Exercise 6.1

- An adjacency matrix:

```
0 1 1 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 1
0 0 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

- PageRanks for $\lambda = 0.1$, $\lambda = 0.5$, and $\lambda = 1$:

  - $\lambda = 0.1$:
    
    0.039 0.042 0.053 0.061 0.072 0.088 0.113 0.167 0.316
  
  - $\lambda = 0.5$:
    
    0.061 0.064 0.073 0.079 0.087 0.098 0.114 0.143 0.214
  
  - $\lambda = 1$:
    
    0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1

Exercise 6.2

a) Differences and similarities between Boolean retrieval and coordination level matching?

b) Definition of TF–IDF? Rationale?

c) The Probabilistic Ranking Principle?
Exercise 6.2

d) Recall and fallout are hard to compute. Why?
   Workaround!

e) How does LSI work? How can it help in IR tasks?

f) Cluster hypothesis? Correct?

g) The key idea behind language models?

h) The main advantage and disadvantage of pseudo relevance feedback?

i) Support vector machines? How do they work?
   Kernel trick?
j) The three most important differences between IR and Web search?

k) The major components of a Web search engine?

l) How can we estimate the size of the Web?

m) What is shingling used for in Web search? Why do we need this complicated randomized approximation algorithm instead of just computing Jaccard coefficients?

Spamdexing

- Spamdexing = The practice of modifying the Web to get certain Web resources unjustifiably ranked high on search engine result lists
- Often a synonym of SEO ("search engine optimization")
Spamdexing usually means finding weaknesses in ranking algorithms and exploiting them.

Usually, it looks like this:

- There are two classes of spamdexing techniques:
  - **Content spam**: Alter a page’s contents
  - **Link spam**: Alter the link structure between pages

Spammer

Search Engine

Finds a new loophole

Fills the loophole

Idea:

- Exploit TF–IDF

Method:

- Repeatedly place the keywords to be found in the text, title, or URI of your page
- Place the keywords in anchor texts of pages linking to your page
- Merge your content by high-quality content taken from (possibly a lot of) other pages

Countermeasures:

- Train classification algorithms to detect patterns that are “typical” for spam pages
- Most difficult part: Find suitable features to describe pages

- Degree of similarity to other pages, degree of term repetitions, …

Example (Google bombing):

Keywords are placed in anchor texts of pages linking to your page

Very hard to detect if many unrelated people do it…

• There is a further way to detect content spam:
  - After a spammer has cheated the search engine, the same must be done for real users
  - Therefore, spammers try to hide the parts of their page used for spamdexing:
    - Place text behind images
    - Write text in the background color
    - Set the font size to 0
    - Dynamically delete text using scripts
    - Deliver different Web pages to Web crawlers (“cloaking”)
    - …

- Most of these techniques can be detected by search engines

But: This kind of analysis is quite expensive…

Doorway pages:

This is the specific technique that recently caused Google to ban ricoh.de and bmw.de!
**Link Spam**

**Idea:**
- Improve your page’s rank by getting in-links from other pages

**Method (comment spamming):**
- Collect a list of high-quality sites that allow other users to post their own comments
  - Comments in blogs
  - Public forums
  - Wikis
- Write (a lot of) comments linking to your page
  - This can easily be automated since most people use standard software for running their forums, blogs, ...

