Information Retrieval (Part V)

[DAT640] Information Retrieval and Text Mining

Krisztian Balog
University of Stavanger

October 8, 2019

So far...

- Representing document content
  - Document-term matrix, term vector, TFIDF weighting
- Retrieval models
  - Vector space model, Language models, BM25
- Scoring queries
  - Inverted index, term-at-a-time/doc-at-a-time scoring
- Fielded document representations
  - Mixture of Language Models, BM25F
- Retrieval evaluation

Today

- Feedback (query expansion)
- Web search

Feedback

Feedback

- Take the results of a user's actions or previous search results to improve retrieval
- Often implemented as updates to a query, which then alters the list of documents
- Overall process is called **relevance feedback**, because we get feedback information about the relevance of documents
    - **Explicit feedback**: user provides relevance judgments on some documents
    - **Pseudo relevance feedback** (or *blind feedback*): we don't involve users but "blindly" assume that the top-k documents are relevant
    - **Implicit feedback**: infer relevance feedback from users' interactions with the search results (clickthroughs)

Feedback in an IR system

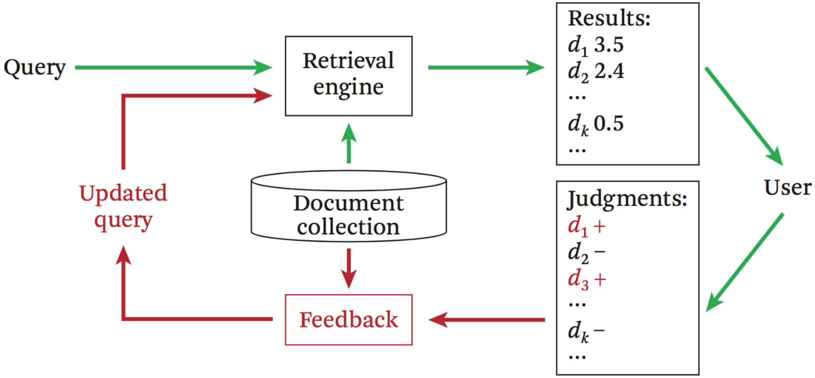


Figure: Illustration is taken from (Zhai&Massung, 2016)[Fig. 7.1]

Feedback in the Vector Space Model

- It is assumed that we have examples of relevant (D^+) and non-relevant (D^-) documents for a given query
- General idea: modify the query vector (adjust weight of existing terms and/or assign weight to new terms)
  - As a result, the query will usually have more terms, which is why this method is often called **query expansion**

Rocchio feedback

- Idea: adjust the weights in the query vector to move it closer to the cluster of relevant documents

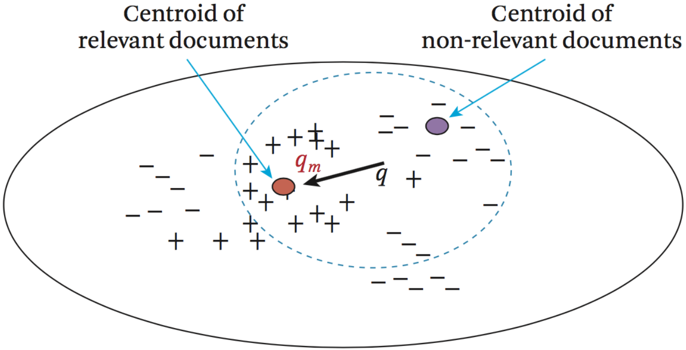


Figure: Illustration is taken from (Zhai&Massung, 2016)[Fig. 7.2]

Rocchio feedback

- Modified query vector:

$$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D^+|} \sum_{d \in D^+} \vec{d} - \frac{\gamma}{|D^-|} \sum_{d \in D^-} \vec{d}$$

  - $\vec{d}$: original query vector
  - D^+, D^-: set of relevant and non-relevant feedback documents
  - α, β, γ: parameters that control the movement of the original vector
- The second and third terms of the equation correspond to the centroid of relevant and non-relevant documents, respectively

Practical considerations

- Modifying all the weights in the query (and then using them all for scoring documents) is computationally heavy
  - Often, only terms with the highest weights are retained
- Non-relevant examples tend not to be very useful
  - Sometimes negative examples are not used at all, or γ is set to a small value

Exercise #1

- Implement Rocchio feedback
- Code skeleton on GitHub: `exercises/lecture_11/exercise_1.ipynb` (make a local copy)

