PRI Presentation, 22/23 Edition

PRI · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Today's Plan

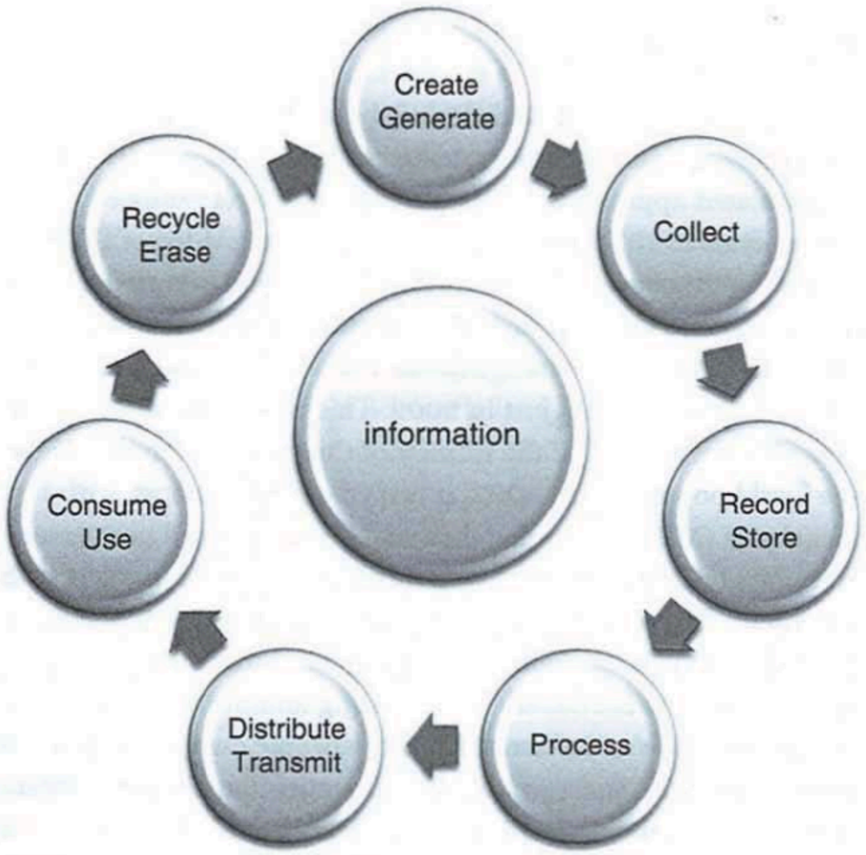
- Information processing and retrieval: context and motivation
- Course presentation: topics, materials, classes, evaluation
- Course projects: groups, themes, rules
- Q&A

Context and Motivation

Information Society

- Information communication technologies are ubiquitous in modern societies.
- An ever-growing number of activities depend on the ability to extract value from information.
- Human progress and welfare is largely dependent on an efficient management of the life cycle of information.
- New professional profiles: data engineer, data architect, data analyst, data scientist.
- This course is an introduction to information processing and to information retrieval.

Information Life Cycle



1. A typical information life cycle

Image from Information - A Very Short Introduction. Luciano Floridi, Oxford University Press, 2010

The Unreasonable Effectiveness of Data

- In 2009, Google researchers published a paper highlighting the virtues of data in successfully tacking complex language-related problems.

- *The Unreasonable Effectiveness of Data (2009) [link]*
  Alon Halevy, Perter Norvig, Fernando Pereira

  - *"[I]nvariably, simple models and a lot of data trump more elaborate models based on less data"*

  - *"simple n-gram models or linear classifiers based on millions of specific features perform better than elaborate models that try to discover general rules"*

An Era of Information Abundance

- Information consumes attention — leading to attention scarcity
  - *"In an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it."*
    Herbert A. Simon (1971!)

- Information is complemented by analysis — information growth results in growth in information analysis
  - *"The ability to take data – to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it's going to be a hugely important skill in the next decades, not only at the professional level but even at the educational level for elementary school kids, for high school kids, for college kids. Because now we really do have essentially free and ubiquitous data. So the complimentary scarce factor is the ability to understand that data and extract value from it."*
    Hal Varian (2009)

Information Processing and Retrieval Tasks

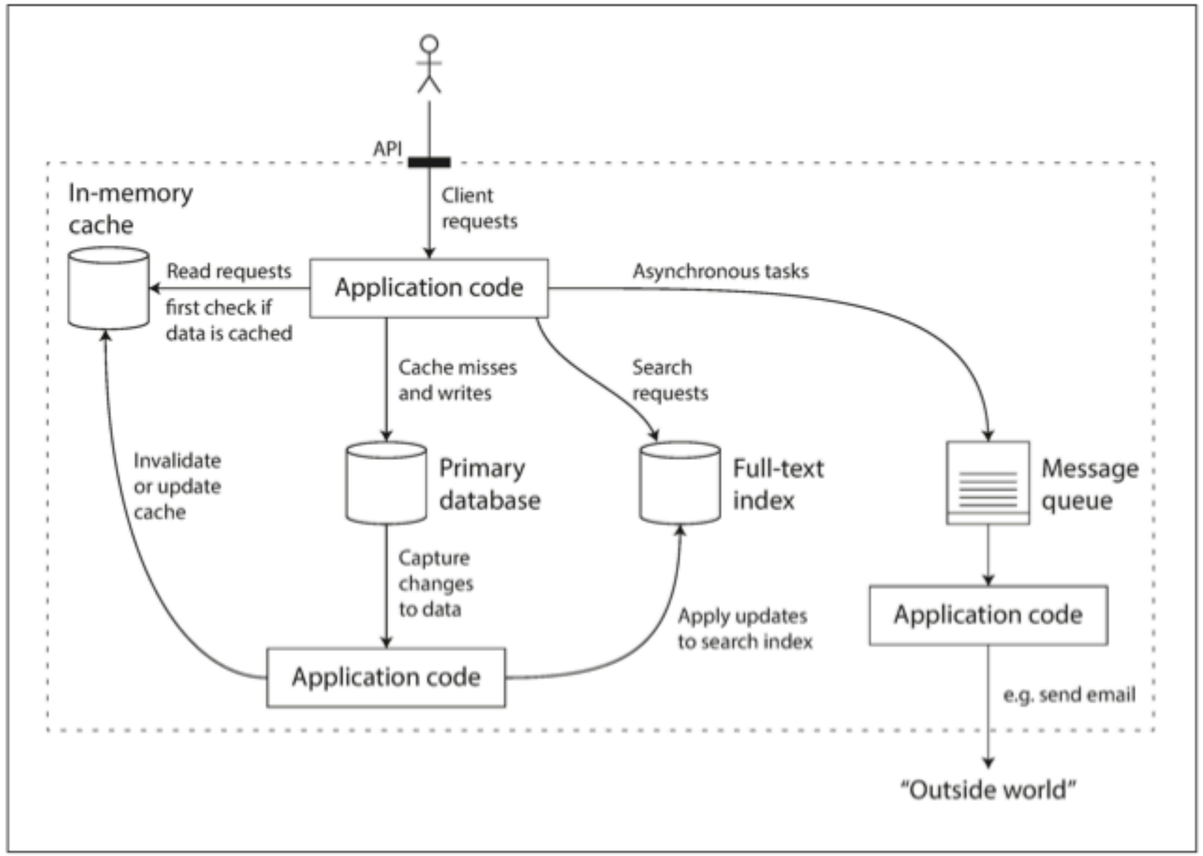


Figure 1-1. One possible architecture for a data system that combines several components.

- Classic workflow and typical tasks.
- Data Ingestion
  - Collect data
  - Describe data
  - Move data
- Data Transformation
  - Data modeling
  - Data migration
  - Pipeline orchestration
- Data Optimization
  - Selection, export, assessment

Image from Designing Data-Intensive Applications O'Reilly, 2017

Course Presentation

PRI Team, 21/22 Edition

Sérgio Nunes
(regente)

João Damas

Sara Fernandes

PRI @ M.EIC

- This is the second edition of PRI
- M.EIC 1st-year course from the area of Information Systems
- 6 classes / ~140 students

- Lectures will be on Mondays, 18h00, in person.
- Practical classes will be in person, throughout the week

Previous Year

Course Objectives

- Prepare students to know, understand, design and develop solutions for information processing and retrieval
  - Make students aware of the challenges associated with building information search systems
  - Familiarize students with the main concepts and techniques associated with information processing and retrieval
  - Enable students to design, implement and evaluate information search systems on document collections

Learning Outcomes

- Identify and describe the main tasks associated with information processing and retrieval
- Describe the architecture and functioning of an information search system
- Describe the tasks associated with the processing phases of a collection (offline) and interrogation processing (online)
- Distinguish the different information retrieval models, identifying their principles, models for document representation, and similarity measures
- Describe and implement different techniques for indexing information
- Describe and implement different techniques for retrieving and ordering results

Information Processing

- Data sources, data provenance and datasets
- Data acquisition and data exploration
- Data pipelines
- Data processing and extraction
- Data characterization

Information Retrieval

- Field of Information Retrieval: history, basic concepts and tools
- Architecture of IR Systems: indexing and retrieval processes
- IR Models: ranking, boolean model, vector space model
- Evaluation in IR: methods and metrics
- Web IR: link analysis, classic algorithms
- Tentative: Entity-Oriented Search, User Interfaces for Search, Applications

Classes

- Lectures, 2h
  - Topic presentation and discussion.
- Practical, 2h
  - Brief presentation of working guides and tutorials
  - Status report with each group
- We plan on having a set of invited talks at the end of the semester (tbd)

Evaluation

- Distributed evaluation with final exam
- Final grade =
  - 60% Group Project +
  - 40% Exam
- Group Project =
  - 20% Data Processing (M1) +
  - 40% Information Retrieval (M2) +
  - 40% Search System (M3)
- The final grade of the project can vary between members of the same group, by plus or minus 3 values, based on the opinion of the teachers and in the self- and hetero- assessment to be carried out internally in each group.
- Minimum grade of 40% (8) required (but not sufficient!) in each of the components.

Main Bibliography

- Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze.
  Introduction to Information Retrieval. Cambridge University Press, 2008
  https://nlp.stanford.edu/IR-book/information-retrieval-book.html

- W. Bruce Croft, Donald Metzler, Trevor Strohman.
  Search Engines: Information Retrieval in Practice. Pearson Education, 2015
  https://ciir.cs.umass.edu/irbook/

- Kristian Balog. Entity-Oriented Search. Springer, 2018
  http://link.springer.com/978-3-319-93935-3

- Martin Kleppmann. Designing Data-Intensive Applications. O'Reilly, 2017

- Jeroen Janssens. Data Science at the Command Line, 2nd Edition, O'Reilly, 2021
  https://www.datascienceatthecommandline.com/2e/

Group Project

Group Project

- Design and implementation of an information processing and retrieval system
- The project is developed in groups of 3 students and starts with the selection of a topic and the relevant data sources
- The project is organized in deliveries and partial presentations, which correspond to the project development phases
  - Milestone 1: Information Processing (week of Oct 10th)
  - Milestone 2: Information Retrieval (week of Nov 14th)
  - Milestone 3: Search System (week of Dec 12th)

Milestones

- Each project delivery (milestone) has a corresponding presentation and discussion
- Electronic submissions of the project deliverables are accepted up to 18:00 on the day before the in-class presentation
- Reports are written as short scientific papers, using a two-column format (4 pages max in each delivery). Each report is a self-contained work-in-progress and is based on the previous deliveries
- In the weeks assigned to project presentations, the practical class will be organized in a workshop format, with project presentations and discussions according to an established schedule
- The final project evaluation corresponds to a weighted average of the milestones evaluations.

M1: Information Processing

- The first milestone is achieved with the preparation and characterization of the datasets selected for the project
- Work on these tasks depends on the nature, volume, organization and accessibility of the selected datasets. As a result of this milestone, a well-documented and reproducible pipeline of data processing is expected
    - search repositories for datasets
    - select convenient data subsets
    - assess the authority of the data source and data quality
    - perform exploratory data analysis
    - prepare and document a data processing pipeline
    - characterize the datasets, identifying and describing some of their properties
    - identify the conceptual model for the data domain
    - identify follow-up information needs in the data domain (**important**)

M2: Information Retrieval

- The second milestone is achieved with the implementation and use of an information retrieval tool on the project datasets and its exploration with free-text queries
- This task makes use of state-of-the-art retrieval tools and involves the view of the datasets as collections of documents, the identification of a document model for indexing, and the design of queries to be executed on the indexed information
    - choose the information retrieval tool (Solr, Elasticsearch, ...)
    - analyze the documents and identify their indexable components
    - use the selected tool to build the indexes
    - use the selected tool to configure and execute the queries
    - demonstrate the indexing and retrieval processes
    - manually evaluate the returned results
    - evaluate the results obtained for the defined information needs

M3: Search System

- The third milestone is achieved with the development of the final version of the search system
- This version is an improvement over the previous milestone, making use of features and techniques with the goal of improving the quality of the search results
- For this milestone, each group is expected to explore innovative approaches and ideas, and will heavily depend on the context and data of each group
- Additionally, an extended evaluation of the results and a comparison with the previous version of the search system is also expected
- Examples of topics to explore include: incorporate new information retrieval algorithms; expand the information available for each document by adding and linking new datasets; work on user interfaces by developing a frontend for the search system

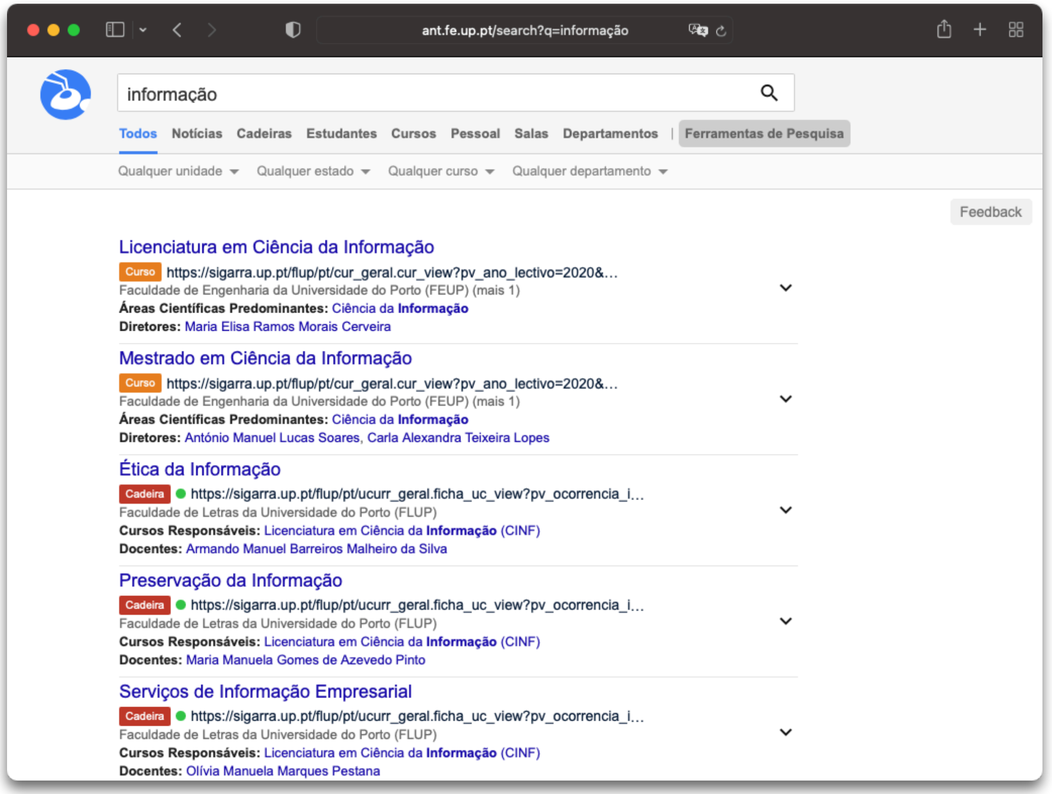
Project Themes

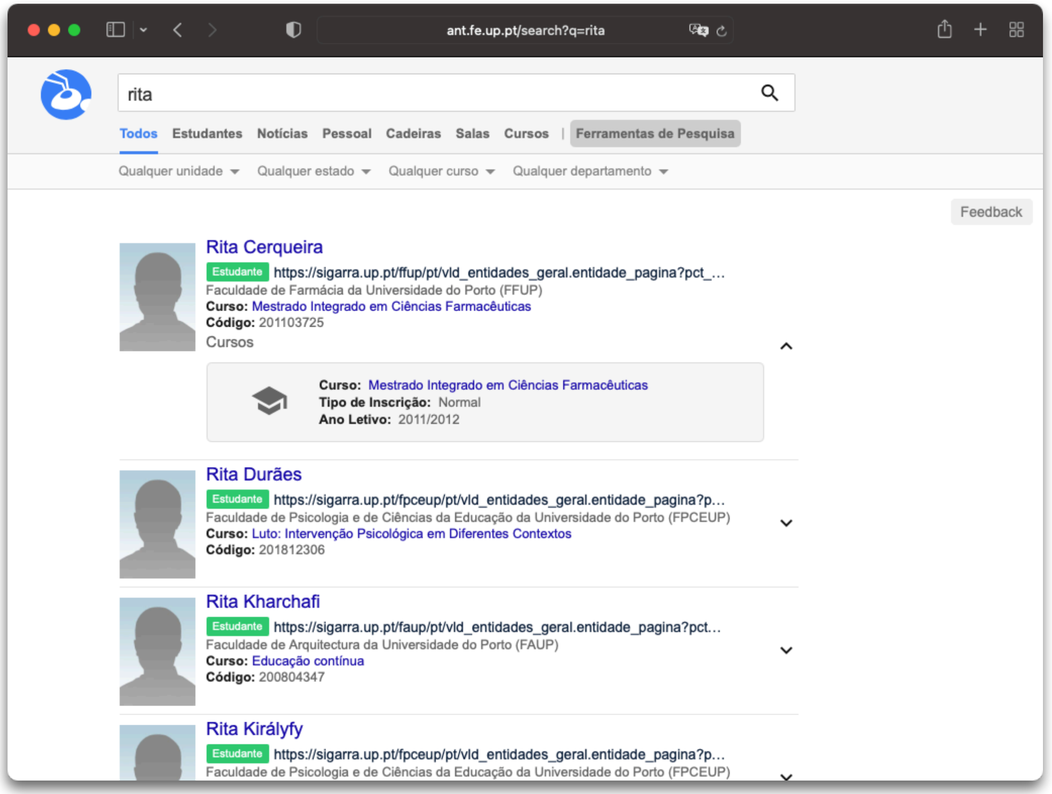
- Project topics are free, but cannot be repeated in the same class
- Need to be approved by the end of the **first practical class**
- Data source(s) must be of unstructured nature and rich in textual data
- Consider your personal interests and motivations

- Many possibilities: education, sports, law, government, media ...

