

Web Search

Class 1: Course presentation

João Magalhães

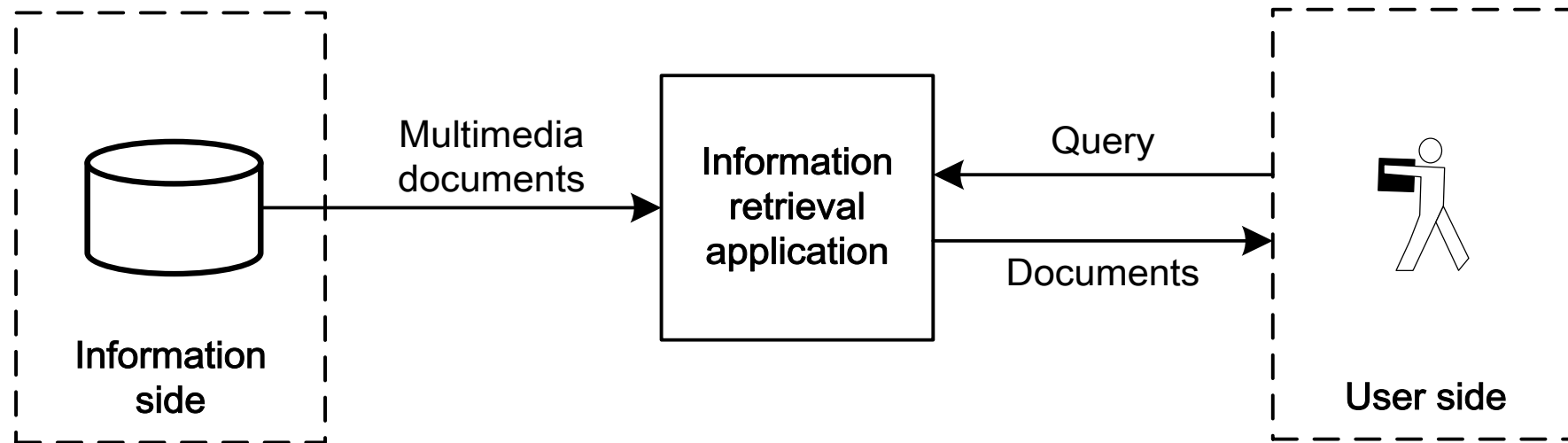
A bit of history – 20th Century

- 1945 Vannevar Bush's **Memex (Memory Extender)**
- 1965 Ted Nelson's **Hypermedia**
- 1975 Keith v. Rijsbergen's "**Information Retrieval**" book
- 1989 Tim Berners-Lee's **first HTTP and HTML** implementations
- 1992 Lynx Web Browser
- 1993 NCSA Mosaic Browser, **first TREC**
- 1994 **Netscape, Yahoo!, WebCrawler** https://en.wikipedia.org/wiki/Web_search_engine
- 1995 Altavista search engine, Apache, PHP
- 1996 Wayback machine
- 1997 PageRank
- 1998 Google <http://www.google.com/intl/en/about/company/history/>

A bit of history – 21st Century

- 2001 Wikipedia
- 2003 MySpace, Hi5, Skype
- 2004 Facebook, Flickr
- 2005 YouTube, Reddit, Mechanical Turk
- 2007 Twitter
- 2010 Instagram, Kaggle
- 2011 SnapChat
- 2012 Vine
- ...

Relevance vs similarity



What is the best [search space + dissimilarity function] to compute the relevance of documents for a given user information need?

What makes a good search application?

- **Efficiency:** application replies to user queries without noticeable delays.
 - 1 sec is the “limit for users feeling that they are freely navigating the command space without having to unduly wait for the computer”
 - Miller, R. B. (1968). Response time in man-computer conversational transactions. *Proc. AFIPS Fall Joint Computer Conference* Vol. 33, 267-277.
- **Effectiveness:** application replies to user queries with relevant answers.
 - This depends on the interpretation of the user query and the stored information.