**Countermeasures:**
- Require users to solve CAPTCHAs

**Link Spam (2)**

**CAPTCHAs:**

- CAPTCHA: “Completely Automated Public Turing test to tell Computers and Humans Apart”
- Character recognition is easy for humans, but hard for machines

**Countermeasures (taken by spammers):**
- Build character recognition algorithms that are hand-tailored to the CAPTCHAs generated by standard CAPTCHA software
- Let real humans solve CAPTCHAs (e.g. pay 1 cent per solution)

**Link Spam (3)**

**Method (link farms):**
- Create a large group of pages that link to each other
- Or: Participate in link exchange programs
- Try to create link patterns that look “normal”
- Set out-links to topically related high-quality pages, which gives you high hub scores
  - This can be done e.g. by cloning directories like DMOZ
- This will consequently lead to high authority scores for your other pages

**Link Spam (4)**

**Method (honeypots):**
- Create a set of pages (called honeypot) that provide some useful resource
  - Examples: Copies of Unix documentation pages or Wikipedia pages
- Insert hidden links to some target pages to be boosted
- This honeypot then attracts people to link to it, boosting indirectly the ranking of the target pages

**Link Spam (5)**

**Method (buy expired domains):**
- Monitor DNS records for domains that will expire soon, i.e. whose registration has not been extended on time
- Buy such domains when they expire
- Replace their pages by pages with links to your own pages
- Using this technique you can get hold of all external links linking to the expired domain’s pages

**Link Spam (6)**

**Countermeasures:**
- In general, link spam is quite hard to detect
  - Heuristic:
    - Remove pages whose in-links look almost the same (can detect Google bombing and comment spamming)
  - Heuristic:
    - Remove (modified) copies of high-quality content (can detect honeypots)
  - Heuristic:
    - Create a white-list of pages known to be “good” and use the link-distance to these pages as an indicator of trustworthiness
As long as you don’t want to sell Viagra or memberships in online casinos:
Invest your time into creating good content!

Usually, the costs of cheating search engines are higher than the benefits
— Recall Google’s ban on bmw.de

Therefore:
— Create high-quality content
— Follow the rule “link = recommendation” when creating links
— Build crawler-friendly Web sites
— Use “white hat” techniques like Google AdWords

Google’s webmaster guidelines:
http://www.google.com/support/webmasters/bin/answer.py?answer=35769

…or how to build one of the most powerful
data centers out of crappy hardware
— For a long time, Google has jealously guarded the design of its data centers
  — In 2007 & 2009 some details have been revealed

The Google Servers
— Google uses only custom built servers
— Google is the world 4th largest server producer
  — They don’t even sell servers...
  — In 2007, it was estimated that Google operates over 1,000,000 servers, spanning 34 major and many more minor data centers

Data centers are connected to each other and major internet hubs via massive fiber lines
— ~7% of all internet traffic is generated by Google
— ~60% of that traffic connects directly to consumer networks without connecting to a global backbone
— If Google was an ISP, it would be the 3rd largest global carrier
Some Google datacenter facts and rumors

- Construction of four new data centers (in 2007):
  - 600 million dollars
- Annual operation costs in 2007:
  - 2.4 billion dollars
- Energy consumption per data center:
  - 50 megawatts
  - The largest center in Oregon: over 110 megawatts
  - The whole region of Braunschweig: 225 megawatts

The largest center in Oregon:

- Slightly outdated hardware
- 12V battery to counter unstable power supplies
- No cases, racks are setup in standard shipping containers and are just wired together
- More info: http://www.youtube.com/watch?v=Ho1GEyftpQ

Google servers are highly unstable but also very cheap

- Typical first year for a new cluster (several racks):
  - ~0.5 overheating
  - ~1 PDU (power distribution unit) failure
  - ~1 rack-move
  - ~1 network rewiring
  - ~20 rack failures
  - 40–80 machines instantly disappear, 1–6 hours to get back
  - ~5 racks go wonky
  - 40–80 machines see 50% packet loss
  - ~8 network maintenance operations
  - ~12 router reloads
  - ~500–1000 machines powered down with plenty of warning, ~6 hours
  - ~500–1000 machines suddenly disappear, ~6 hours
  - ~1 network rewiring
  - Rolling ~5% of machines down over 2-day span
  - ~12 network maintenance operations
  - ~300 machines powered down with plenty of warning, ~6 hours
  - ~1000 machines powered down with plenty of warning, ~6 hours
  - ~500–1000 machines suddenly disappear, ~6 hours
  - ~1200 machines powered down with plenty of warning, ~6 hours

Challenges to the data center software

- Deal with all these hardware failures
- Avoiding any data loss
- Guarantee ~100% global uptime
- Decrease maintenance costs to minimum
- Allow flexible extension of data centers
- Solutions
  - Use cloud technologies
  - GFS (Google File System) and Google Big Table Data System

More details:
Lecture “Distributed databases”

Lecture 13: Miscellaneous

1. Spamdexing
2. Hardware for Large Scale Web Search
3. Metasearch
4. Privacy Issues
Thus, whatever method we choose to solve our aggregation problem, it will have severe weaknesses.