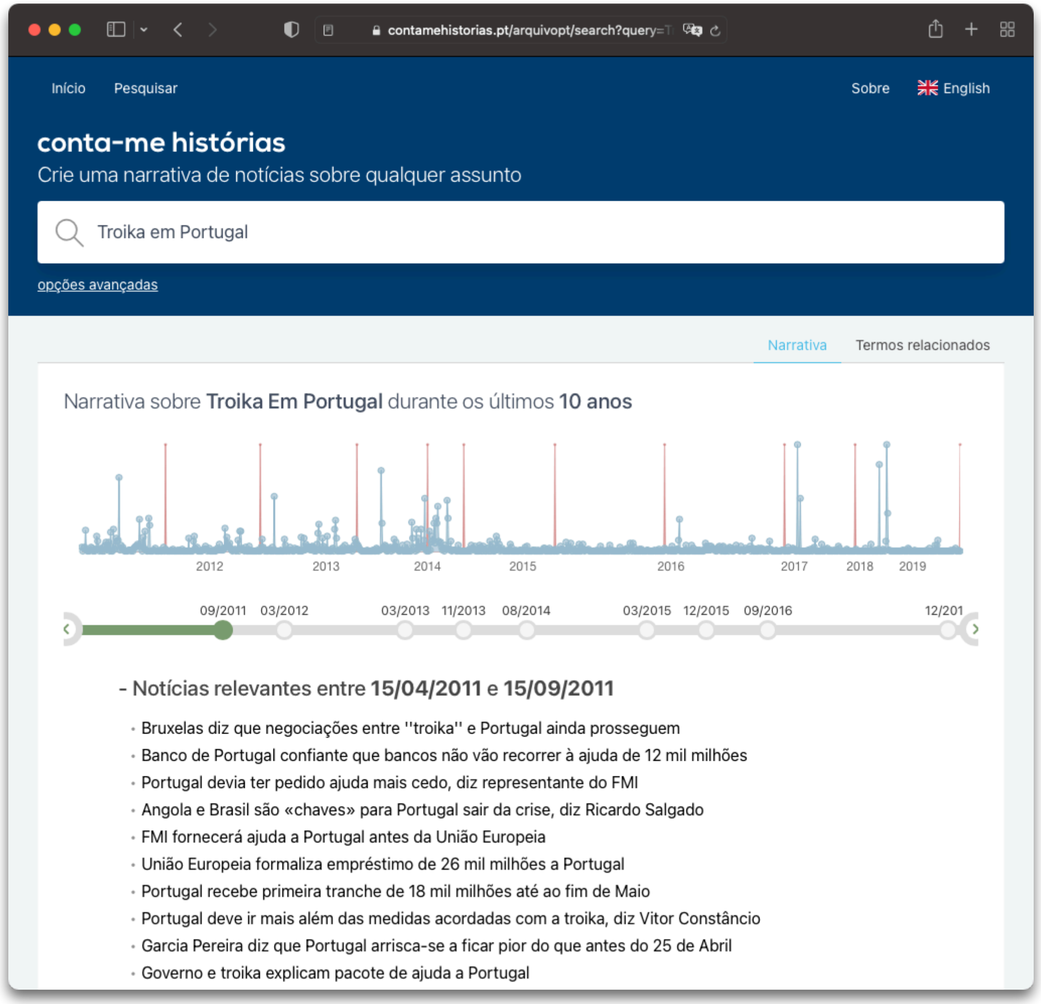
Project Examples

ANT


ant.fe.up.pt/search?q=informação
informação
Todos Notícias Cadeiras Estudantes Cursos Pessoal Salas Departamentos | Ferramentas de Pesquisa
Qualquer unidade Qualquer estado Qualquer curso Qualquer departamento
Feedback
Licenciatura em Ciência da Informação
Curso https://sigarra.up.pt/flup/pt/cur_geral.cur_view?pv_ano_lectivo=2020&...
Faculdade de Engenharia da Universidade do Porto (FEUP) (mais 1)
Áreas Científicas Predominantes: Ciência da Informação
Diretores: Maria Elisa Ramos Morais Cerveira
Mestrado em Ciência da Informação
Curso https://sigarra.up.pt/flup/pt/cur_geral.cur_view?pv_ano_lectivo=2020&...
Faculdade de Engenharia da Universidade do Porto (FEUP) (mais 1)
Áreas Científicas Predominantes: Ciência da Informação
Diretores: António Manuel Lucas Soares, Carla Alexandra Teixeira Lopes
Ética da Informação
Cadeira https://sigarra.up.pt/flup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_i...
Faculdade de Letras da Universidade do Porto (FLUP)
Cursos Responsáveis: Licenciatura em Ciência da Informação (CINF)
Docentes: Armando Manuel Barreiros Malheiro da Silva
Preservação da Informação
Cadeira https://sigarra.up.pt/flup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_i...
Faculdade de Letras da Universidade do Porto (FLUP)
Cursos Responsáveis: Licenciatura em Ciência da Informação (CINF)
Docentes: Maria Manuela Gomes de Azevedo Pinto
Serviços de Informação Empresarial
Cadeira https://sigarra.up.pt/flup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_i...
Faculdade de Letras da Universidade do Porto (FLUP)
Cursos Responsáveis: Licenciatura em Ciência da Informação (CINF)
Docentes: Olívia Manuela Marques Pestana


ant.fe.up.pt/search?q=rita
rita
Todos Estudantes Notícias Pessoal Cadeiras Salas Cursos | Ferramentas de Pesquisa
Qualquer unidade Qualquer estado Qualquer curso Qualquer departamento
Feedback
Rita Cerqueira
Estudante https://sigarra.up.pt/ffup/pt/vld_entidades_geral.entidade_pagina?pct_...
Faculdade de Farmácia da Universidade do Porto (FFUP)
Curso: Mestrado Integrado em Ciências Farmacêuticas
Código: 201103725
Cursos
Curso: Mestrado Integrado em Ciências Farmacêuticas
Tipo de Inscrição: Normal
Ano Letivo: 2011/2012
Rita Durães
Estudante https://sigarra.up.pt/fpceup/pt/vld_entidades_geral.entidade_pagina?p...
Faculdade de Psicologia e de Ciências da Educação da Universidade do Porto (FPCEUP)
Curso: Luto: Intervenção Psicológica em Diferentes Contextos
Código: 201812306
Rita Kharchafi
Estudante https://sigarra.up.pt/faup/pt/vld_entidades_geral.entidade_pagina?pct...
Faculdade de Arquitectura da Universidade do Porto (FAUP)
Curso: Educação contínua
Código: 200804347
Rita Királyfy
Estudante https://sigarra.up.pt/fpceup/pt/vld_entidades_geral.entidade_pagina?p...
Faculdade de Psicologia e de Ciências da Educação da Universidade do Porto (FPCEUP)

Conta-me Histórias



Prémio Arquivo.pt

- *O objetivo é galardoar trabalhos inovadores que utilizem informação histórica preservada da Web.*
- *Estes trabalhos deverão utilizar os serviços de pesquisa e acesso disponibilizados publicamente pelo Arquivo.pt e demonstrar claramente a utilidade do serviço.*
- *Não há restrição de tema. Porém, é necessário ter o Arquivo.pt como fonte primordial de informação.*
- *O concurso está aberto a todos os interessados, a título individual ou em grupo.*

- https://arquivo.pt/premios

Data Sources

- A diversity of data sources exist.
- Keep in mind when choosing sources that:
  - Direct access, i.e. not dependent on third party actions
  - Volume, ideally thousands
  - Rich in textual data, i.e. long texts, not just labels or titles
- List of datasets sources

Materials

- The course's web page in Moodle is the starting point
- For each lecture and lab class an information page is available
- Moodle is also used for:
  - Group registration
  - Announcements and discussion
  - Submission of milestone materials (report and presentation)
- Slack:
  - Last minute warnings and in-group communication

PRI Tutorials

- You have access to a collection of tutorials to guide to introduce concepts and tools, and guide your explorations during the semester.
- https://git.fe.up.pt/pri/tutorials
- Start this week with the first tutorial — Command Line Practice

- *A work in progress. Your feedback is welcomed.*

Next steps

- Answer 'PRI Survey' (if you haven't done so)
- Read the project rules
- Prepare for the first practical class:
  - organize groups before class (3 students) — register in Moodle (you can change later)
  - explore data sources and contexts to identify datasets
- Explore the first PRI tutorial — Command Line

- First delivery in four weeks (10th October week) - Milestone 1: Data Processing.
- **Next week lecture is recorded.**

Questions or comments?

Data Collection

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Today's Plan

- Context and motivation for data processing
- Focus on data collection and preparation
  - Data collection and preparation tasks
  - Overview of a typical pipeline
  - Example projects
- Review of the first milestone delivery
- Review the plan for the next practical class

Overview

"Data"

- In Latin, data is the plural of datum.
- In specialized fields, data is treated as plural, e.g. "data were collected".
- Generally, it is treated as a mass noun, like "information", e.g. "data was collected".
- We adopt the use of "data" as mass noun.

Terminology: Data, Metadata and Information

- **Data**
  - is a measurement of something on a scale;
  - a fact known by direct observation.
- **Metadata**
  - is "data about data";
  - not the content of data but data providing information about one or more aspects of the data, such as description (date, time, author), structure (format, version), administrative (permissions), legal, etc.
- **Information**
  - is data with a context / meaning, thus enabling decision making;
  - is data that has been processed, organized and structured.

Terminology: Decimal and Binary Systems

- The binary system uses power of 2 units.
- The decimal system uses power of 10 units.
- In the International System of Units standard, kilo, mega, giga, correspond to powers of 1000 — thus decimal prefixes.
- Historically, the computer industry used the same prefix with two different meanings, i.e. 1MB could either be 1 048 576 bytes (binary) or 1 000 000 bytes (decimal).
- In 2008, binary prefixes — i.e. that refer to powers of 1024 — were officially introduced: kiwi (Ki), mebi (Mi), gibi (Gi), tebi (Ti), pedi (Pi), exbi (Ei), zebi (Zi), yobi (Yi).
  - 1MiB (1024^2) = 1 048 576 bytes
  - 1MB (1000^2) = 1 000 000 bytes

"Binary prefix" on Wikipedia: https://en.wikipedia.org/wiki/Binary_prefix

Out of Scope

- Outside the scope of this course are:
  - Ethical dimensions of data and information
  - Economic aspects of data
  - Legal aspects
  - ....
- Note that these concepts should not be foreign to informatics or computer science.
- Informatics engineers need to be aware of many of these aspects, particular those with strong social impact, in their work. There are many recent examples, e.g. social media, cloud services.

- Possible topics for invited talks.

Information Life Cycle

- In modern societies, progress and welfare is increasingly dependent on the successful and efficient management of the life cycle of information.
- The life cycle of information typically includes the following phases:
  - Occurrence: discover, design, author, etc;
  - Transmission: networking, accessing, retrieving, transmitting, etc;
  - Processing and Management: collecting, validating, modifying, indexing, classifying, filtering, sorting, storing, etc;
  - Usage: monitoring, explaining, planning, forecasting, decision-making, educating, learning, etc;
- Information and Communication Technologies (ICT) evolved from being mainly recording systems, to being communication systems, to also (and currently) being processing and producing systems.

Information Life Cycle

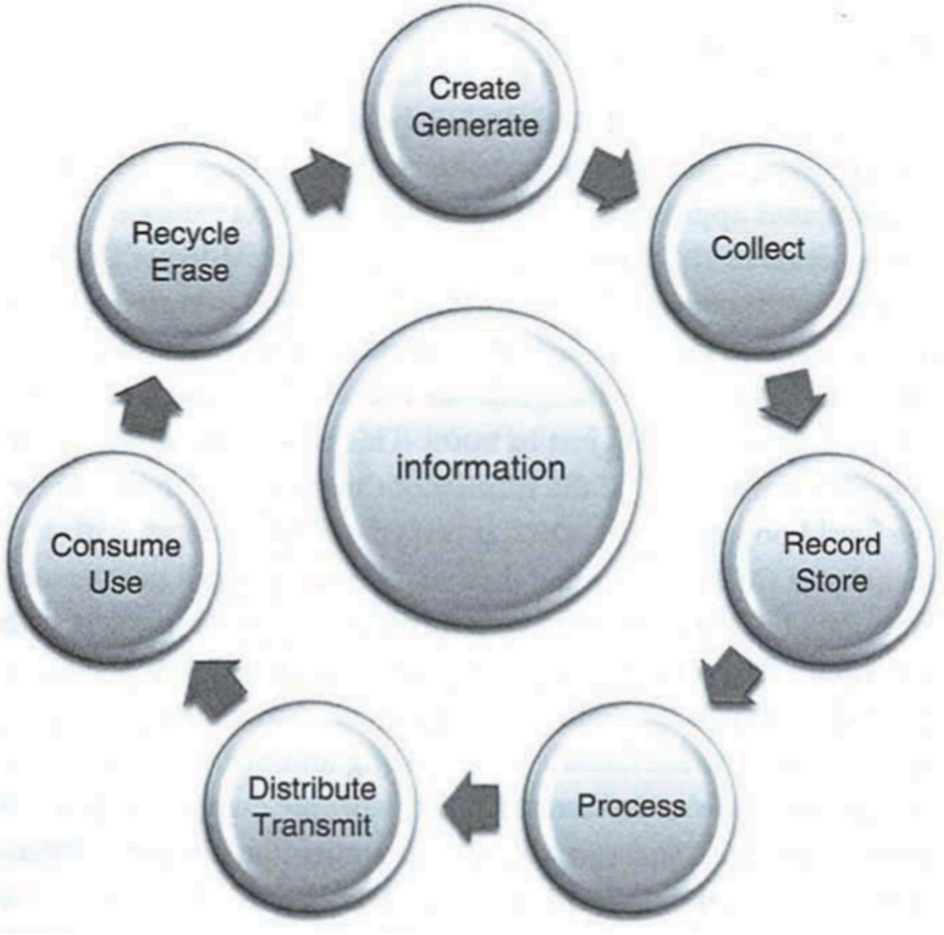


Image from Floridi, L. Information - A Very Short Introduction (2010).

The zettabyte* era

- A study from 2003, reported that humanity had accumulated approximately 12 exabytes of data until the emerge of the personal computer.
- Of these, 92% were stored on magnetic media (i.e. digital media).
- A more recent study from 2018, reported that "*the total amount of data created, captured, copied and consumed in the world was 33 zettabytes (ZB) – the equivalent of 33 trillion gigabytes. This grew to 59ZB in 2020 and is predicted to reach a mind-boggling 175ZB by 2025. One zettabyte is 8,000,000,000,000,000,000,000 bits*" **
- Also, "*there are around 600 hyperscale data centres – ones with over 5,000 servers – in the world. Around 39% of them are in the US, while China, Japan, UK, Germany and Australia account for about 30% of the total*" **
- With these trends, i.e. an annual growth rate of 50%, ~150 years from now the number of bits would surpass the number of atoms on Earth …

** 1 zettabyte = 1000 exabytes; 1 exabyte = 1000 petabytes; 1 petabytes = 1000 terabytes.*

*** Vopson, M. The world's data explained: how much we're producing and where it's all stored. The Conversation, 2021*
https://theconversation.com/the-worlds-data-explained-how-much-were-producing-and-where-its-all-stored-159964

Value in Data

- "Data is the new oil", *everybody* (circa 20xx).
- Data is a source of value generation, providing evidence and content for the design of new products, new processes, and contribute to more efficient operations.
- In data-driven approaches, multidisciplinary teams experiment and explore large and diverse sources of data to "extract signal from the noise".
- Indirect value — data provides value by influencing of supporting decisions, e.g. risk analysis in insurance, purchase decisions in retail.
- Direct value — data provides value by feeding automated systems, e.g. search system, product recommendation system.

Data Value Funnel

- A large number of data sources and exploratory analysis are required to produce a single valuable application of data.
- Minimize the time spent on non-relevant data by empowering business-experts (i.e. people who know about the business) to explore data.
- Additionally, make data analysis processes as efficient as possible, e.g. by implementing effective data processing workflows.