Feedback in Language Models

- We generalize the query likelihood function to allow us to include feedback information more easily
- (Log) query likelihood

$$\log P(q|d) \propto \sum_{t \in q} f_{t,q} \times \log P(t|\theta_d)$$

- Generalize $f_{t,q}$ to a query model $P(t|\theta_q)$

$$\log P(q|d) \propto \sum_{t \in q} P(t|\theta_q) \times \log P(t|\theta_d)$$

  - Often referred to as **KL-divergence** retrieval, because it provides the same ranking as minimizing the Kullback-Leibler divergence between the query model θ_m and the document model θ_d
  - Using a maximum likelihood query model this is rank-equivalent to query likelihood scoring

Query models

- Maximum likelihood estimate (original query)

$$P_{ML}(t|\theta_q) = \frac{f_{t,q}}{|q|}$$

  - I.e., the relative frequency of the term in the query
- Linear interpolation with a feedback query model $\hat{\theta}_q$

$$P(t|\theta_q) = \alpha P_{ML}(t|\theta_q) + (1-\alpha)P(t|\hat{\theta}_q)$$

  - α has the same interpretation as in the Rocchio feedback model, i.e., how much we rely on the original query

Relevance models

- **Relevance models** are a theoretically sound and effective way of estimating feedback query models
- Main idea: consider other terms that co-occur with the original query terms in the set of feedback documents $\hat{D}$
  - Commonly taken to be the set of top-k documents (k=10 or 20) retrieved using the original query with query likelihood scoring
- Two variants with different independence assumptions
- Relevance model 1
  - Assume full independence between the original query terms and the expansion terms:

$$P_{RM1}(t|\hat{\theta}_q) \approx \sum_{d \in \hat{D}} P(d) P(t|\theta_d) \prod_{t' \in q} P(t'|\theta_d)$$

  - Often referred to as *RM3* when linearly combined with the original query

Relevance models

- Relevance model 2
  - The original query terms $t' \in q$ are still assumed to be independent of each other, but they are dependent on the expansion term t:

$$P_{RM2}(t|\hat{\theta}_q) \approx P(t) \prod_{t' \in q} \sum_{d \in \hat{D}} P(t'|\theta_d) P(d|t)$$

  - where $P(d|t)$ is computed as

$$P(d|t) = \frac{P(t|\theta_d)P(d)}{P(t)} = \frac{P(t|\theta_d)P(d)}{\sum_{d' \in \hat{D}} P(t|\theta_{d'})P(d')}$$

Illustration

t	$P_{ML}(t\|\theta_q)$	t	$P(t\|\theta_q)$
machine	0.5000	vision	0.2796
vision	0.5000	machine	0.2762
		image	0.0248
		vehicles	0.0224
		safe	0.0220
		cam	0.0214
		traffic	0.0178
		technology	0.0176
		camera	0.0173
		object	0.0147

Table: Baseline (left) and expanded (right) query models for the query *machine vision*; only the top 10 terms are shown.

Feedback summary

- Overall goal is to get a richer representation of the user's underlying information need by enriching/refining the initial query
- Interpolation with the original query is important
- Relevance feedback is computationally expensive! Number of feedback terms and expansion terms are typically limited (10..50) for efficiency considerations
- Queries may be hurt by relevance feedback ("query drift")

Web Search

Web search

- Before the Web: search was small scale, usually focused on libraries
- Web search is a major application that everyone cares about
- Challenges
  - Scalability (users as well as content)
  - Ensure high-quality results (fighting SPAM)
  - Dynamic nature (constantly changing content)

Some specific techniques

- Crawling
  - Freshness
  - Focused crawling
  - Deep Web crawling
- Indexing
  - Distributed indexing
- **Retrieval** $\Leftarrow$
  - Link analysis

Deep (or hidden) Web

- Much larger than the “conventional” Web
- Three broad categories:
  - Private sites
    - No incoming links, or may require log in with a valid account
  - Form results
    - Sites that can be reached only after entering some data into a form
  - Scripted pages
    - Pages that use JavaScript, Flash, or another client-side language to generate links

Discussion

Question

How to make content on the Deep Web searchable (indexable)?