Fortunately, in our case, the aggregate ranking will not be used for anything of importance, so violations are not that critical.

There are many different aggregation methods available, two of which we will discuss briefly:

- Majority rule
- The Borda count

Let’s assume that any page being ranked by at least one individual engine, is ranked by all of them.

In fact, this usually is not true.

But it is possible to extend the methods to handle this problem.

A metasearch engine can only access the result lists returned by the individual engines.

It is not able to exploit any engine’s internal information.

Therefore, we have to solve the following problem:

- Given: A set of $k$ individual ordered result lists of size $n$
- Task: Aggregate these $k$ rankings into a single ranking
- Of course, some constraints should hold here
- That define which properties a “good” aggregate should have

This is a well-known problem from social choice theory having a lot of different solutions.

### What’s a Good Aggregate?

- **Pareto efficiency:**
  If every individual engine ranks a certain page higher than another, then so must the aggregate ranking.

- **Non-dictatorship:**
  The aggregate ranking is not just always the same as a certain fixed individual engine’s ranking.

- **Independence of irrelevant alternatives:**
  If page $A$ is ranked higher than page $B$ in the aggregate ranking, then adding a new page $C$ to each of the input rankings will not change $A$’s relation to $B$.

- **Maybe some more…**

These three constraints sound completely reasonable.

Clearly, any “reasonable” aggregation algorithm should adhere to these constraints.

In fact, “should” means “cannot” in this case.

Kenneth Arrow’s impossibility theorem (1951):

“If there are more than two different pages to be ranked, then there is no deterministic aggregation algorithm that satisfies Pareto efficiency, non-dictatorship, and independence of irrelevant alternatives.”

### Majority Rule

- For any pair of pages $(a, b)$, count how many search engines rank $a$ higher than $b$.

- If the majority of engines ranks $a$ higher than $b$, then place $a$ before $b$ in the aggregate ranking.

- Ties also can be handled but let’s assume that the number of search engines is odd.

- Construct the aggregate ranking from this comparisons.

- **Example:**

<table>
<thead>
<tr>
<th>Engine 1</th>
<th>Engine 2</th>
<th>Engine 3</th>
<th>Aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
### Majority Rule (2)

- One important drawback of majority vote are **cycles**
- **Example:**

<table>
<thead>
<tr>
<th>Engine 1</th>
<th>Engine 2</th>
<th>Engine 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>

2 engines rate A > B
2 engines rate B > C
2 engines rate C > A

- There are **many methods** available to break cycles...

### The Borda Count (2)

- **Advantages** of the Borda count:
  - It is easy to compute
  - It allows for ties in the aggregate ranking
  - It allows for weight the individual engine's importance
  - Multiply the scores assigned by "good" engines by a factor higher than 1
  - Multiply the scores assigned by "bad" engines by a factor smaller than 1

- **Disadvantage:**
  - It assumes a uniform degradation of relevance in each ranking

### The Borda Count (3)

- **The Borda Count avoids cycles**
- Every engine assigns a numerical score to each page:
  - The best page gets a score of \( n \) (if there are \( n \) pages in total)
  - The second-best page gets a score of \( n - 1 \), ...