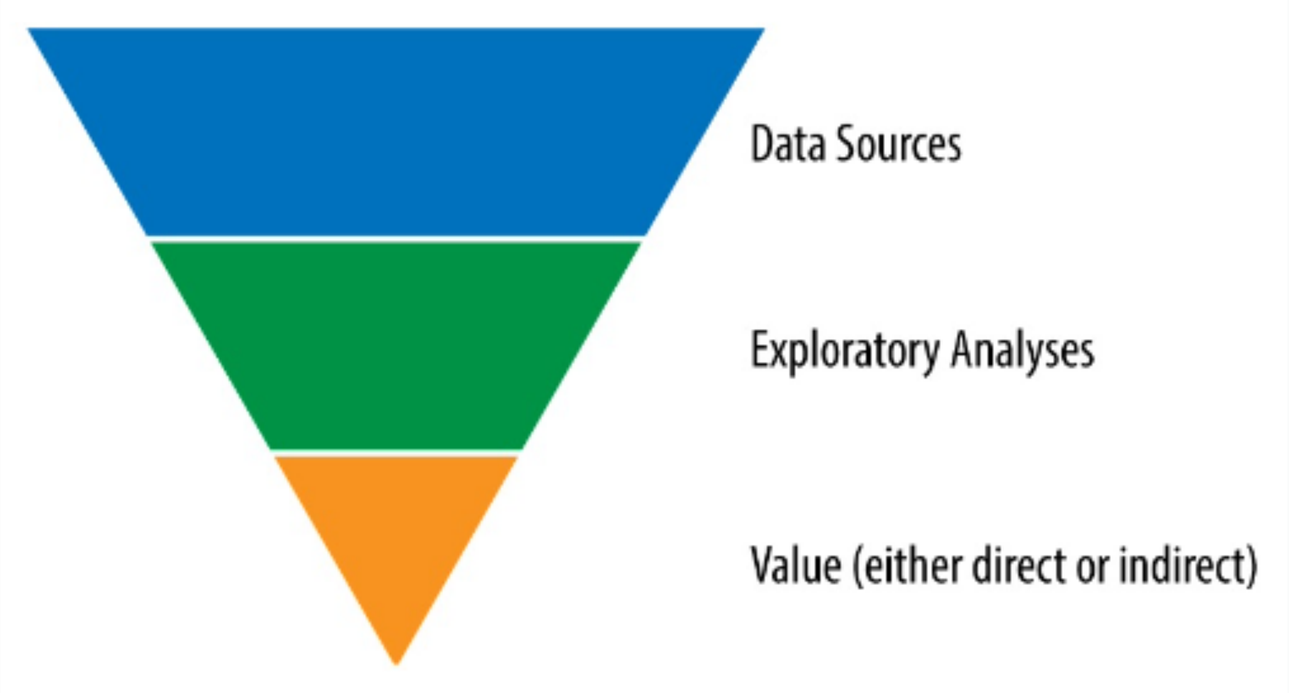


Image from Densmore, J. Data Pipelines Pocket Reference (2021).

Increasing Data Value

- Make data available, i.e. simply make previously unaccessible data, available.
- Combine data, i.e. create a single coherent whole from disperse data sources; e.g. collection of news from the main news outlets during a year.
- Clean data, i.e. eliminate problems such as incomplete values, duplicates; or create a subset according to specific criteria.
- Structure data, i.e. provide structure to unstructured data; e.g. derive a mentioned entities field from a textual field.
- Enrich data, i.e. complement existing data with data from other sources, including the computation necessary to do so.

Data-Intensive Applications

- Many applications today are data-intensive, as opposed to computing-intensive.
- In this context, existing problems are:
  - the amount of data available;
  - the complexity of the data;
  - the speed at which it changes.
- Common building blocks in data-intensive application include:
  - Store data, for use or sharing (databases);
  - Remember the result of an expensive operation (caches);
  - Enable search and filtering (indexes)
  - Send messages between systems (stream processing);
  - Periodically process large amount of data (batch processing);

Example of a Data System

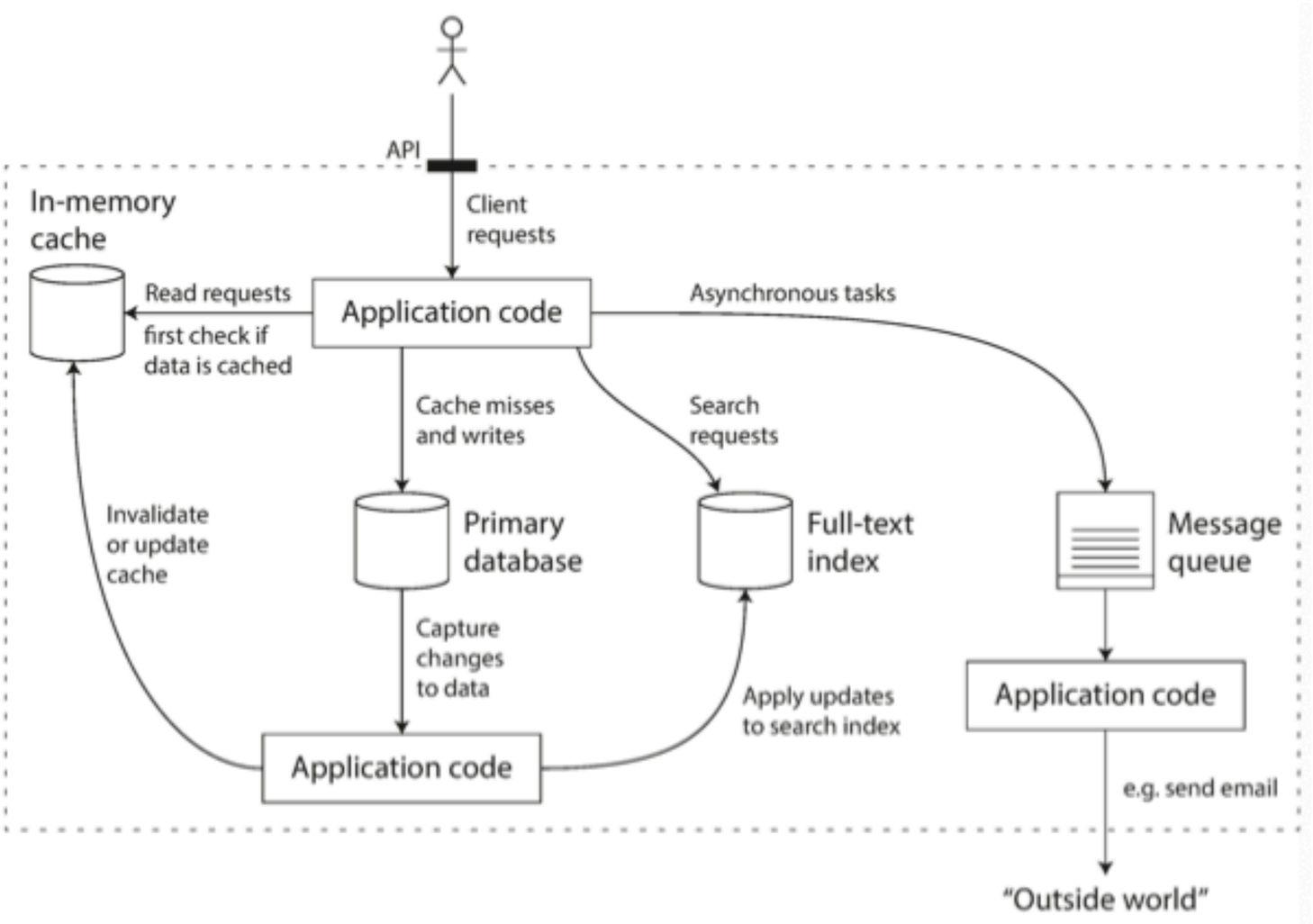


Image from Kleppmann, M. Designing Data-Intensive Applications (2017).

Data Stages

- Data moves through three main stages:
  - Raw — focus is on data discovery; the primary goals are ingestion, understanding, and metadata creation; common questions include: what kinds of records are in the data? how are record fields encoded?
  - Refined — focus is on data preparation for further exploration; tasks include removing unwanted parts, reshaping poorly formatted elements; establishing relationships between datasets; assessing data quality issues.
  - Production — focus is on integrating the data into production processes or products.
- Several data processing pattern exist in the literature, including: ETL, ELT, OSEMN.

ETL Pattern

- The ETL framework (extract-transform-load) was coined in the 1970s and popularized in the context of data warehousing.
  - Extract, involves extracting data from the source system.
  - Transform, a series of operations or transformations are applied to the extracted data.
  - Load, involves publishing data to the target system, either simples flat files or other infrastructures.
- ETL is usually associated with classic centralized IT driven operations.

ELT and EtLT Frameworks

- ELT (extract-load-transform) is a recent evolution over the ETL framework.
- Increasingly common access to data storage infrastructures capable of handling large volumes of data has lead to a more flexible pattern.
- Column-oriented data structures are particularly well-suited to typical data processing tasks, i.e. organizing operations per field or property.
- The sub-pattern EtLT introduces a transformation step before the loading, typically associated with data cleaning tasks.
- Load-transform, in contrast with transform-load, is a pattern more well-suited to the division of responsibilities in multidisciplinary teams.

ELT Pattern

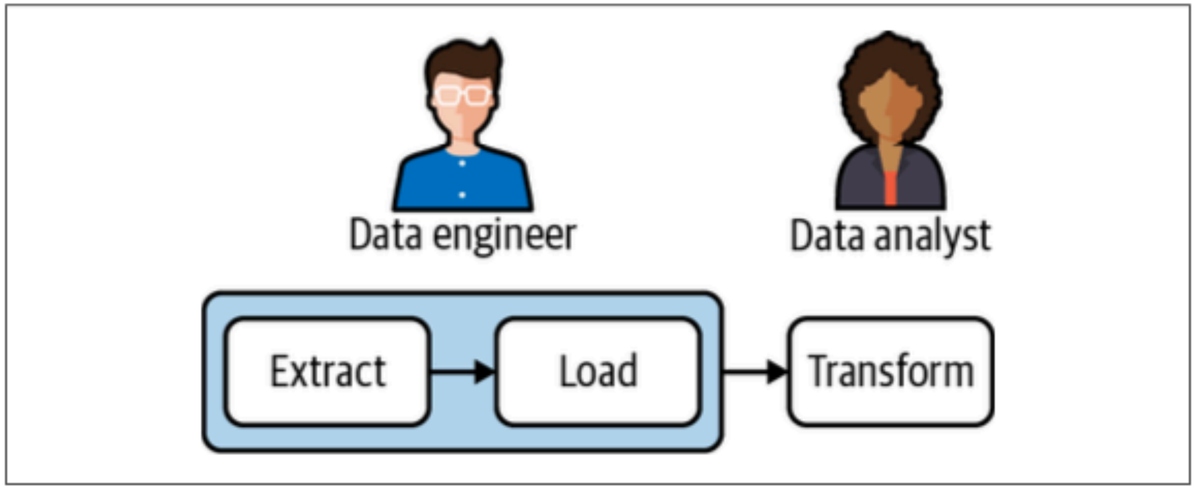


Figure 3-3. The ELT pattern allows for a clean split of responsibilities between data engineers and data analysts (or data scientists). Each role can work autonomously with the tools and languages they are comfortable in.

Image from Densmore, J. Data Pipelines Pocket Reference (2021).

OSEMN Framework

- In the context of Data Science, the OSEMN (pronounced awesome) was coined.
  - Obtain, gathering data.
  - Scrub, clear, arrange, prepare data.
  - Explore, observe, experiment, visualize.
  - Model, create a statistical model of the data.
  - Interpret, drawn conclusions, evaluating and communicating results.

- Although presented as a series of steps, real-word processes are typically non-linear.

Iterative Process

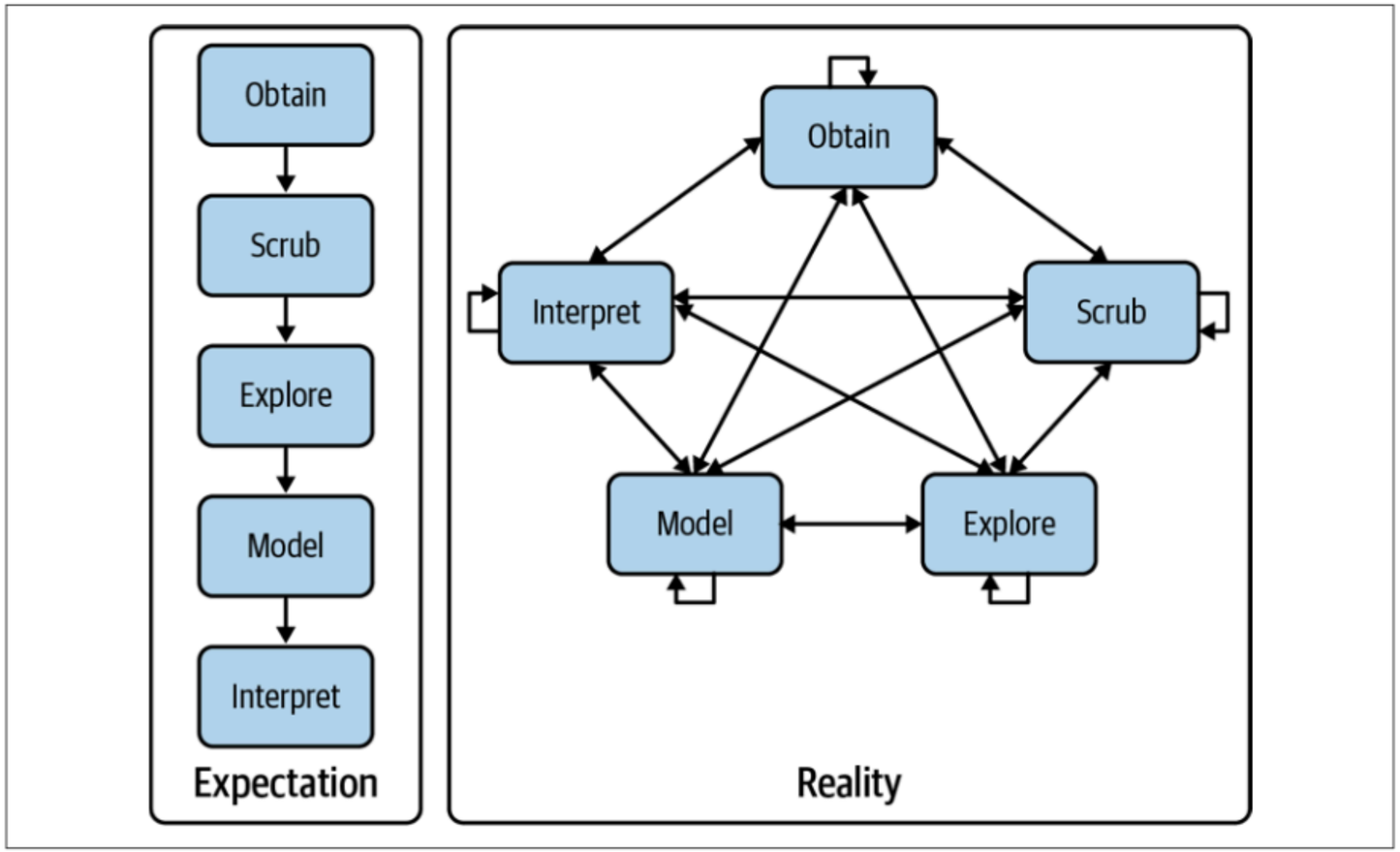


Figure 1-1. Doing data science is an iterative and nonlinear process

Image from Janssens, J. Data Science at the Command Line (2021).

Data Engineers

- Data engineers have emerged as an autonomous key role in this context.
- Design, implement and maintain data processing pipelines.
- Work closely with data scientists and analysts to understand what will be done with the data.
- Wide range of technical skills:
  - SQL and Data Warehousing
  - Programming - Python, Java, Go (common in this context)
  - Distributed Computing
  - Cloud Infrastructures
  - System Administration

Data Collection

Diversity of Data Sources

- Data sources vary across many dimensions:
  - Ownership — either owned or from third-parties; understanding ownership is central, i.e. know what data you have access to and what you can do with it;
  - Ingestion interface and structure — how do you get the data and in what form is in;
  - Volume —in each step of the pipeline, volume needs to be taken into account; high- and low- volume are difficult to define and depend on available infrastructures and algorithms;
  - Cleanliness and validity —duplicate data, missing or incomplete data, encoding, etc;
  - Latency and bandwidth of the source — need to consider internal update requirements and also source system limits, speed, timeouts, etc.

Open Data

- The idea that data should be freely available to anyone, to use, modify, and republish for any purpose.
- Associated with a movement, see Open Knowledge Foundation, https://okfn.org/
- One of the most important forms of open data is open government data.
- Open data can also be linked, known as linked open data (LOD), see https://www.w3.org/DesignIssues/LinkedData.html (2009)
- "Web of data" is an expression to represent the set of technologies and practices that enable a space where data can be automatically discovered and accessed by machines.
- Also related is the concept of FAIR: findable, accessible, interoperable, and reusable; emphasizing machine-actionability over data.
- FAIR/O is used to indicate that a data source complies with FAIR and is also of open nature.

Data Sources (Examples)

- Structured
  - Google Dataset Search, https://toolbox.google.com/datasetsearch
  - dados.gov, https://dados.gov.pt
  - Dados Abertos em Portugal, http://dadosabertos.pt
  - Dados Abertos do Parlamento, https://www.parlamento.pt/Cidadania/Paginas/DadosAbertos.aspx
- Unstructured
  - Legislação Portuguesa Consolidada, https://dre.pt/web/guest/legislacao-consolidada
  - Acórdãos do Tribunal Constitucional, http://www.tribunalconstitucional.pt/tc/acordaos/

- *See Moodle for a list of selected data sources and example datasets.*

Data Selection - Things to Consider

- Is the author a trustable source that can be contacted?
- Is the data regularly updated?
- Does the data include information about how, when it was acquired?
- Does it seem plausible through observation?

Ingestion Interfaces and Data Structures

- Examples of ingestion interfaces include:
  - A relational database behind an application, such as PostgreSQL, SQLite or Oracle;
  - A layer of abstraction on top of a system, such as a REST API;
  - An endpoint to a message queue system, such as RabbitMQ;
  - A shared network filesystem, containing logs, CSV files, or other flat files.
- Examples of data structures include:
  - JSON from REST API;
  - Well-structured data from a relational database;
  - Semistructured log data;
  - CSV (comma-separated values) datasets;
  - PDF, or other proprietary format, files;
  - HTML, or other semi-structured, files;

Data Formats

- Data formats enable the representation of different types of data in a computer-usable form.
- Common data representations:
  - Alphanumeric: Unicode, ASCII
  - Image (bitmap): PNG, JPG
  - Image (object / vector): PostScript, SVG
  - Sound: AVI, MP3, AAC
  - Documents: PDF, HTML, XML
- Formats used by individual applications are known as proprietary formats.
- Proprietary formats can be open if its specifications are published. Although not free of licensing.
- Some proprietary formats become "proprietary standards" when they become de facto standards due to general use.
- An open format is defined by a published specification, usually maintained by a standards organization (e.g. PNG, FLAC).