Surfacing the Deep Web

- Pre-compute all interesting form submissions for each HTML form
- Each form submission corresponds to a distinct URL
- Add URLs for each form submission into search engine index

Link analysis

- Links are a key component of the Web
- Important for navigation, but also for search

- Both anchor text and links are used by search engines

Anchor text

- Aggregated from all incoming links and added as a separate document field
- Tends to be short, descriptive, and similar to query text
    - Can be thought of a description of the page "written by others"
- Has a significant impact on effectiveness for *some types of queries*

Example

"winter school"

page1

```
I'll be presenting our work at a
<a href="pageX">winter school</a>
in Bressanone, Italy.
```

"information retrieval"

page2

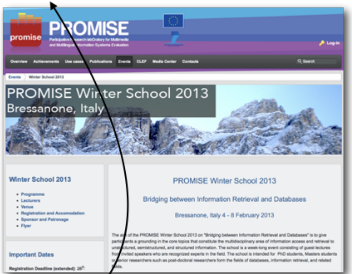
```
List of winter schools in 2013:
<ul>
<li><a href="pageX">information
retrieval</a></li>
...
</ul>
```

page3

```
The PROMISE Winter School in will
feature a range of <a
href="pageX">IR lectures</a> by
experts from the field.
```

"IR lectures"

pageX



Fielded document representation

title	`Winter School 2013`
meta	`PROMISE, school, PhD, IR, DB, [...]` `PROMISE Winter School 2013, [...]`

headings	`PROMISE Winter School 2013`
	`Bridging between Information Retrieval and Databases`
	`Bressanone, Italy 4-8 February 2013`

body	`The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured,semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as postdoctoral researchers form the fields of databases, information retrieval, and related fields. [...]`

anchors	`winter school`
	`information retrieval`
	`IR lectures`

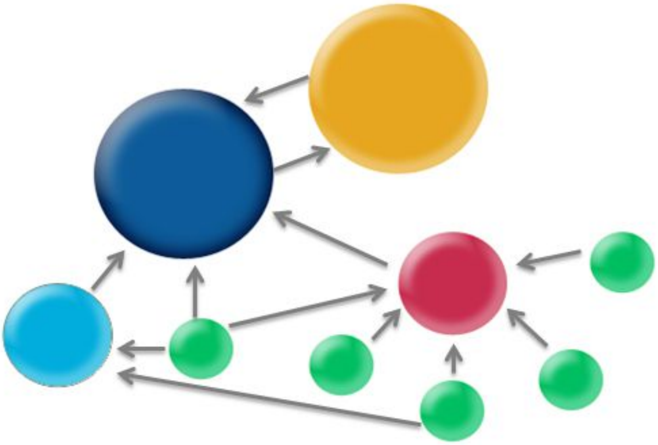
Document importance on the Web

- What are web pages that are popular and useful to *many* people?
- Use the links between web pages as a way to measure popularity
- The most obvious measure is to count the number of *inlinks*
    - Quite effective, but very susceptible to SPAM

PageRank

- Algorithm to rank web pages by popularity
- Proposed by Google founders Sergey Brin and Larry Page in 1998
- Main idea: **A web page is important if it is pointed to by other important web pages**
- PageRank is a numeric value that represents the importance of a web page
    - When one page links to another page, it is effectively casting a vote for the other page
    - More votes implies more importance
    - Importance of each vote is taken into account when a page's PageRank is calculated

Illustration



Source: `https://www.shoutmeloud.com/how-to-calculate-pagerank-google-seo.html`

Random Surfer Model

- PageRank simulates a user navigating on the Web randomly as follows
- The user is currently at page a
  - She moves to one of the pages linked from a with probability $1 - q$
  - She jumps to a random web page with probability q
    - This is to ensure that the user doesn't "get stuck" on any given page (i.e., on a page with no outlinks)
- Repeat the process for the page she moved to
- The PageRank score of a page is the average probability of the random surfer visiting that page

PageRank formula

Jump to a random page
with this probability
(q is typically set to 0.15)