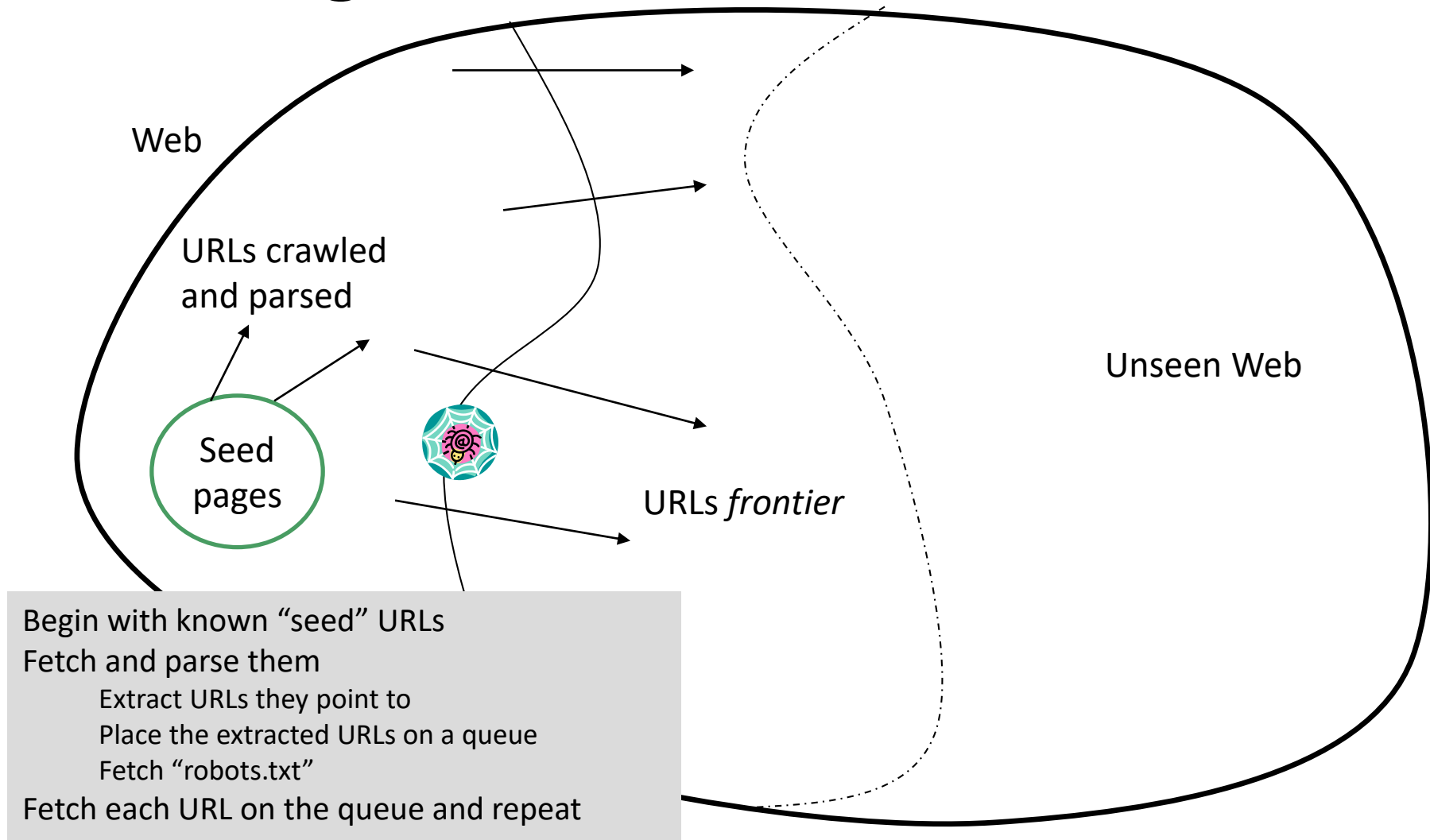
The tasks of a search application

- **Collect** data for storage
 - Crawler
- Analyse collected data and compute the **relevant information**
 - Information analysis
- Store data in an **efficient** manner
 - Indexing
- Process **user** information needs
 - Querying
- Find the documents that best **match** the user information need
 - Ranking

Crawler

- Discovers data for storage and indexing
- Applied when data has to be “discovered”
 - i.e., only a sample of the full data can be collected.
- If the sample is unbiased, it is a faithful representation of the real scenario.
- Implements a strategy to collect relevant data
 - e.g. on the Web the crawler needs to decide which links are more fruitful to follow and if a page should be indexed or not (it can be spam or phishing)

Web crawling



Information analysis

- This stage deals with the extraction of the information to be made searchable
- Extract meaningful words, pairs of words or n-grams
- Extract images and their main characteristics
- Link visual characteristics and text data

This patient had a sudden loss of her motor functions (she wasn't able to move her right arms and legs) 2 months before the study. She went through a slow recovery with a lot of physical therapy and drugs. She was recovering some of her movements but suddenly all the improvement stopped. We performed an MRI that showed the changes expected for a lesion of that time (2 months old) but also showed an increase in the size of the ventricular system (where the Cerebrospinal fluid or CSF flows) that was causing hydrocephalus. Due to this finding, the patient went through another surgery and had a shunt valve installed; the last word we had from one of her relatives is that she is again on recovery.