Data Encoding

- Data can be encoded
  - in memory, in specific structures such as objects, lists, arrays;
  - as a self-contained sequence of bytes, for file storage or network transmission, e.g. JSON document.
- The process of translating from the in-memory representation to a byte sequence is called **encoding** (also known as serialization), and the inverse is called **decoding** (also parsing, deserialization).
- Most programming languages have built-in support for encoding (and decoding) in-memory data to byte sequences, e.g. Java's java.io.Serializable, Python's pickle, PHP's serialize.
- Useful for transient purposes but in general not adequate in data pipelines — limited to a programming language, reduced interoperability, lower performance, etc.

JSON, XML Serialization

- JSON and XML are the most common text-based encoding standards.
- JSON is widely supported by many applications.
- CSV is also popular, but less powerful.
- These are (somewhat) human readable (an advantage).
- Are the *de facto* solution for data interchange, between organization, between applications, as export formats in applications, as API outputs.
- Limitations include: ambiguous support for number formats, limited support for binary data (e.g. images).

Binary Serialization

- Binary serialization is more compact and faster to parse.
- Is a common solution for within organization data exchange.
- There are many binary formats for JSON - MessagePack, BSON, BJSON, etc.
- Apache Thrift (originally Facebook) and Protocol Buffers (protobuf) are open-source binary encoding libraries that produce a binary encoding of a given record.

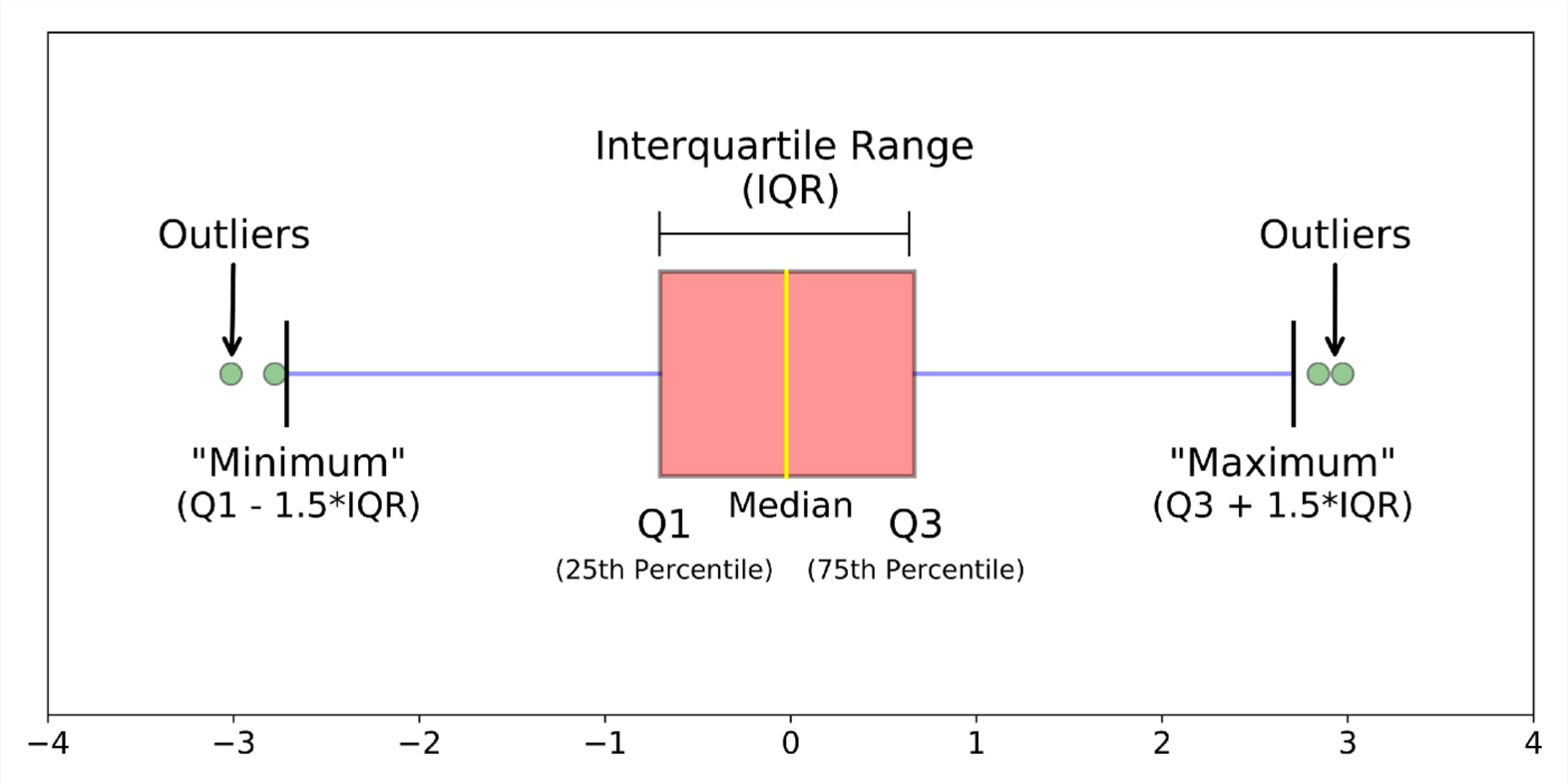
Data Quality

- Common problem affecting data quality:
  - Missing data — due to error, specific meaning (e.g. n/a, 0, NULL).
  - Inconsistent values — distinct timezones in time/dates, multiple units (e.g. m, km).
  - Precision problems — rounding decisions may result in fake patterns (e.g. maps).
  - Duplicate values — due to errors, or valid data.
  - Many other: text encoding problems, mislabeled data, incomplete, outdated, etc.
- There is a need to investigating and understand data properties during the data selection phase — often called data investigation or assessment.

Overall View of the Distribution

- Descriptive statistics are commonly used to begin the investigation.
- Common measures and techniques estimate the:
  - Central tendency, i.e. central, average value of observations:
    - Mean (sum divided by the number of items), median (middle value that separates the observations), mode (most frequent value);
  - Dispersion, i.e. how much the values vary:
    - Standard deviation, interquartile range (difference between values in upper and lower quartile), difference between the maximum and minimum.
- Box plots are a methods that graphically depicts most of these descriptive statistics.

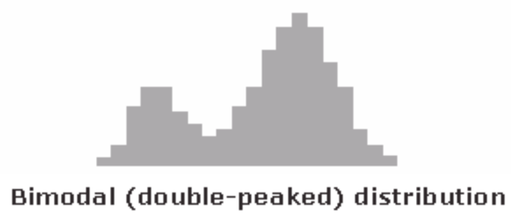
Box plots



- Image from Understanding Boxplots, Towards Data Science
  [https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51]

Frequency Histograms

- For large data volumes, observing distributions of attribute values can be done using histograms, which represent how numerical data is distributed.
- A key aspect of producing histograms is exploring with different bin sizes.
- Numerous distributions exist and are described in detail in statistical literature.



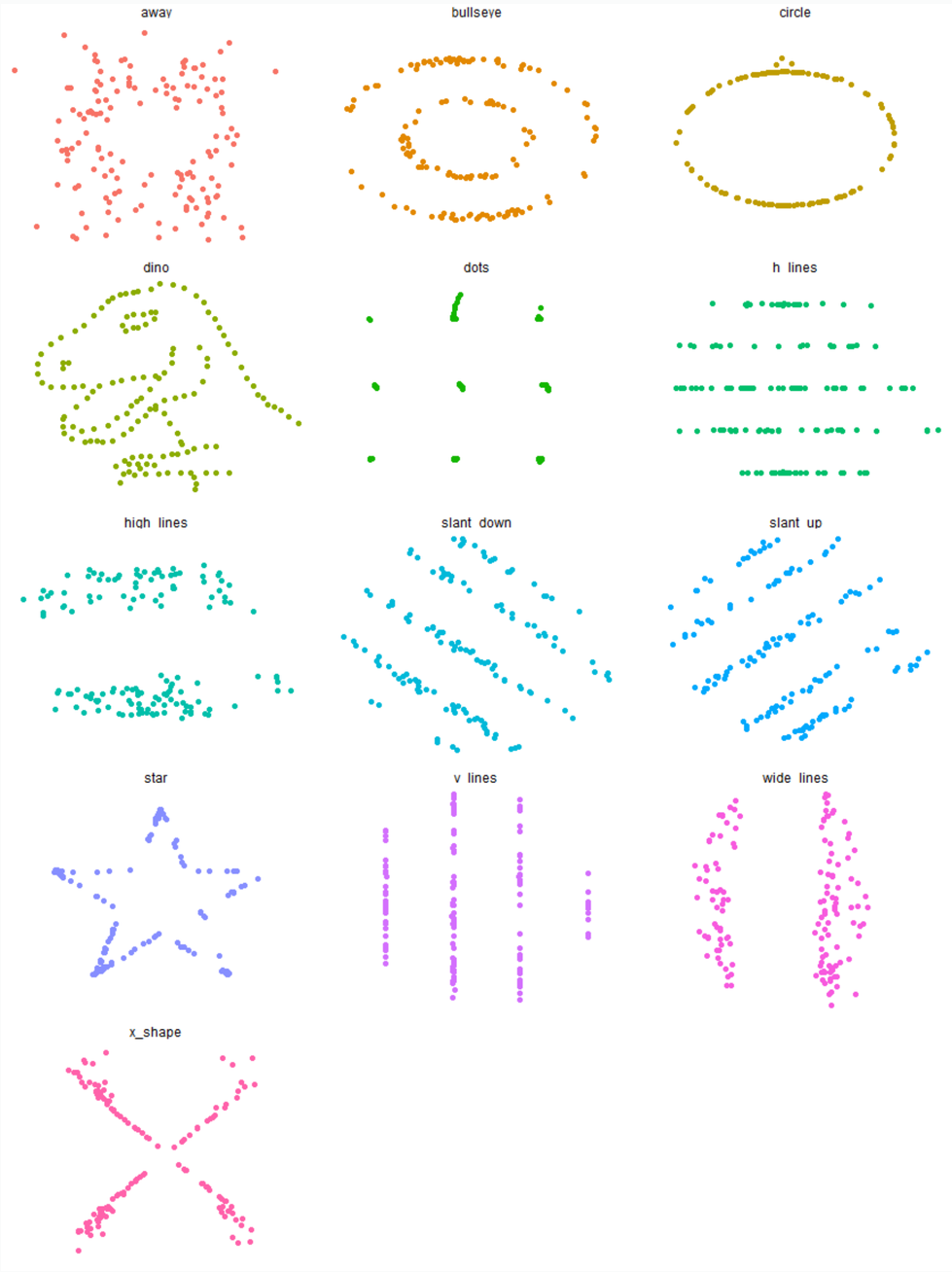
- Examples from "What are Histograms?", ASQ [https://asq.org/quality-resources/histogram]

Descriptive statistics only provide a summary

- The image on the right depicts 13 sets of x-y data, where basic descriptive statistics have the same values, i.e. x-mean, y-mean, x-std, y-std), but look very different.

- Don't reply only on descriptive statistics, include exploratory visualization in your process.

- Image from the The Datasaurus data package [https://cran.r-project.org/web/packages/datasauRus/vignettes/Datasaurus.html]



Outliers

- Outliers are items that differ significantly from others.
- It is necessary to understand if these values are exception but valid cases or if are errors that need to be removed. Expert domain-knowledge if often necessary.
- Errors resulting in outliers may be the result of:
  - Problems in the data collection procedure;
  - Hardware or software problems in data collection tools;
  - Human mistakes in data recording.
- Outliers may significantly distort descriptive statistics or visualizations.

Missing Data

- Missing data is an important aspect of data quality that always needs a detailed investigation to determine its origin and impact on following steps.
- Isolated instances of missing data aren't usually a problem, however if the missing data is not randomly distributed or occurs in large numbers globally or in specific variables, the data set will be biased and thus not appropriate for a valid analysis.
- Missing data can also be an indicator of flaws in the data collection process.

Data Quality Summary

- Investigating the properties of the data at its origin and at different points of the data processing pipeline is a key aspect of a solid data-based project, helping on:
  - Deciding on the data sources to select;
  - Determining possible bias and limitations of the data *a priori;*
  - Detecting and correcting problems in the pipeline;
  - Framing the conclusions or built products;
- Data quality investigations should rely on multiple methods, from descriptive statistics to exploratory visualization.
- Best practices: clean and validate in the best system to do so; validate often.

Tools for Data Collection

- SQL for extracting data from databases.
- Custom code for APIs.
- Unix commands to work with external sources, e.g. curl.
- Web crawling platforms:
  - Scrapy, https://scrapy.org/ [Python]
  - Internet Archive Heritrix, https://github.com/internetarchive/heritrix3 [Java]
  - Apache Nutch, http://nutch.apache.org [Java]

Bibliography and Further Reading

- **Visual Analytics for Data Scientists** [VADS20]
  Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C., and Wrobel, S. Springer, 2020

- **Designing Data-Intensive Applications** [DDIA17]
  Kleppmann, M. O'Reilly, 2017

- **Principles of Data Wrangling** [PDW17]
  Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., and Carreras, C. O'Reilly, 2017

- **Data Pipelines Pocket Reference** [DPPR21]
  Densmore, J. O'Reilly, 2021

- **Information - A Very Short Introduction** [IVSI10]
  Floridi, L. Oxford University Press, 2010

Questions or comments?

Data Preparation

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Today's Plan

- Focus on data preparation
  - Data preparation tasks
  - Overview of a typical pipeline
  - Example projects

- Review of the first milestone delivery
- Review the plan for the next practical class

Quick Review

Status

- Starting the 3rd week of classes (2nd practical class).
- Lectures: data collection >> **data preparation** >> data processing
- Practical: groups and topics >> **start working data collection and preparation**
- Milestone 1 workshop in 2 weeks.

Project

- The development of an **information search system**, including work on
  - data collection and preparation,
  - information querying and retrieval,
  - and retrieval evaluation.

- Organized in **three milestones and workshops**:
  - M1: Data Preparation
  - M2: Information Retrieval
  - M3: Search System

Ad Hoc Search

- Focus is on the **ad hoc search task**
  - *standard retrieval task in which the user specifies his information need through a query which initiates a search (executed by the information system) for documents which are likely to be relevant to the user.*
- Examples (consider the collection, the data processing pipeline, the information needs)
  - Google search
  - Desktop search
  - Email search
- Other standard information retrieval tasks:
  - keyword extraction, summarization, question answering, information filtering …
  - not our focus.

Milestone #1: Data Preparation

- **Preparation and characterization** of the datasets selected for the project.
- This task is heavily dependent on the datasets, which may require some extraction actions such as crawling or scraping.
- Work on these tasks depends on the nature, volume, organization and accessibility of the selected datasets. As a result of this milestone, a well-documented and reproducible pipeline of data processing is expected.
- Groups must include the Makefile in the milestone submission.