Follow one of the hyperlinks in the
current page with this probability

PageRank value of page p_i

$$PR(a) = \frac{q}{T} + \overbrace{(1-q)}\sum_{i=1}^{n}\frac{PR(p_i)}{L(p_i)}$$

PageRank of page a

Number of outgoing links of
page p_i

Total number of pages in
the Web graph

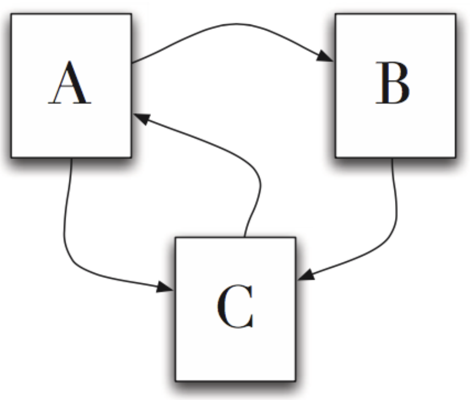
page a is pointed by
pages $p_1 \ldots p_n$

Technical issues

- This is a recursive formula. PageRank values need to be computed iteratively
    - We don't know the PageRank values at start. We can assume equal values $(1/T)$
- Number of iterations?
    - Good approximation already after a small number of iterations; stop when change in absolute values is below a given threshold

Example #1

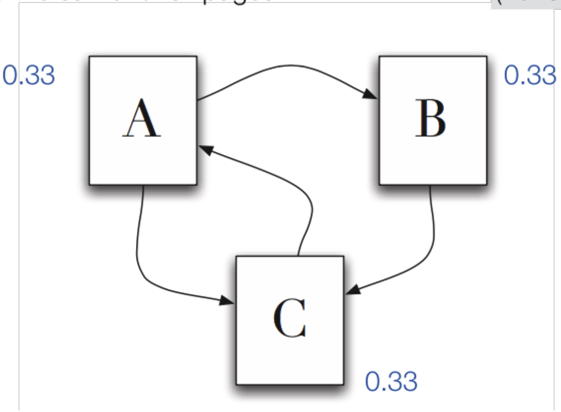
q=0
(no random jumps)



Example #1

Iteration 0: assume that the PageRank values are the same for all pages

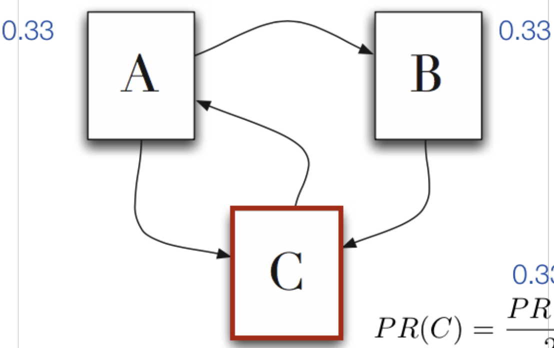
q=0
(no random jumps)



Example #1

Iteration 1

q=0
(no random jumps)



$$PR(C) = \frac{\overset{0.33}{PR(A)}}{2} + \frac{\overset{0.33}{PR(B)}}{1}$$

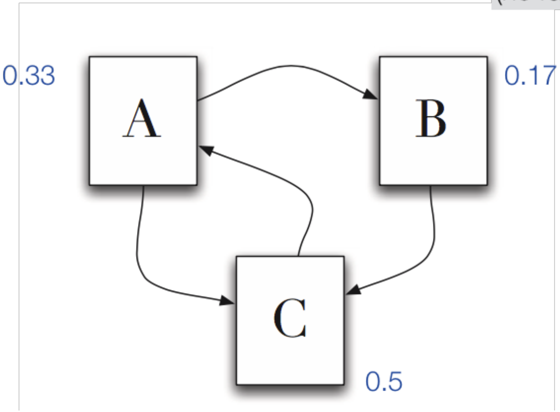
=0.5

PageRank of C depends on the PageRank values of A and B

Example #1

at the end of **Iteration 1**

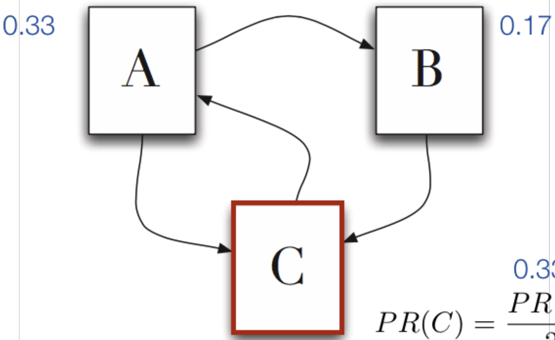
q=0
(no random jumps)