The *official* report included this: T 1 coronal SE (spin echo) sequence that shows an area of infarction in the left parietal lobe. Also enlargement of the ventricular system is observed.

Indexing

- This stage creates an index to quickly locate relevant documents
- An index is an aggregation of several data structures (e.g. several B-trees)
- Index compression is used to reduce the amount of space and the time needed to compute similarities
- The distribution of the index pages across a cluster improves the search engine responsiveness

Querying

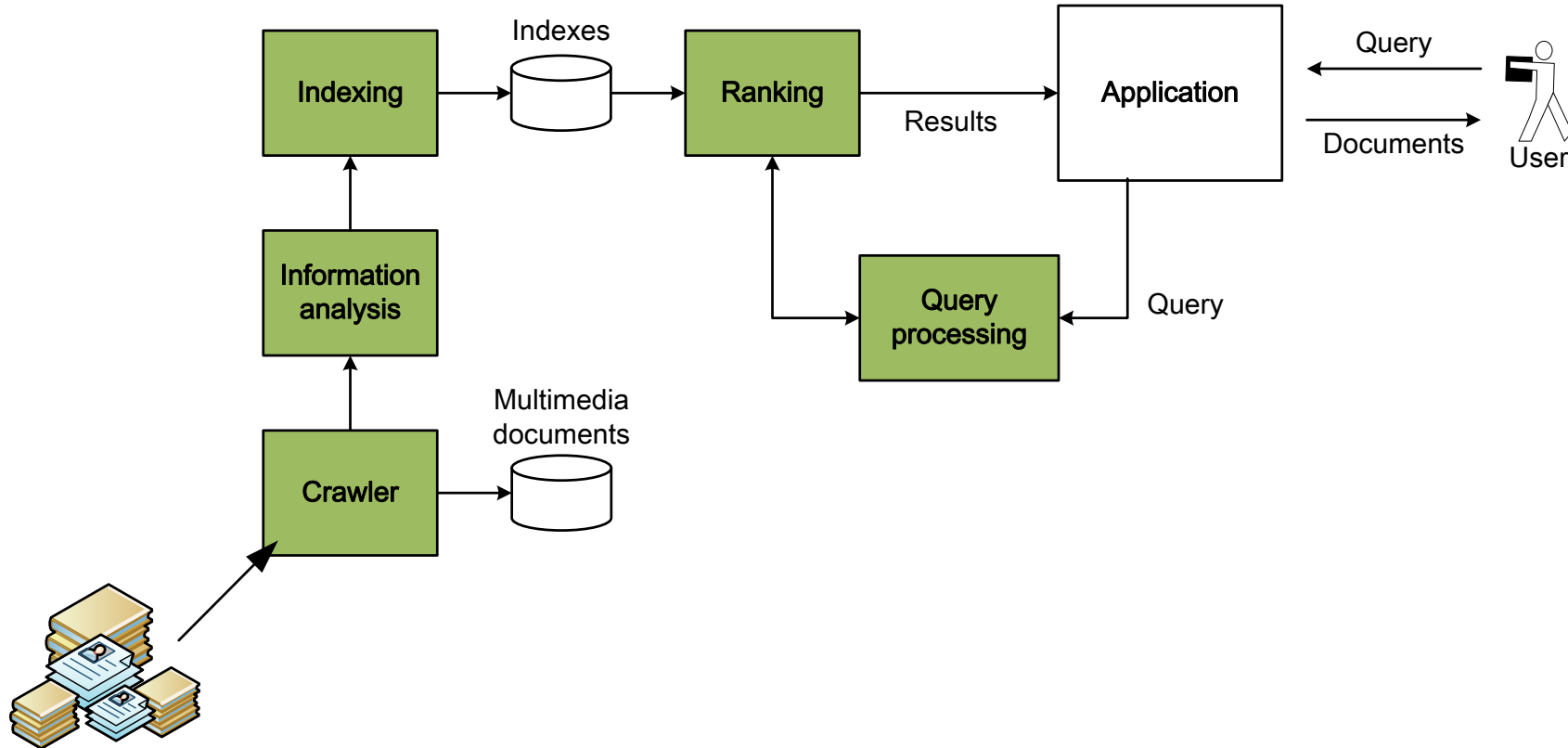
- Conversion of the user query into the internal search space
 - Parsing
- Usage history
 - Cookies, profiles, etc.
- User intention
 - What type of task is the user doing?