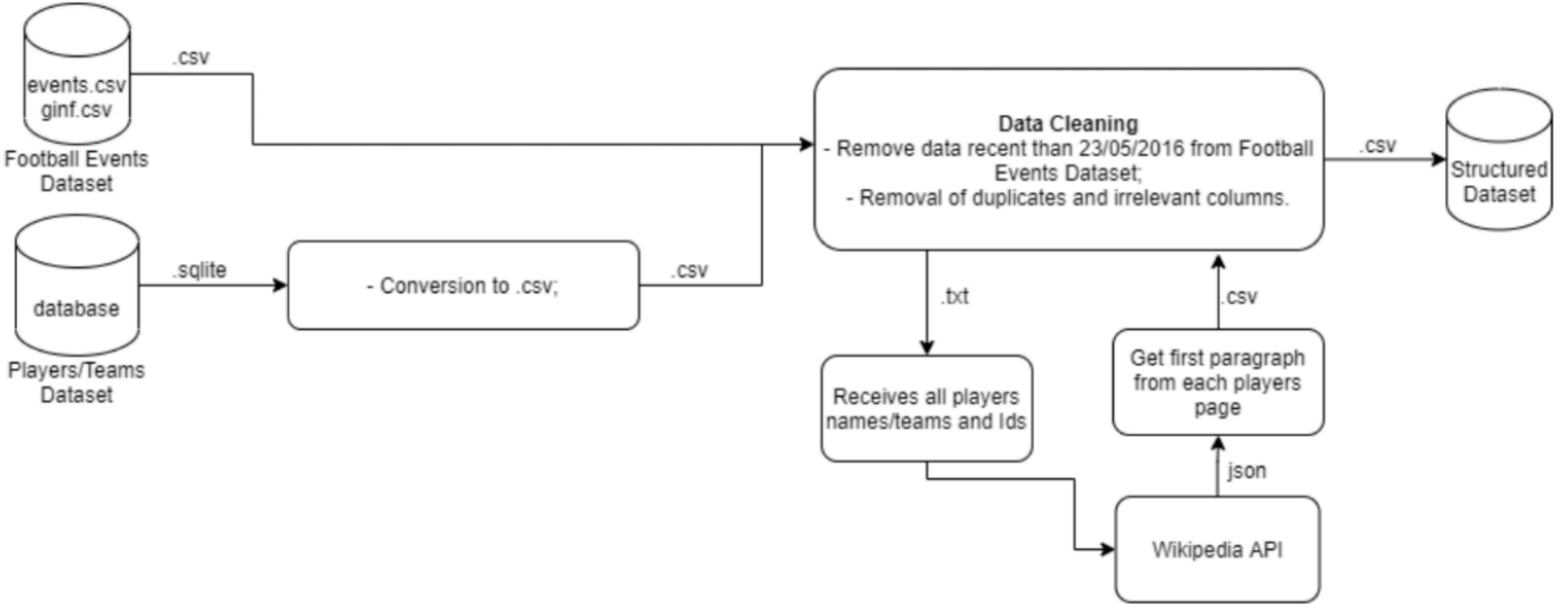
Milestone #1: Sample Actions

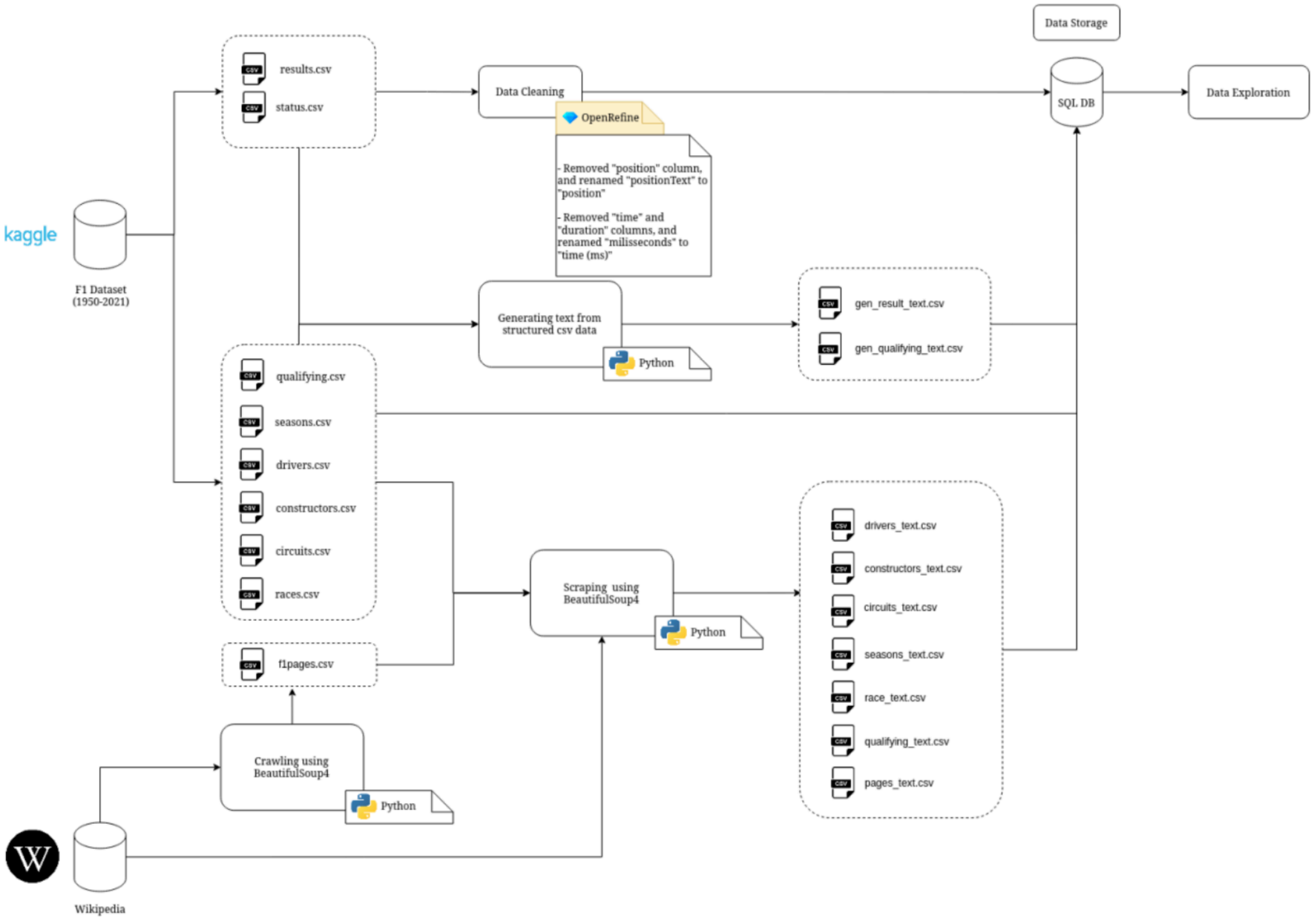
- The following list has a **sample of the actions which are required**:
  - search repositories for datasets;
  - select convenient data subsets;
  - assess the authority of the data source and data quality;
  - perform exploratory data analysis;
  - prepare and document a data processing pipeline;
  - characterize the datasets, identifying and describing some of their properties;
  - identify the conceptual model for the data domain;
  - **identify follow-up information needs** in the data domain.

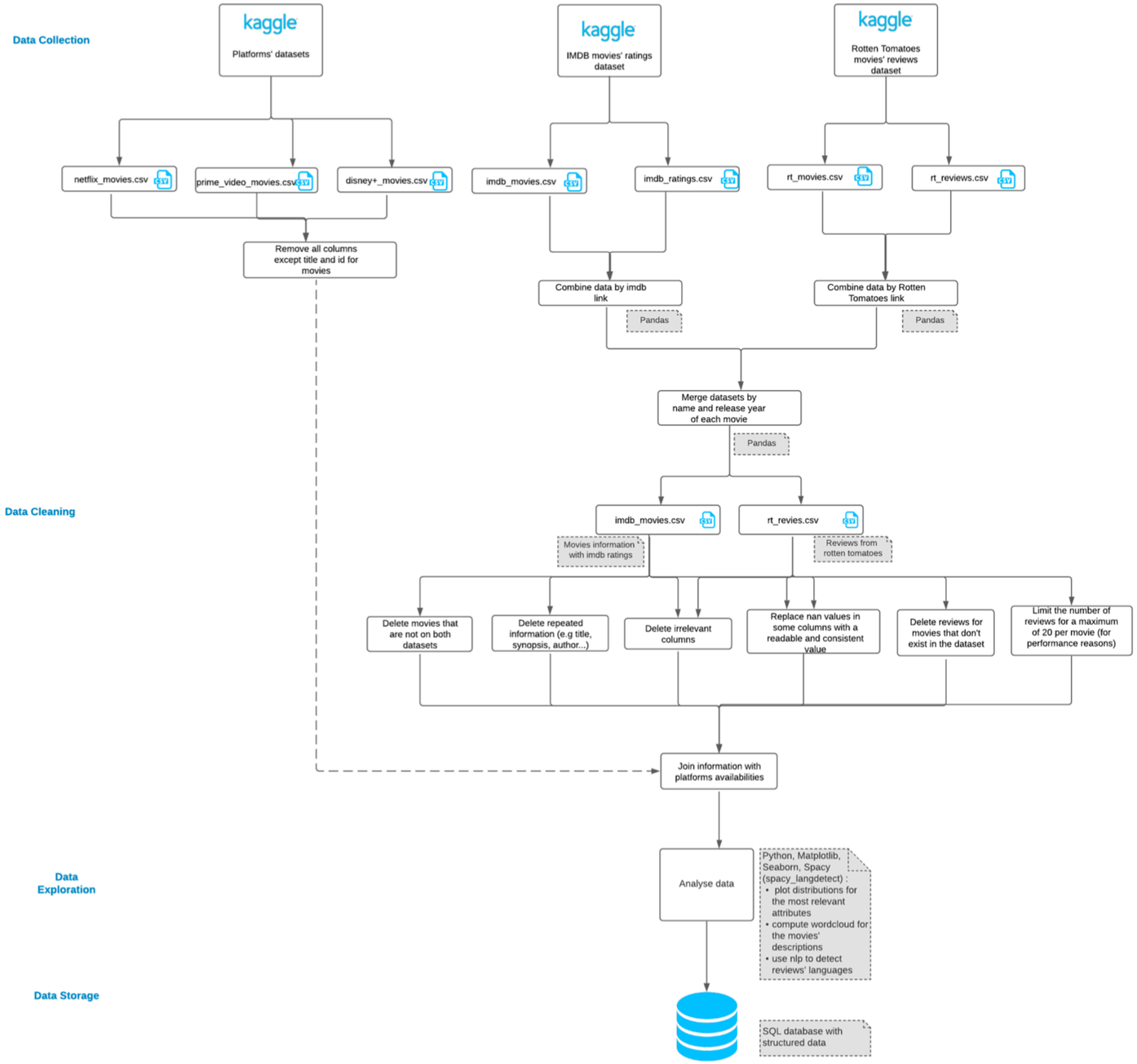
Examples

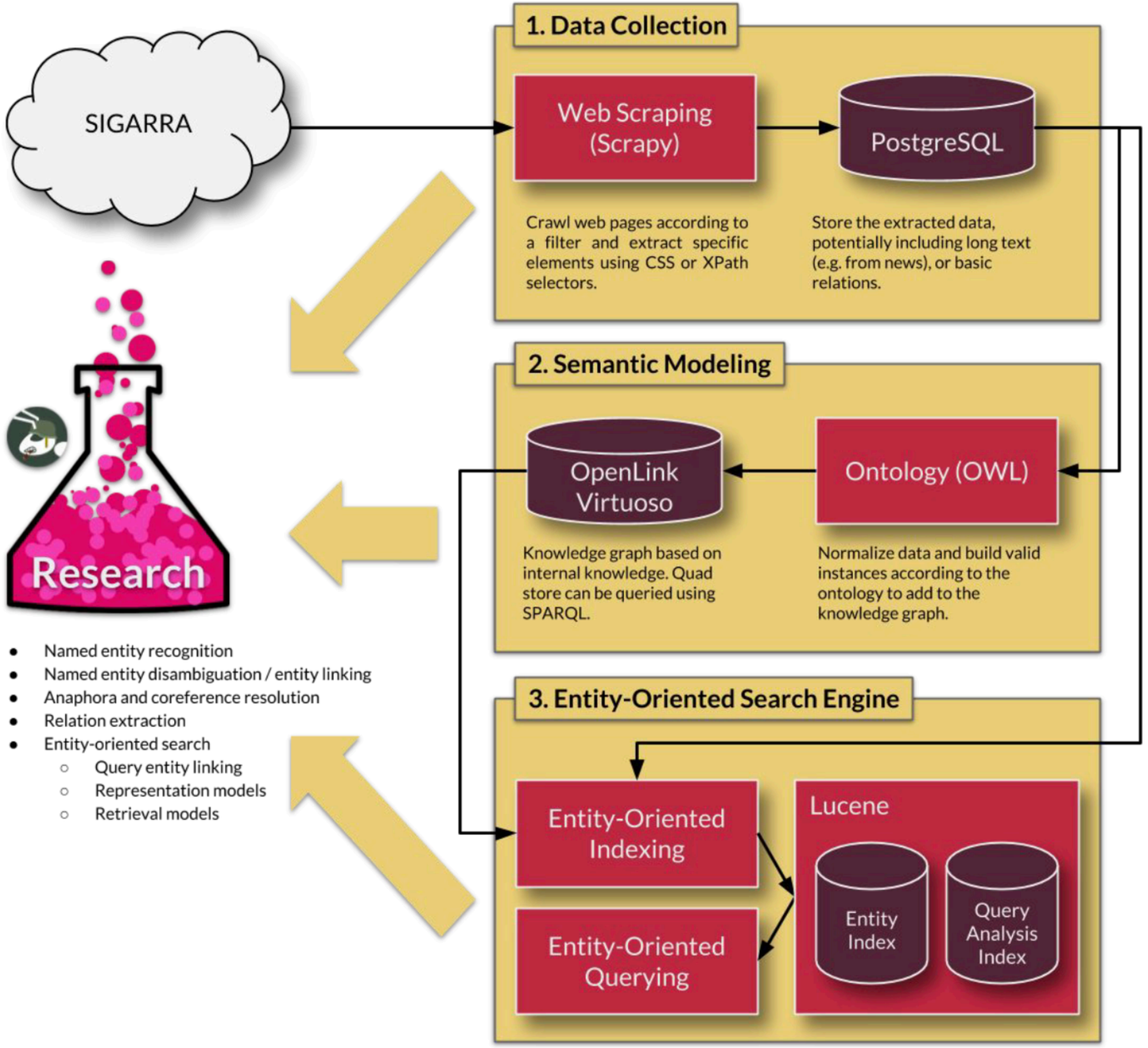
Topic Examples

- Search over MIEIC / M.EIC dissertations:
  - Data sources: U.Porto Open Repository
  - (Fallback: scientific articles)
  - Collection: web scraping
  - Processing: PDF parsing
- Search over political parties web pages throughout history:
  - Data source: arquivo.pt API
  - Collection: JSON API requests
  - Processing: parsing HTML


events.csv
ginf.csv
Football Events Dataset
.csv
database
Players/Teams Dataset
.sqlite
- Conversion to .csv;
.csv
Data Cleaning
- Remove data recent than 23/05/2016 from Football Events Dataset;
- Removal of duplicates and irrelevant columns.
.csv
Structured Dataset
.txt
Receives all players names/teams and Ids
Wikipedia API
json
Get first paragraph from each players page
.csv


Data Storage
results.csv
status.csv
Data Cleaning
OpenRefine
- Removed "position" column, and renamed "positionText" to "position"
- Removed "time" and "duration" columns, and renamed "milisseconds" to "time (ms)"
SQL DB
Data Exploration
kaggle
F1 Dataset (1950-2021)
Generating text from structured csv data
Python
gen_result_text.csv
gen_qualifying_text.csv
qualifying.csv
seasons.csv
drivers.csv
constructors.csv
circuits.csv
races.csv
Scraping using BeautifulSoup4
Python
drivers_text.csv
constructors_text.csv
circuits_text.csv
seasons_text.csv
race_text.csv
qualifying_text.csv
pages_text.csv
f1pages.csv
Crawling using BeautifulSoup4
Python
Wikipedia


Data Collection
kaggle
Platforms' datasets
kaggle
IMDB movies' ratings dataset
kaggle
Rotten Tomatoes movies' reviews dataset
netflix_movies.csv
prime_video_movies.csv
disney+_movies.csv
imdb_movies.csv
imdb_ratings.csv
rt_movies.csv
rt_reviews.csv
Remove all columns except title and id for movies
Combine data by imdb link
Pandas
Combine data by Rotten Tomatoes link
Pandas
Merge datasets by name and release year of each movie
Pandas
Data Cleaning
imdb_movies.csv
rt_revies.csv
Movies information with imdb ratings
Reviews from rotten tomatoes
Delete movies that are not on both datasets
Delete repeated information (e.g title, synopsis, author...)
Delete irrelevant columns
Replace nan values in some columns with a readable and consistent value
Delete reviews for movies that don't exist in the dataset
Limit the number of reviews for a maximum of 20 per movie (for performance reasons)
Join information with platforms availabilities
Data Exploration
Analyse data
Python, Matplotlib, Seaborn, Spacy (spacy_langdetect) :
• plot distributions for the most relevant attributes
• compute wordcloud for the movies' descriptions
• use nlp to detect reviews' languages
Data Storage
SQL database with structured data


SIGARRA
1. Data Collection
Web Scraping (Scrapy)
PostgreSQL
Crawl web pages according to a filter and extract specific elements using CSS or XPath selectors.
Store the extracted data, potentially including long text (e.g. from news), or basic relations.
2. Semantic Modeling
OpenLink Virtuoso
Ontology (OWL)
Knowledge graph based on internal knowledge. Quad store can be queried using SPARQL.
Normalize data and build valid instances according to the ontology to add to the knowledge graph.
3. Entity-Oriented Search Engine
Entity-Oriented Indexing
Entity-Oriented Querying
Lucene
Entity Index
Query Analysis Index
Research
Named entity recognition
Named entity disambiguation / entity linking
Anaphora and coreference resolution
Relation extraction
Entity-oriented search
Query entity linking
Representation models
Retrieval models

Lab #2: Class Plan

- Data collection and processing.
- Tools exploration and evaluation.
- Work on practical tutorials #1 (command line) and #2 (datasets).
- Continue work on project milestone 1.
- Obtaining datasets from the domain chosen for practical work.
- Datasets storage.
- Conduct exploratory data analysis.
- Conceptual domain modeling.

Data Preparation

Data Preparation

- Real-word data is "messy".
- In practice, 50% to 80% of the time spent in data processing is spent in data preparation tasks.
- Data preparation, often called "data wrangling", captures activities like:
  - Understand what data is available;
  - Choose what data to use and at what level of detail;
  - Understand how to combine multiple sources of data;
  - Deciding how to distill the results to a size and shape that enables the follow-up steps.

Data Preparation

- After data properties are investigated and understood, a data preparation phase is generally needed to make the data suitable for the follow-up phases.
- Common data preparation tasks include:
  - Data cleaning — identify and fix data quality issues;
  - Data transformation — transform data to improve analysis or manipulation;
  - Synthesis of Data — create new attributes derived from existing data;
  - Data integration — combine data from different sources;
  - Data reduction or selection — eliminate data from the collection.

Data Transformation

- Data transformation operations can be performed to improve or facilitate data handling in subsequent stages.
- Transformation of data elements include:
  - Normalization of values to a comparable scale;
  - Scaling values to the same range (e.g. 0 to 1 for a follow-up method);
  - Non-linear transformations to deal with skewed distributions;
  - Discretization or binning, which transforms a numeric continuous value into ordered categorical values (e.g. having bins of the same size vs. having bins with the same amount of items)
- Note that all operations over the data introduce layers of distortion and bias that are dependent on assumptions and interpretations. Documentation is key.

