query String**

**Document corpus**

**IR System**

**Ranked Documents**
Web Crawling (σταχυολόγηση ιστού)

Web crawler or spider

*How hard and why?*

- Getting the content of the documents is easier for many other IR systems.
  - E.g., indexing all files on your hard disk: just do a recursive descent on your file system
- For web IR, getting the content of the documents takes longer, because of latency.
  - But is that really a design/systems challenge?
Begin with known “seed” URLs
Fetch and parse them
  Extract URLs they point to
  Place the extracted URLs on a queue
Fetch each URL on the queue and repeat
URL frontier

- URLs crawled and parsed
- URL frontier: found, but not yet crawled
- Unseen URLs
Processing steps in crawling

- Pick a URL from the frontier
- Fetch the document at the URL
- Parse the URL
  - Extract links from it to other docs (URLs)
- Check if URL has content already seen
  - If not, add to indexes
- For each extracted URL
  - Ensure it passes certain URL filter tests
  - Check if it is already in the frontier (duplicate URL elimination)

E.g., only crawl .edu, obey robots.txt, etc.
Simple picture – complications

- Web crawling isn’t feasible with one machine
  - All of the above steps distributed
- Malicious pages
  - Spam pages
  - Spider traps – incl dynamically generated
- Even non-malicious pages pose challenges
  - Latency/bandwidth to remote servers vary
  - Webmasters’ stipulations
    - How “deep” should you crawl a site’s URL hierarchy?
  - Site mirrors and duplicate pages
- Politeness – don’t hit a server too often
Simple picture – complications

Magnitude of the problem

To fetch 20,000,000,000 pages in one month . . . we need to fetch almost 8000 pages per second!

- Actually: many more since many of the pages we attempt to crawl will be duplicates, unfetchable, spam etc.
Explicit and implicit politeness

- **Explicit politeness**: specifications from webmasters on what portions of site can be crawled
  - robots.txt

- **Implicit politeness**: even with no specification, avoid hitting any site too often
Robots.txt

- Protocol for giving spiders ("robots") limited access to a website, originally from 1994
  - [www.robotstxt.org/wc/norobots.html](http://www.robotstxt.org/wc/norobots.html)
- Website announces its request on what can(not) be crawled
  - For a server, create a file `/robots.txt`
  - This file specifies access restrictions
Robots.txt example

- No robot should visit any URL starting with "/yoursite/temp/", except the robot called "searchengine":

  User-agent: *
  Disallow: /yoursite/temp/

  User-agent: searchengine
  Disallow: 
Βασική αρχιτεκτονική του σταχυολογητή

**WWW**

**DNS**

**Fetch**

**Parse**

**Content seen?**

**Doc FP’s**

**robots filters**

**URL filter**

**Dup URL elim**

**URL Frontier**
DNS (Domain Name Server)

- A lookup service on the internet
  - Given a URL, retrieve its IP address
  - Service provided by a distributed set of servers – thus, lookup latencies can be high (even seconds)

- Common OS implementations of DNS lookup are blocking: only one outstanding request at a time

- Solutions
  - DNS caching
  - Batch DNS resolver – collects requests and sends them out together
**Parsing: URL normalization**

- When a fetched document is parsed, some of the extracted links are *relative* URLs.
- During parsing, must normalize (expand) such relative URLs.
Content seen?

- Duplication is widespread on the web
- If the page just fetched is already in the index, do not further process it
- This is verified using document fingerprints or shingles
Distributing the crawler

- Run multiple crawl threads, under different processes – potentially at different nodes
  - Geographically distributed nodes
Distributing the crawler
Distributing the crawler

- Partition hosts being crawled into nodes
  - Hash used for partition
- How do these nodes communicate and share URLs?
Communication between nodes

- Output of the URL filter at each node is sent to the Dup URL Eliminator of the appropriate node
URL frontier: two main considerations

- **Politeness**: do not hit a web server too frequently
- **Freshness**: crawl some pages more often than others
  - E.g., pages (such as News sites) whose content changes often

These goals may conflict each other.
(E.g., simple priority queue fails – many links out of a page go to its own site, creating a burst of accesses to that site.)
ΤΕΛΟΣ 10ου Μαθήματος

Ερωτήσεις?

Χρησιμοποιήθηκε κάποιο υλικό από:
✓ Pandu Nayak and Prabhakar Raghavan, CS276:Information Retrieval and Web Search (Stanford)
✓ Hinrich Schütze and Christina Lioma, Stuttgart IIR class