- The final ranking is created by adding all scores

<table>
<thead>
<tr>
<th>Engine 1 Score</th>
<th>Engine 2 Score</th>
<th>Engine 3 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3</td>
<td>A 3</td>
<td>A 2</td>
</tr>
<tr>
<td>B 2</td>
<td>C 2</td>
<td>B 1</td>
</tr>
<tr>
<td>C 1</td>
<td>I 1</td>
<td>C 1</td>
</tr>
</tbody>
</table>

Aggregate Score

- For each page, add up its individual scores:
- \( m \): The number of pages ranked by both engines
- \( p^+ \): The number of agreeing pairs of pages ranked by both engines
- \( p^- \): The number of disagreeing pairs of pages ranked by both engines

### Measures of Agreement

- Sometimes it is useful to measure the agreement between **two search engines**
  - Search engines that often yield very similar rankings should be considered as dependent
  - Therefore, they should get a lower influence at aggregation
- One of the most popular measures is **Kendall’s \( \tau \)**
  - For each pair of pages \((a, b)\) ranked by both engines, determine if both engines agree in their relative ranking or if one engine ranks \( a \) higher than \( b \) and the other ranks \( b \) higher than \( a \)
  - Basically, Kendall’s \( \tau \) is the ratio of agreeing pairs compared to all pairs ranked by both engines

### Kendall’s \( \tau \)

- **Define:**
  - \( m \): The number of pages ranked by both engines
  - \( p^+ \): The number of agreeing pairs of pages ranked by both engines
  - \( p^- \): The number of disagreeing pairs of pages ranked by both engines
- **Example:**

\[
\tau = \frac{p^+ - p^-}{\sqrt{(m - 1)(m - 2)}} = \frac{2 \cdot (p^+ - p^-)}{m \cdot (m - 1)}
\]

<table>
<thead>
<tr>
<th>Engine 1</th>
<th>Engine 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

Therefore, \( \tau = \frac{1}{3} \)
Today, metasearch is well-suited for answering very special queries with maximum recall. Unfortunately, it fails to increase result quality for most other queries. Why?

Metasearch works best if...
- The engines used are completely independent
- The engines used are all of similar (high) quality

The reality:
- Most search engines use similar methods, thus being dependent
- There are just a few good engines and many bad ones

On August 6, 2006, AOL tried to help IR researchers. They released very private data about its users (without their permission)
- 20 million Web queries from 650,000 AOL users
- All searches from those users for a three month period
- Whether they clicked on a result
- Where this result appeared on the result page

Of course, the data has been made anonymous by replacing each AOL username by a random ID number. Unfortunately, this did not work too well… Let’s look at some examples.

User 311045:
- how to change brake pads on scion xb
- 2005 us open cup florida state champions
- how to get revenge on a ex
- replacement bumper for scion xb
- florida department of law enforcement
- crime stoppers florida

User 11574916:
- cocaine in urine
- asian mail order brides
- states reciprocity with florida
- florida dui laws
- extradition from new york to florida
- mail order brides from largos
- will one be extradited for a dui
- cooking jobs in french quarter new orleans
- will i be extradited from ny to fl on a dui charge
• User 3540871:
  – I have an interview at Comcast and I need help
  – Cheap rims for a Ford Focus
  – How can I get a job in Joliet IL with a theft on my background
  – I need to trace a cellular location
  – I need to know if my spouse is cheating and I need to do a cellular trace for free
  – Jobs with no background checks
  – How can I get a job with a conviction
  – Motels in Joliet IL
  – Motels in Gurnee IL area for under 40 dollars
  – My baby’s father physically abuses me

• User 17556639:
  – How to kill your wife
  – Wife killer
  – How to kill a wife
  – Dead people
  – Pictures of dead people
  – Killed people
  – Murder photo
  – Steak and cheese
  – Decapitated photos
  – Car crashes
  – Car crash photo

• User 4417749:

• This has just been a very small sample…
  – AOL removed the data on August 7, 2006, from their Web site, one day after its release
  – They apologized: “This was a screw up”
  – However, the data is still out there…
    http://gregsadetsky.com/aol-data
    …and probably always will be

• Netflix, America’s largest online DVD rental service, had similar problems
  – They released data about what DVDs has been rented by each user, along with the users’ movie ratings
  – As with the AOL data set, user IDs have been replaced by random numbers
  – Unfortunately, researchers have been able to reconstruct the identity of some customers by comparing their movie ratings with reviews written at imdb.com, a public movie database

• The Netflix Data Set

• 123people.com
  – http://www.123people.com
Next Semester

- Knowledge-Based Systems and Deductive Databases
- Data Warehousing and Data Mining Techniques
- Spatial Databases and Geographic Information Systems
- Seminar “Best of Data Mining”