Discretization

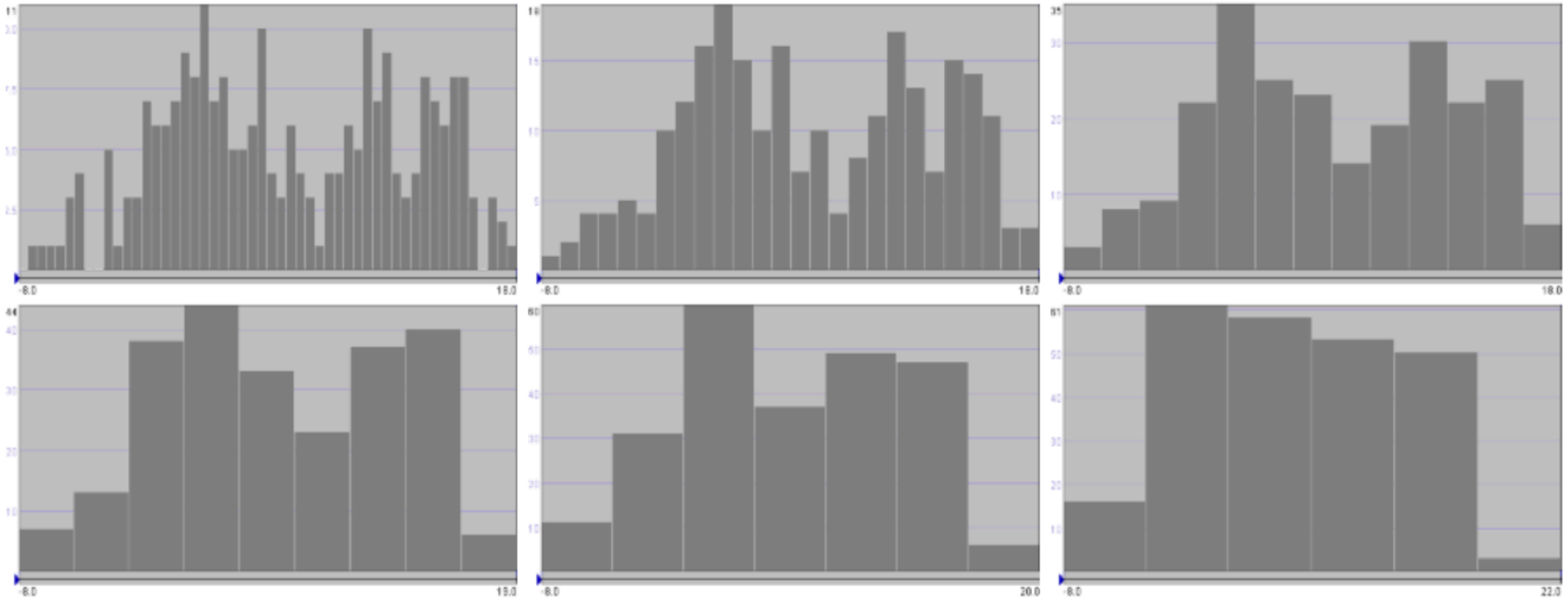


Fig. 3.13: Frequency histograms of the temperature data with the bin sizes equal to 0.5, 1, 2, 3, 4, and 5 degrees Celsius, correspondingly (left to right, top to bottom).

- Image from "Visual Analytics for Data Science", Springer (2020).

Synthesis of Data

- Combine existing attributes to produce new additional attributes that are more convenient to analyze or use as input in follow-up phases. Examples:
  - a new attribute representing the difference between two timestamps (e.g. duration);
  - the maximum value from a series of numerical attributes;
  - an integrated score that combines several attributes;
  - splitting an existing numerical series in two independent series (e.g. day and night);
  - most important keywords or topics extracted from a textual field;
  - etc.

Data Integration

- Combine data that originally exists in multiple sources.
- Key task in many projects, e.g. integrate data from multiple databases in an organization; enrich individual records with data from external sources.
- Complexity of data integration tasks differs significantly, from using join operations in a single database management system, to downloading, processing and linking external data sources.
- A central step of many data integration tasks is linking the corresponding records. A task that is easier if unique identifiers are available (rarely!), but of high complexity if based on other information (e.g. names, birth dates, addresses).

Data Reduction or Selection

- Data reduction or selection may be justified due to several reasons, namely:
  - data is not relevant;
  - data is outdated;
  - data volume exceeds the exiting capacity for processing it;
  - existing precision is excessive, while lower precision is sufficient.
- Techniques for performing data reduction or selection include:
  - data filtering
  - data sampling
  - data aggregation

Data Filtering

- Data filtering is used to remove data from the dataset, e.g. items with unsuitable values, data that is irrelevant for the scope of the project, outdated data items, data items that must be removed due to legal reasons, etc.
- Data filtering can also be used during development to test the planned approach using a manageable portion of the original collection.
- Data filtering operations are deterministic in nature, e.g. remove data from a given year, remove all references to a specific keyword, remove a specific data attribute, etc.

Data Sampling

- Data sampling is a non-deterministic process that takes a random subset of the data items of a requested size.
- When designing the sampling method it is important to ensure that the resulting sample is representative of the complete collection.
- Thus, it is important to analyze the distribution of data attributes before and after the sampling process.

Data Aggregation

- Data aggregation may be used to reduce excessive detail in data, decisions require:
  - choice of the grouping method, depending of domain-specific criteria;
  - selection of the aggregation operator (e.g. mean, median, min, max, percentile).

Visualization in Data Preparation

- Data preparation requires a good understanding of the data properties, thus data visualization is usually also used at this stage.
- The role of visualization at this stage is to visually present the result of the execution of different methods, e.g. outlier removal.
- Visualization in data preparation can be applied:
  - Before the application of computational methods to explore the properties of the data, e.g. choose an adequate computational method, detect disparities in data.
  - During the application of computational methods to inspect how data is being processes, e.g. observe intermediate data structures, track the data pipeline execution.
  - After the application of computational methods to evaluate the quality if the results, compare the results between different executions of the pipeline.

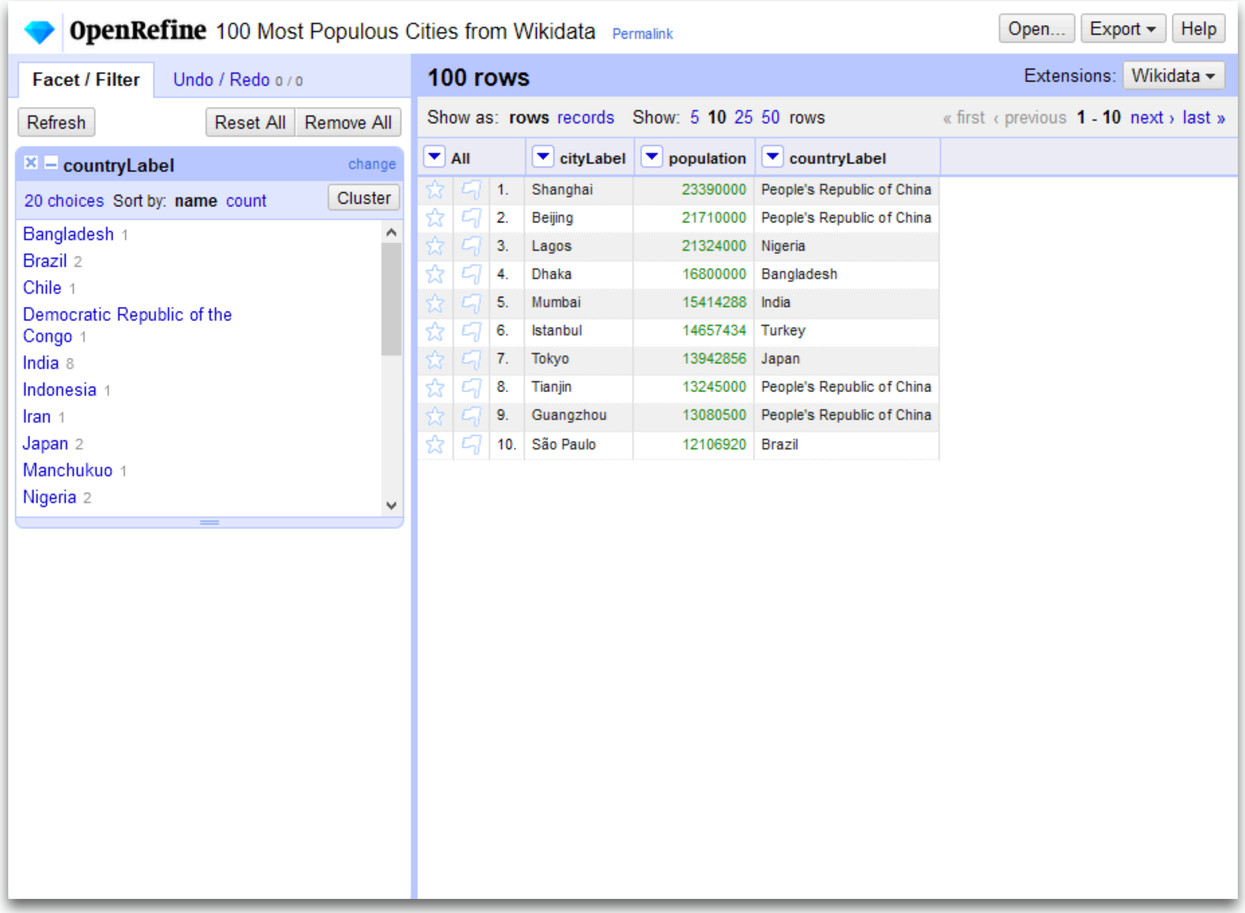
Tools for Data Preparation

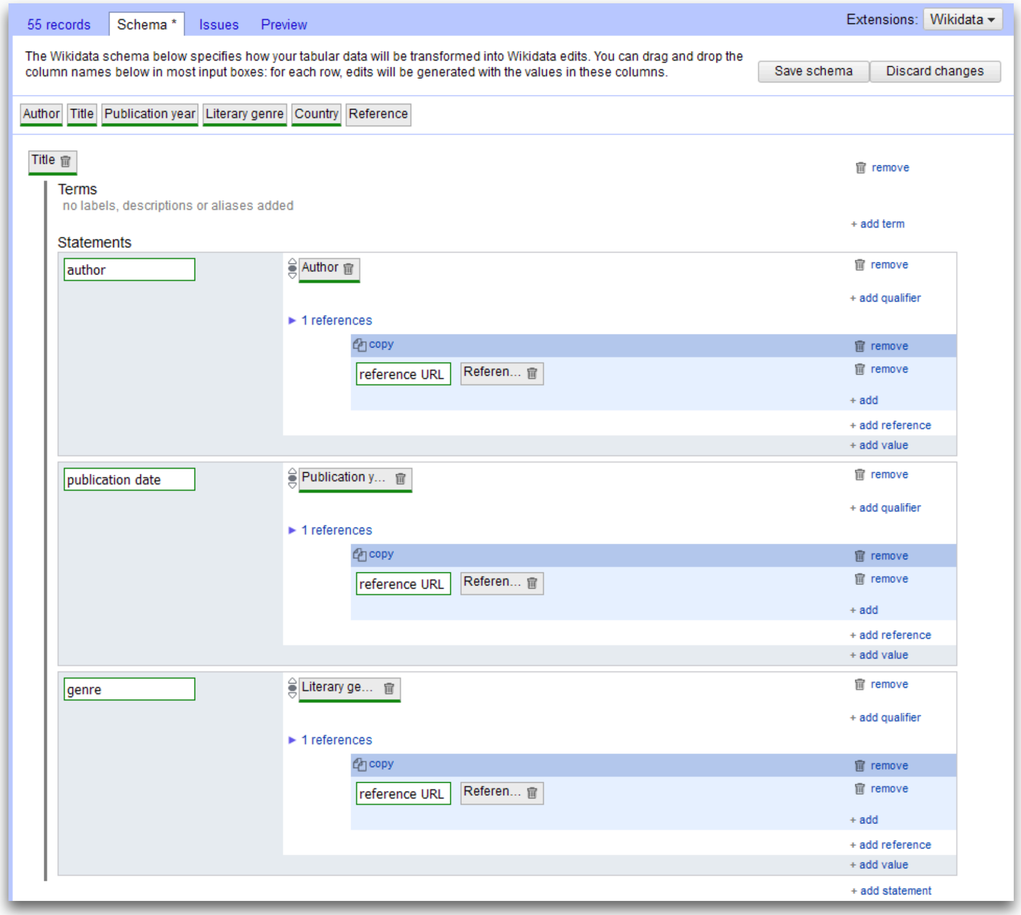
Tools for Data Cleaning

- Existing tools combine visual and computational techniques for identifying and fixing data problems.
- A freely available open-source reference software is OpenRefine (formerly Google Refine)
  - https://openrefine.org/
  - Focus on tabular data;
  - Supports data cleaning operations on numerical data, dates and time, and textual data;
  - Data exploration with descriptive statistics, facets, and histograms;
  - Data transformation with batch detection and replacement of inconsistent or missing values, time and date formatting, textual misspellings;
  - Integration with Wikidata for both obtaining and publishing data.

  - Explore available tutorials in lab classes.

OpenRefine


OpenRefine 100 Most Populous Cities from Wikidata
Permalink
Open...
Export
Help
Facet / Filter
Undo / Redo 0 / 0
100 rows
Extensions: Wikidata
Refresh
Reset All
Remove All
Show as: rows records
Show: 5 10 25 50 rows
« first ‹ previous 1 - 10 next › last »
countryLabel
change
20 choices Sort by: name count
Cluster
Bangladesh 1
Brazil 2
Chile 1
Democratic Republic of the Congo 1
India 8
Indonesia 1
Iran 1
Japan 2
Manchukuo 1
Nigeria 2
All
cityLabel
population
countryLabel
1. Shanghai 23390000 People's Republic of China
2. Beijing 21710000 People's Republic of China
3. Lagos 21324000 Nigeria
4. Dhaka 16800000 Bangladesh
5. Mumbai 15414288 India
6. Istanbul 14657434 Turkey
7. Tokyo 13942856 Japan
8. Tianjin 13245000 People's Republic of China
9. Guangzhou 13080500 People's Republic of China
10. São Paulo 12106920 Brazil


55 records
Schema *
Issues
Preview
Extensions: Wikidata
The Wikidata schema below specifies how your tabular data will be transformed into Wikidata edits. You can drag and drop the column names below in most input boxes: for each row, edits will be generated with the values in these columns.
Save schema
Discard changes
Author
Title
Publication year
Literary genre
Country
Reference
Title
remove
Terms
no labels, descriptions or aliases added
+ add term
Statements
author
Author
remove
+ add qualifier
1 references
copy
remove
reference URL
Referen...
remove
+ add
+ add reference
+ add value
publication date
Publication y...
remove
+ add qualifier
1 references
copy
remove
reference URL
Referen...
remove
+ add
+ add reference
+ add value
genre
Literary ge...
remove
+ add qualifier
1 references
copy
remove
reference URL
Referen...
remove
+ add
+ add reference
+ add value
+ add statement

Tools for Data Preparation

- Parse and extract text and metadata from files.
  - Apache Tika (e.g. PDF, PPT, XLS), https://tika.apache.org
  - Beautiful Soup (HTML), https://www.crummy.com/software/BeautifulSoup/
- NLP toolkits provide natural text processing tools
  - spaCy, http://spacy.io
  - NLTK, http://nltk.org
  - Apache OpenNLP, http://opennlp.apache.org
- Data processing
  - R, http://www.r-project.org
  - Pandas, https://pandas.python.org

Cloud Computing

- Cloud-based services are not a focus of this course but are a relevant piece in modern data processing and analytics pipelines.
- Managed services make building and deploying data pipelines more accessible, due to the elastic properties of the service, and the outsourcing of the setup and configuration part.
- Models based on pay-per-use, enable setting up and discarding data processing pipelines as needed.
- Cloud solutions are a good option to ad-hoc, highly diverse needs.

Data Pipelines

Data Pipelines

- Data pipelines are sets of processes that move and transform data from various sources to various destinations where new value can be derived.

- Complexity of data pipelines varies widely, depending on size, state, structure of the data sources as well as the needs of the project.
  - Simple example: obtain data from a REST API and load it to a relational database;
  - Complex example: obtain data from multiple web pages at regular intervals; parse HTML and extract main keyword from each page; automatically identity main trending topics using a previously trained machine learning model; update model with new data; integrate topics and web pages in a cloud-based storage linked to a live dashboard;

Characteristics of a Data Pipeline

- A data pipeline is a software system, thus general software best practices apply. Those highlighted here are particularly relevant in the context of data pipelines.
  - Reliable — work as expected even in face of adversity (i.e. software, hardware or human faults).
  - Scalable — cope with increased load (in volume, traffic, complexity) in a manageable way (e.g. adding resources, distributing load).
  - Maintainable — evolve through time and teams (reduce complexity, make changes easier).
- Data collection and preparation is usually an ad-hoc and exploratory process, easily leading to a dispersion in threads and activities. Adoption of pipeline management systems (e.g. Makefiles) and a complete and detailed documentation is key.
- Treating data processing pipelines as software makes them more maintainable, versionable, testable and collaborative.