Example #1

Iteration 2

q=0
(no random jumps)



$$PR(C) = \frac{\overset{0.33}{PR(A)}}{2} + \frac{\overset{0.17}{PR(B)}}{1}$$

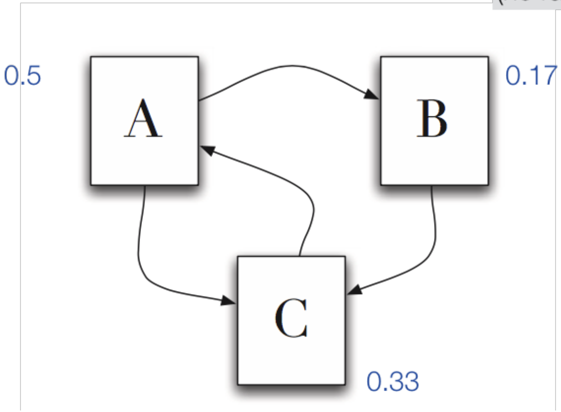
=0.33

PageRank of C depends on the PageRank values of A and B

Example #1

at the end of **Iteration 2**

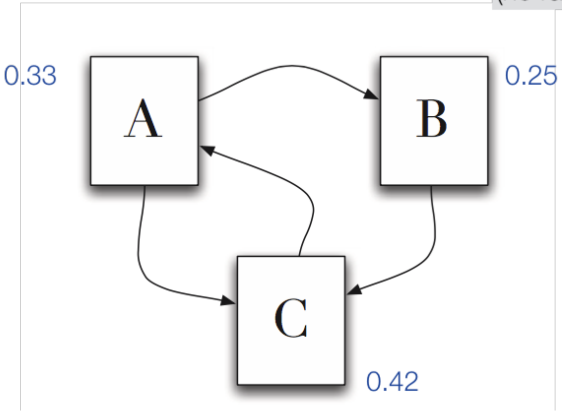
q=0
(no random jumps)



Example #1

at the end of **Iteration 3**

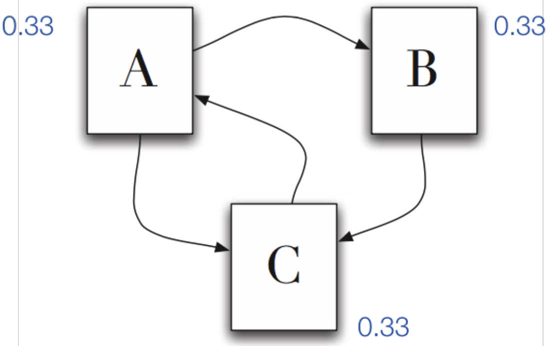
q=0
(no random jumps)



Example #2

Iteration 0: assume that the PageRank values are the same for all pages

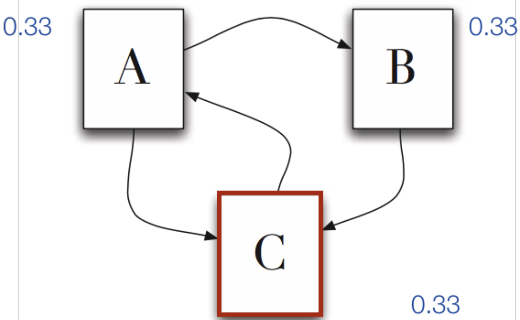
q=0.2
(with random jumps)



Example #2

Iteration 1

q=0.2
(with random jumps)



$$PR(C) = \frac{0.2}{3} + 0.8\left(\frac{\overset{0.33}{PR(A)}}{2} + \frac{\overset{0.33}{PR(B)}}{1}\right) \mathbf{=0.47}$$

Exercise #2

- PageRank computation (paper-based)

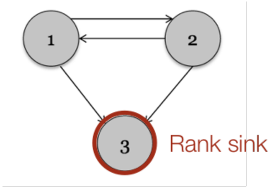
Discussion

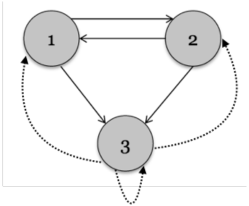
Question

How are PageRank scores affected by pages that do not have any outgoing links?