Ranking

- Once the user query is converted into the internal search space...
 - The ranking function sorts the information according to its relevance to the user query
- Ranking functions should model the human notion of relevance
 - We don't really know the mathematical form of the human notion of similarity... it is highly subjective and dynamic. 😊

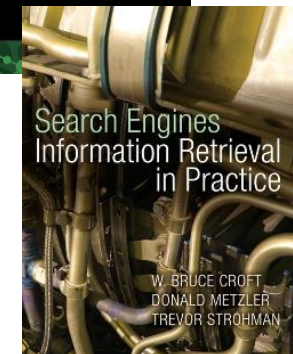
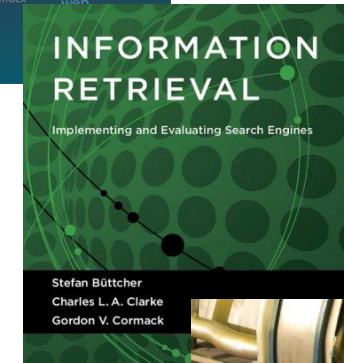
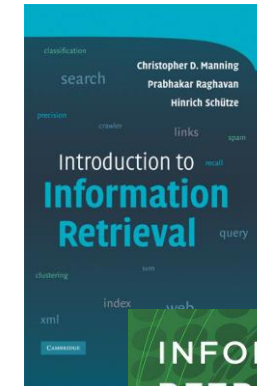
**Since all models are wrong,
one can only hope for useful approximations.**

Putting all together...



References

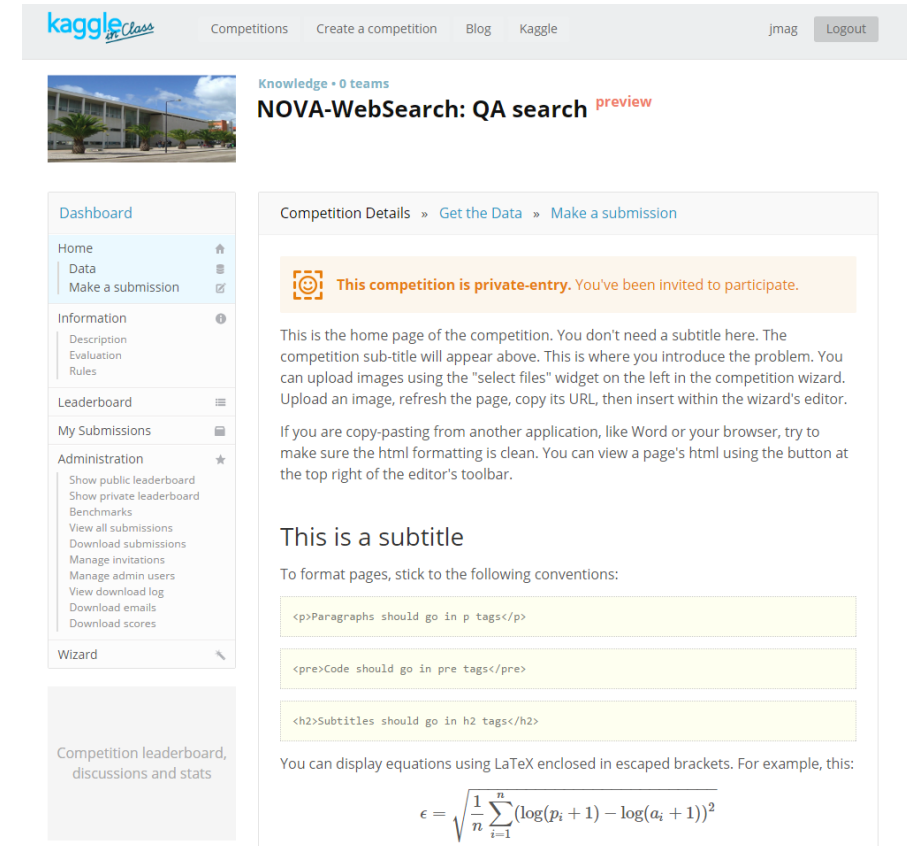
- Slides and articles provided during classes.
- Books:
 - C. D. Manning, P. Raghavan and H. Schütze, “**Introduction to Information Retrieval**”, Cambridge University Press, 2008.
 - Stefan Buettcher, Charles L. A. Clarke, Gordon V. Cormack, “**Information Retrieval: Implementing and Evaluating Search Engines**”, The MIT Press, 2010.
 - W. Bruce Croft, Donald Metzler, and Trevor Strohman, “***Search engines: Information retrieval in practice***,” Addison-Wesley, 2010.