Tools for Data Pipelines

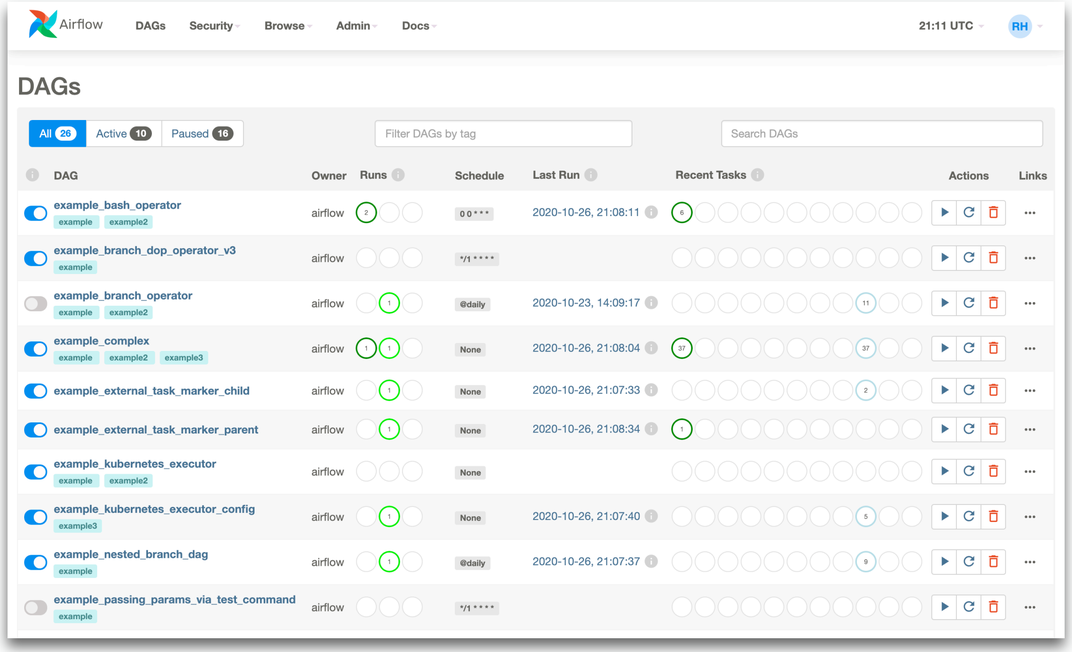
Makefiles

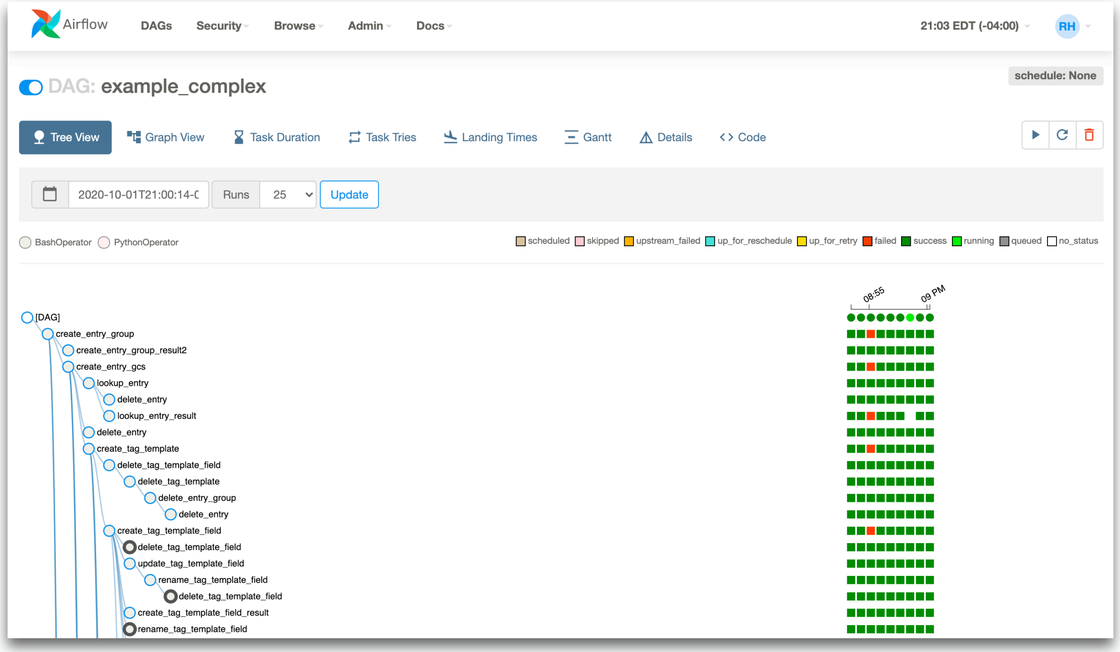
- Makefiles are used to automate software build processes, by defining targets and rules to execute. The underlying abstraction is of a dependency graph, where tasks depend on the execution of other tasks.
- Make is language agnostic and implementations exist for most operating systems.
- Can be used to document and setup data pipelines.

- Expected to be used in the final project.
- To be explored in a lab class.

Apache Airflow

- A Python-based, open-source platform, to "programmatically author, schedule and monitor workflows" [https://airflow.apache.org/]





Documentation of Data Pipelines

Data Documentation

- Documentation is central in any engineering process.
- It distinguishes between ad-hoc processes and repeatable, inspectable, shareable processes.
- It is essential to document:
  - Data formats across the pipeline (input, output, as well as intermediary files);
  - Platform and software versions (OS, tools);
  - Operations performed (extract, sample, link, etc);
- For understanding underlying assumptions (bias) in data documentation is a key element.

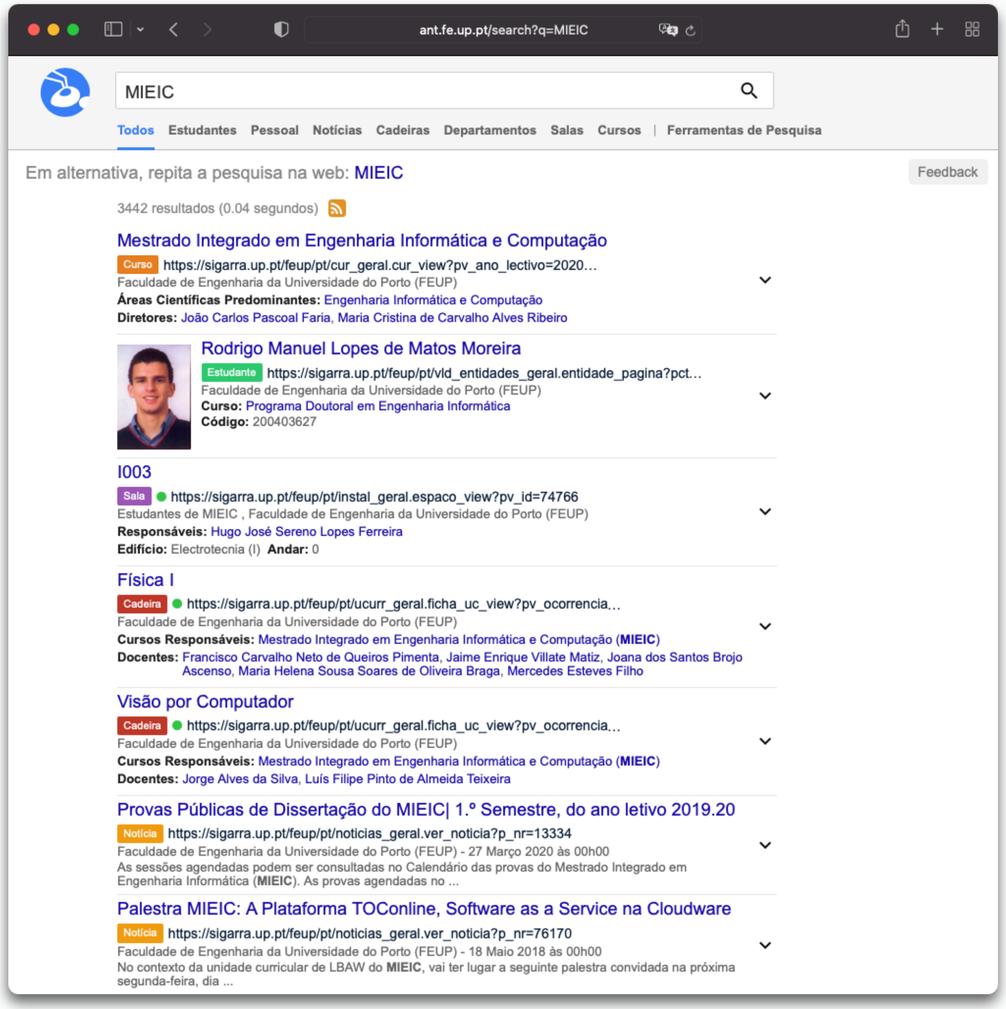
Data Flow Diagrams (DFD)

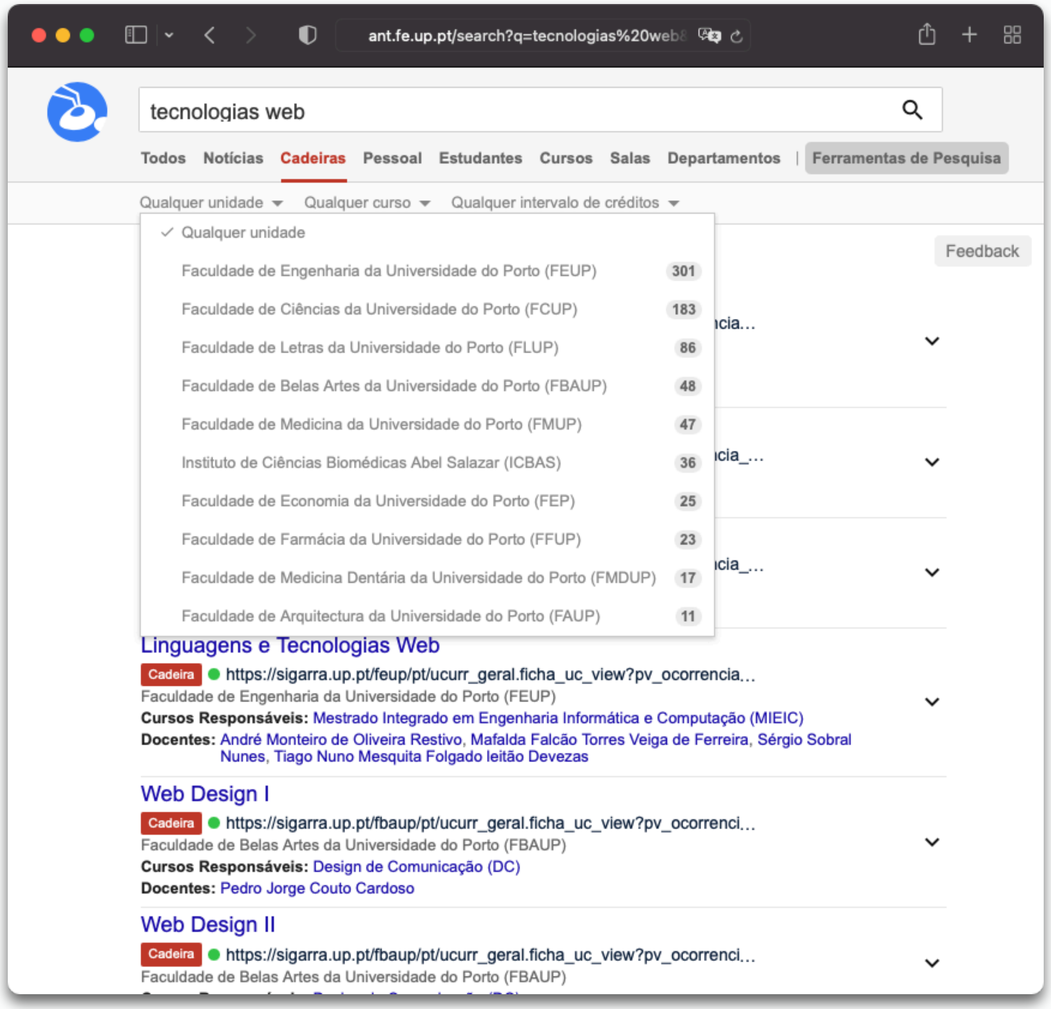
- Data-flow diagrams can be used to represent the flow of data from external entities into the system, show how data moves from one process to another, and data's logical storage.

- The notation includes four symbols:
  - **Squares** represent external entities, i.e. sources or destinations of data
  - **Rounded rectangles** represent processes, which takes data as input, perform operations over it, and then output it.
  - **Arrows** represent data flows, i.e. how data moves around.
  - **Open-ended rectangles** represent data stores, e.g. databases, files.