Dealing with “rank sinks”

- How to handle *rank sinks* (“dead ends”), i.e., pages that have no outlinks?
- Assume that it links to all other pages in the collection (including itself) when computing PageRank scores





Exercise #3

- PageRank computation (paper-based)

Online PageRank checkers

Google

pagerank checker

All | Images | News | Maps | Videos | More | Settings | Tools

About 3,450,000 results (0.47 seconds)

https://checkpagerank.net
Check Page Rank - Check Your PageRank Free!
Page Rank **Checker** and Domain Analysis Tool. Free tool reports **PageRank** and other important SEO statistics about your website. Fake Page Rank Detection!
What is PageRank? · PageRank is BACK!!! · PageRank Blog · Business Directory

https://www.prchecker.info › check_page_rank
Google PageRank Checker - Check Google page rank instantly
Page Rank **Checker** is a completely free service to **check** Google **pagerank** instantly using our online page rank **check** tool or a small **pagerank** button.

https://www.prchecker.info
Google PageRank Checker - Check Google page rank of any ...
Page Rank **Checker** is a completely Free tool to **check** Google PR, page rank of your web site easily and possibly display your Google **PageRank** on your web ...

https://dnschecker.org › pagerank
Page Rank Checker - Check Your Website Pagerank
With **Pagerank Checker**, you can check the page rank of your website. Just enter domain and check what is the current pagerank of your website.

PageRank summary

- Important example of query-independent document ranking
  - Web pages with high PageRank are preferred
- It is, however, not as important as conventional wisdom holds
  - Just one of the many features a modern web search engine uses
  - It tends to have the most impact on popular queries

Incorporating document importance (e.g., PageRank)

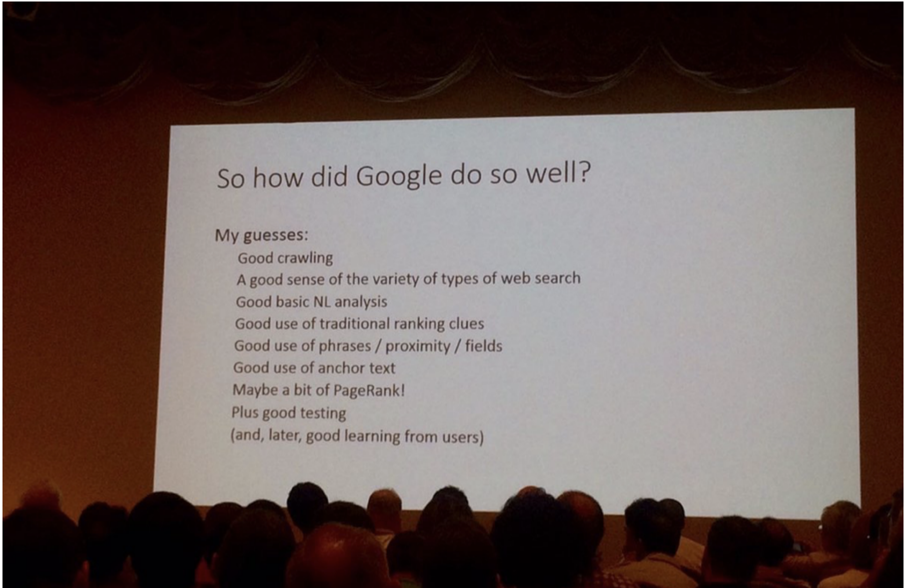
- How to incorporate document importance into the ranking?
- As a query-independent ("static") score component

$$score'(d, q) = score(d, q) \times score(d)$$

- In case of Language Models, document importance is encoded as the document prior $P(d)$

$$P(d|q) \propto P(q|d)P(d)$$

Stephen Robertson, SIGIR'17 keynote



Discussion

Question

What is search engine optimization (SEO)?

Search Engine Optimization (SEO)

- A process aimed at making the site appear high on the list of (organic) results returned by a search engine
- Considers how search engines work
  - Major search engines provide information and guidelines to help with site optimization
    - Google/Bing Webmaster Tools
  - Common protocols
    - Sitemaps (`https://www.sitemaps.org`)
    - robots.txt

White hat vs. black hat SEO

- White hat
  - Conforms to the search engines' guidelines and involves no deception
  - "Creating content for users, not for search engines"
- Black hat
  - Disapproved of by search engines, often involve deception
    - Hidden text
    - Cloaking: returning a different page, depending on whether it is requested by a human visitor or a robot

Some SEO techniques

- Editing website content and HTML source
- Increase relevance to specific keywords
- Increasing the number of incoming links (“backlinks”)
- Focus on long tail queries
- Social media presence

Reading

- Text Data Management and Analysis (Zhai&Massung)
  - Chapter 7
  - Section 10.3