Course grading

- 30% theoretical part (test or exam)
- Labs (groups of 3 students):
 - 20% Guided project (3 weekly checkpoints)
 - 50% Independent project
- Important dates:

• Guided project	25 April
• Independent project	4 June
• Test	9 June
• Exam	23 June



The screenshot shows the Kaggle website interface. At the top, there's a navigation bar with 'kaggle inClass' logo, 'Competitions', 'Create a competition', 'Blog', 'Kaggle', 'jmag', and 'Logout'. Below the navigation bar, there's a header section with 'Knowledge • 0 teams' and 'NOVA-WebSearch: QA search preview'. The main content area is divided into two columns. The left column contains a sidebar with links: 'Dashboard', 'Home', 'Data', 'Make a submission', 'Information', 'Leaderboard', 'My Submissions', 'Administration', and 'Wizard'. The right column shows the competition details, including a warning that the competition is private-entry, instructions on how to format the problem description, and a section for the competition subtitle. The subtitle section includes a warning about private-entry and instructions on how to format the subtitle. Below the subtitle section, there's a section for the competition description, which includes a warning about private-entry and instructions on how to format the description. The description section also includes a warning about private-entry and instructions on how to format the description.

Groups must use Kaggle to register by March 17.

Projects will be run in Kaggle,
for the sake of students' instant satisfaction.

(URL will be provided next week)

Guided project: StackOverflow answer search

- Task:
 - Implement a search engine to search for StackOverflow CrossValidated Answers.
 - Understand the roles of each component of a search engine in the performance of the search results.
 - If you prefer you can use Python Q&A (it is a much larger dataset, so you should have a very good computer to run experiments).
- Guided steps:
 - Week 1 (0%): baseline provided in the first class.
 - Week 2 (25%): Fields and ranking: indexing fields, cosine distance, parsing, etc.
 - Week 3 (25%): Analysers: Tokenization, stop words, stemming, n-grams, skip-grams, part of speech.
 - Week 4 (25%): PageRank for users.
- Report (25%):
 - Evaluation: metrics, graphs, etc.
 - 4 pages report (no cover page, no annexes: just meat!). ACM template (word or latex).
 - Weekly submission of the guided steps (at end of each week).

Indep. project: Information stream summaries

- Task: for a given query, provide a temporal summary of information
- Scoring
 - Implementation originality: 25%
 - Code “cleanness”: 25%
 - Critical discussion: 25%
 - Report: 25%
- Report organization:
 - Introduction
 - Algorithms from classes used in the project
 - Implementation:
 - What are your ideas? What makes your project unique?
 - Evaluation
 - Dataset description
 - Baselines
 - Results
 - Critical discussion
 - References
- Format:
 - 8 pages maximum (no-cover, references and annexes don't count to the page limit).
 - ACM template (word or latex)

Course plan

Date	#	Lectures	#	Labs
10-Mar-16	1	Introduction	1	Guided Project: StackOverflow Answers Search (20%) Libraries, luke, Lucene demo, first kaggle submission
17-Mar-16	2	Ranking text data	2	Fields and ranking: indexing fields, cosine distance, parsing, etc.
24-Mar-16	3	Ranking linked data	3	Analysers: tokenization, stop words, stemming, n-grams, skip-grams
31-Mar-16	4	Evaluation protocols	4	PageRank for users
07-Apr-16	5	Query processing	5	Evaluation: metrics, graphs, etc.
14-Apr-16	Easter holidays			
21-Apr-16	6	Index construction	6	Project submission (25 -Apr-16)
28-Apr-16	7	Efficient query processing	1	Independent Project: Summarization of information streams (50%) Edinburgh Music Festival (1 month) + test queries + training queries
05-May-16	8	Probabilistic retrieval models	2	Crowdsourcing, results pooling and inter-annotator agreement
12-May-16	9	Language models	3	Instagram: Ranking images (ImageNet + Visual composition)
19-May-16	10	Rank fusion + LETOR (CA)	4	Ranking for diversity and novelty detection
26-May-16	11	Recommendation + LSI	5	Project support
02-Jun-16	12	Revisions	6	Project submission
09-Jun-16	13	Teste (30%)	7	-

Summary

- “Web Search” course context
- Course objectives and plan
- Grading
- Labs