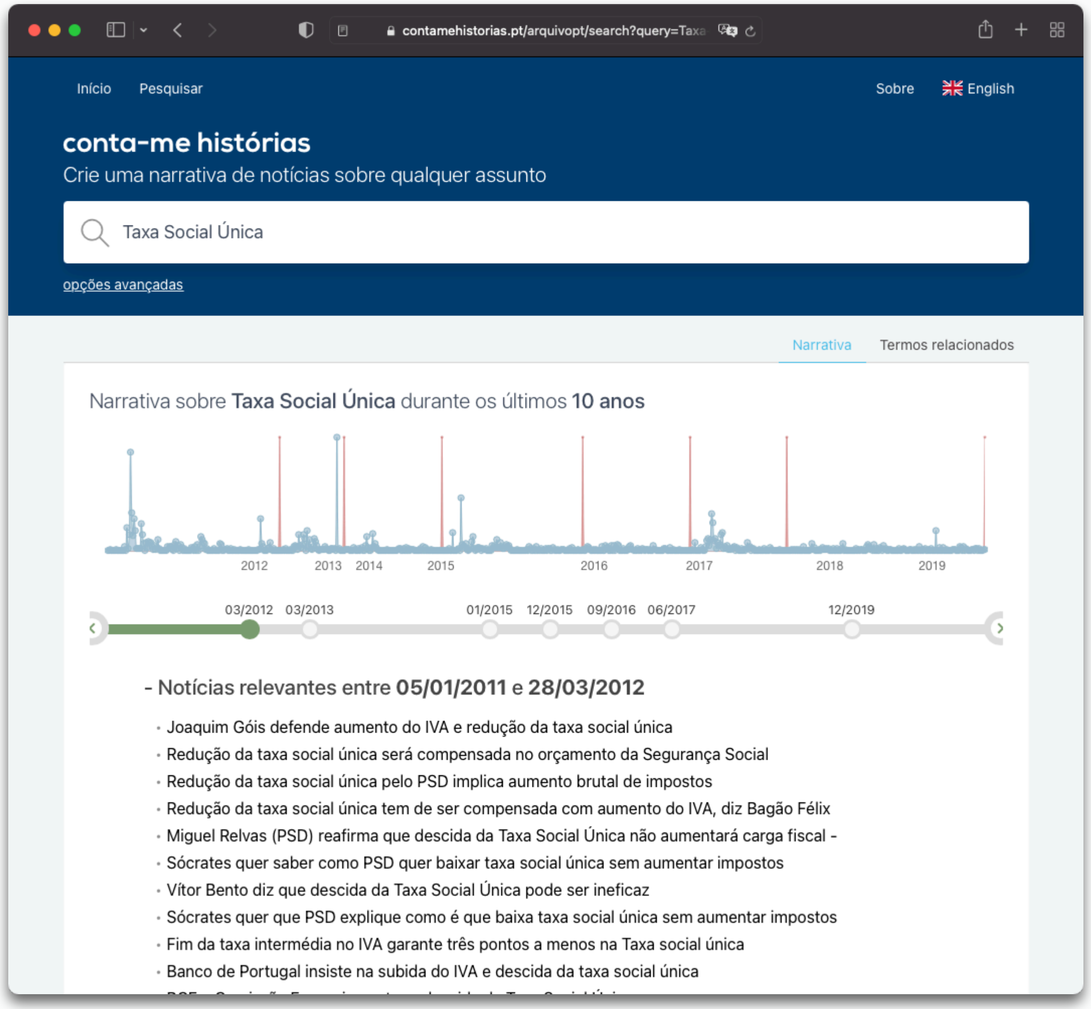
Example Projects

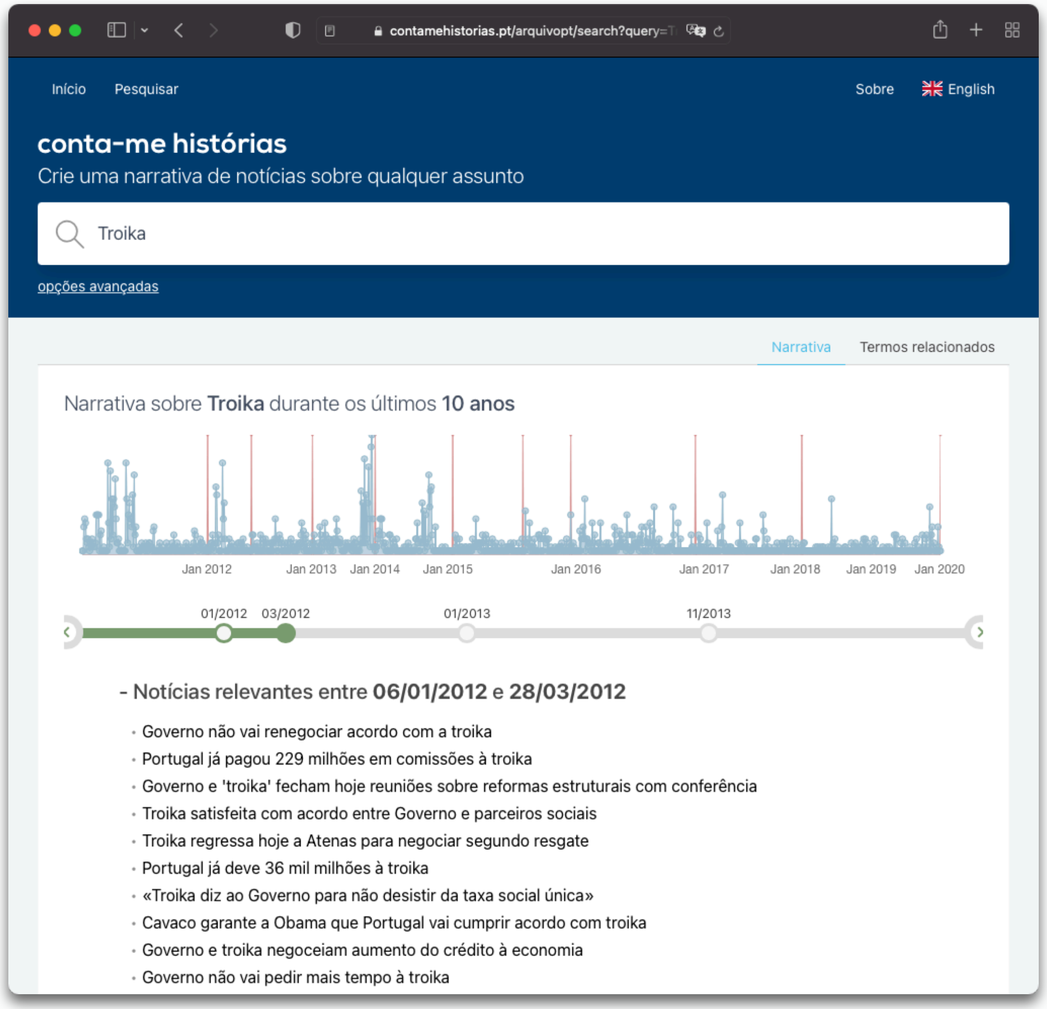
ANT

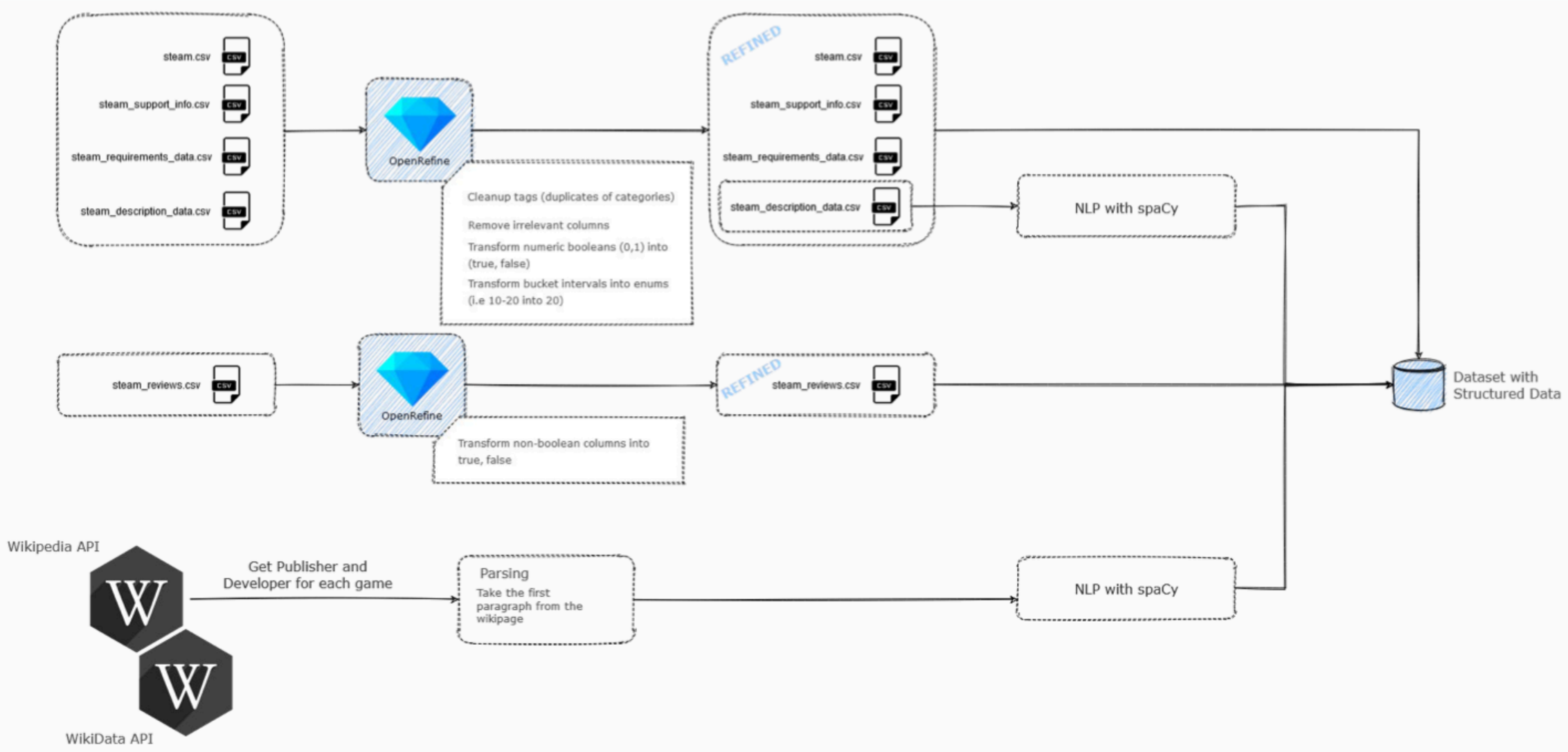

ant.fe.up.pt/search?q=MIEIC
MIEIC
Todos Estudantes Pessoal Notícias Cadeiras Departamentos Salas Cursos | Ferramentas de Pesquisa
Em alternativa, repita a pesquisa na web: MIEIC
Feedback
3442 resultados (0.04 segundos)
Mestrado Integrado em Engenharia Informática e Computação
Curso https://sigarra.up.pt/feup/pt/cur_geral.cur_view?pv_ano_lectivo=2020...
Faculdade de Engenharia da Universidade do Porto (FEUP)
Áreas Científicas Predominantes: Engenharia Informática e Computação
Diretores: João Carlos Pascoal Faria, Maria Cristina de Carvalho Alves Ribeiro
Rodrigo Manuel Lopes de Matos Moreira
Estudante https://sigarra.up.pt/feup/pt/vld_entidades_geral.entidade_pagina?pct...
Faculdade de Engenharia da Universidade do Porto (FEUP)
Curso: Programa Doutoral em Engenharia Informática
Código: 200403627
I003
Sala https://sigarra.up.pt/feup/pt/instal_geral.espaco_view?pv_id=74766
Estudantes de MIEIC , Faculdade de Engenharia da Universidade do Porto (FEUP)
Responsáveis: Hugo José Sereno Lopes Ferreira
Edifício: Electrotecnia (I) Andar: 0
Física I
Cadeira https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia...
Faculdade de Engenharia da Universidade do Porto (FEUP)
Cursos Responsáveis: Mestrado Integrado em Engenharia Informática e Computação (MIEIC)
Docentes: Francisco Carvalho Neto de Queiros Pimenta, Jaime Enrique Villate Matiz, Joana dos Santos Brojo Ascenso, Maria Helena Sousa Soares de Oliveira Braga, Mercedes Esteves Filho
Visão por Computador
Cadeira https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia...
Faculdade de Engenharia da Universidade do Porto (FEUP)
Cursos Responsáveis: Mestrado Integrado em Engenharia Informática e Computação (MIEIC)
Docentes: Jorge Alves da Silva, Luís Filipe Pinto de Almeida Teixeira
Provas Públicas de Dissertação do MIEIC| 1.º Semestre, do ano letivo 2019.20
Notícia https://sigarra.up.pt/feup/pt/noticias_geral.ver_noticia?p_nr=13334
Faculdade de Engenharia da Universidade do Porto (FEUP) - 27 Março 2020 às 00h00
As sessões agendadas podem ser consultadas no Calendário das provas do Mestrado Integrado em Engenharia Informática (MIEIC). As provas agendadas no ...
Palestra MIEIC: A Plataforma TOConline, Software as a Service na Cloudware
Notícia https://sigarra.up.pt/feup/pt/noticias_geral.ver_noticia?p_nr=76170
Faculdade de Engenharia da Universidade do Porto (FEUP) - 18 Maio 2018 às 00h00
No contexto da unidade curricular de LBAW do MIEIC, vai ter lugar a seguinte palestra convidada na próxima segunda-feira, dia ...

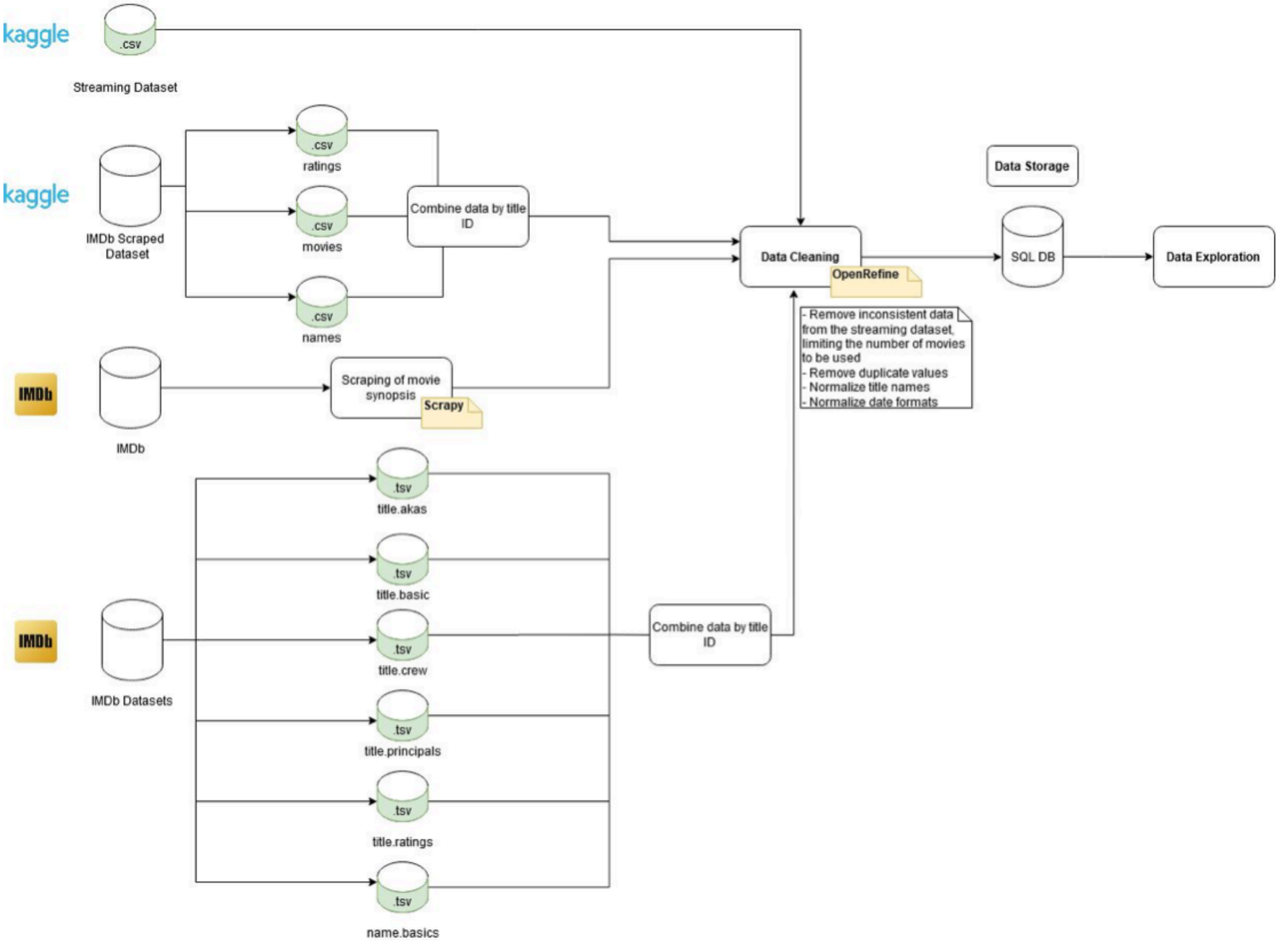

ant.fe.up.pt/search?q=tecnologias%20web
tecnologias web
Todos Notícias Cadeiras Pessoal Estudantes Cursos Salas Departamentos | Ferramentas de Pesquisa
Qualquer unidade Qualquer curso Qualquer intervalo de créditos
Qualquer unidade
Faculdade de Engenharia da Universidade do Porto (FEUP) 301
Faculdade de Ciências da Universidade do Porto (FCUP) 183
Faculdade de Letras da Universidade do Porto (FLUP) 86
Faculdade de Belas Artes da Universidade do Porto (FBAUP) 48
Faculdade de Medicina da Universidade do Porto (FMUP) 47
Instituto de Ciências Biomédicas Abel Salazar (ICBAS) 36
Faculdade de Economia da Universidade do Porto (FEP) 25
Faculdade de Farmácia da Universidade do Porto (FFUP) 23
Faculdade de Medicina Dentária da Universidade do Porto (FMDUP) 17
Faculdade de Arquitectura da Universidade do Porto (FAUP) 11
Feedback
Linguagens e Tecnologias Web
Cadeira https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia...
Faculdade de Engenharia da Universidade do Porto (FEUP)
Cursos Responsáveis: Mestrado Integrado em Engenharia Informática e Computação (MIEIC)
Docentes: André Monteiro de Oliveira Restivo, Mafalda Falcão Torres Veiga de Ferreira, Sérgio Sobral Nunes, Tiago Nuno Mesquita Folgado leitão Devezas
Web Design I
Cadeira https://sigarra.up.pt/fbaup/pt/ucurr_geral.ficha_uc_view?pv_ocorrenci...
Faculdade de Belas Artes da Universidade do Porto (FBAUP)
Cursos Responsáveis: Design de Comunicação (DC)
Docentes: Pedro Jorge Couto Cardoso
Web Design II
Cadeira https://sigarra.up.pt/fbaup/pt/ucurr_geral.ficha_uc_view?pv_ocorrenci...
Faculdade de Belas Artes da Universidade do Porto (FBAUP)

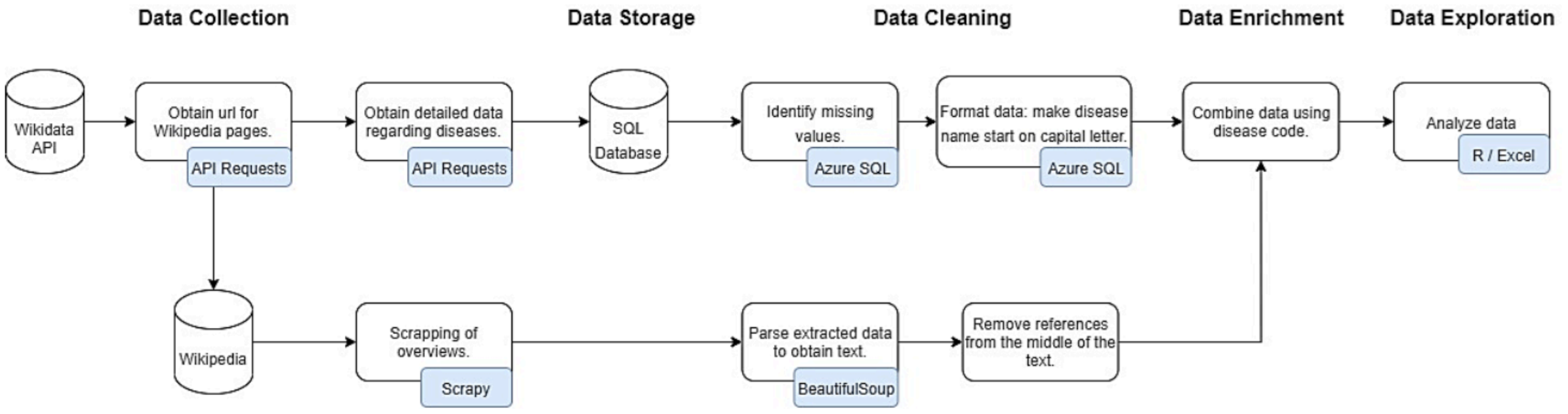
Conta-me Histórias






steam.csv
steam_support_info.csv
steam_requirements_data.csv
steam_description_data.csv
OpenRefine
Cleanup tags (duplicates of categories)
Remove irrelevant columns
Transform numeric booleans (0,1) into (true, false)
Transform bucket intervals into enums (i.e 10-20 into 20)
REFINED
steam.csv
steam_support_info.csv
steam_requirements_data.csv
steam_description_data.csv
NLP with spaCy
steam_reviews.csv
OpenRefine
Transform non-boolean columns into true, false
REFINED
steam_reviews.csv
Dataset with Structured Data
Wikipedia API
Get Publisher and Developer for each game
Parsing
Take the first paragraph from the wikipage
NLP with spaCy
WikiData API


kaggle
.csv
Streaming Dataset
kaggle
IMDb Scraped Dataset
.csv
ratings
.csv
movies
.csv
names
Combine data by title ID
IMDb
IMDb
Scraping of movie synopsis
Scrapy
IMDb
IMDb Datasets
.tsv
title.akas
.tsv
title.basic
.tsv
title.crew
.tsv
title.principals
.tsv
title.ratings
.tsv
name.basics
Combine data by title ID
Data Cleaning
OpenRefine
- Remove inconsistent data from the streaming dataset, limiting the number of movies to be used
- Remove duplicate values
- Normalize title names
- Normalize date formats
Data Storage
SQL DB
Data Exploration


Data Collection
Data Storage
Data Cleaning
Data Enrichment
Data Exploration
Wikidata API
Obtain url for Wikipedia pages.
API Requests
Obtain detailed data regarding diseases.
API Requests
SQL Database
Identify missing values.
Azure SQL
Format data: make disease name start on capital letter.
Azure SQL
Combine data using disease code.
Analyze data
R / Excel
Wikipedia
Scrapping of overviews.
Scrapy
Parse extracted data to obtain text.
BeautifulSoup
Remove references from the middle of the text.

Bibliography and Further Reading

- **Visual Analytics for Data Scientists** [VADS20]
  Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C., and Wrobel, S. Springer, 2020

- **Designing Data-Intensive Applications** [DDIA17]
  Kleppmann, M. O'Reilly, 2017

- **Principles of Data Wrangling** [PDW17]
  Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., and Carreras, C. O'Reilly, 2017

- **Data Pipelines Pocket Reference** [DPPR21]
  Densmore, J. O'Reilly, 2021

- **Information - A Very Short Introduction** [IVSI10]
  Floridi, L. Oxford University Press, 2010

Questions or comments?

Data Processing

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Today's Plan

- Focus on data processing
  - Practical perspective
  - Storage solutions
  - UNIX philosophy
  - Makefiles for data pipelines
- Data presentation

- Review "Milestone 1 - Data Preparation"
- Next lab class

Data Processing

Data Storage

Data Models

- Data models have a role part in software development, not only how software is implemented but also on how we think about the problem.
- Defining abstraction layers is a common pattern in software architectures, where each layer hides the complexity of the layers below.
- When thinking in data models, each layer hides the data complexity of previous layers, and provides a cleaner data model.
- Different data models make different operations faster/slower, natural/awkward, possible/impossible.
- Models discussed: relational model, document model, and graph-based models.

Relational Model

- The relational model is the best-known and most dominant data model.
- In the relational model data is organized in *relations* (called *tables* in SQL), where each relation is an unordered collection of *tuples* (*rows* in SQL).
- The NoSQL (*Not Only SQL*) movement emerged in the 2010s as a need to provide greater scalability, specialized query operations, and a more flexible and expressive data model.
- The "*object-relational mismatch*" (or *impedance mismatch*) is an expression used to represent a common criticism to the SQL data model — the need of a translation layer between "tables, rows, and columns" and the "objects" used in object-oriented programming.

Document Model

- In the document models data is organized as single instances (documents).
- In document databases, nested records are stored together with their parent records.
- Also known as document stores, document-oriented databases.
- Advantages of document data models are:
  - schema flexibility;
  - better performance due to locality (i.e. related data is stored together);
  - for some applications, the data model is closer to the application's data structures.
- The relational model is better suited to support joins, N-1, and N-N relationships.

Relational vs Document Representations

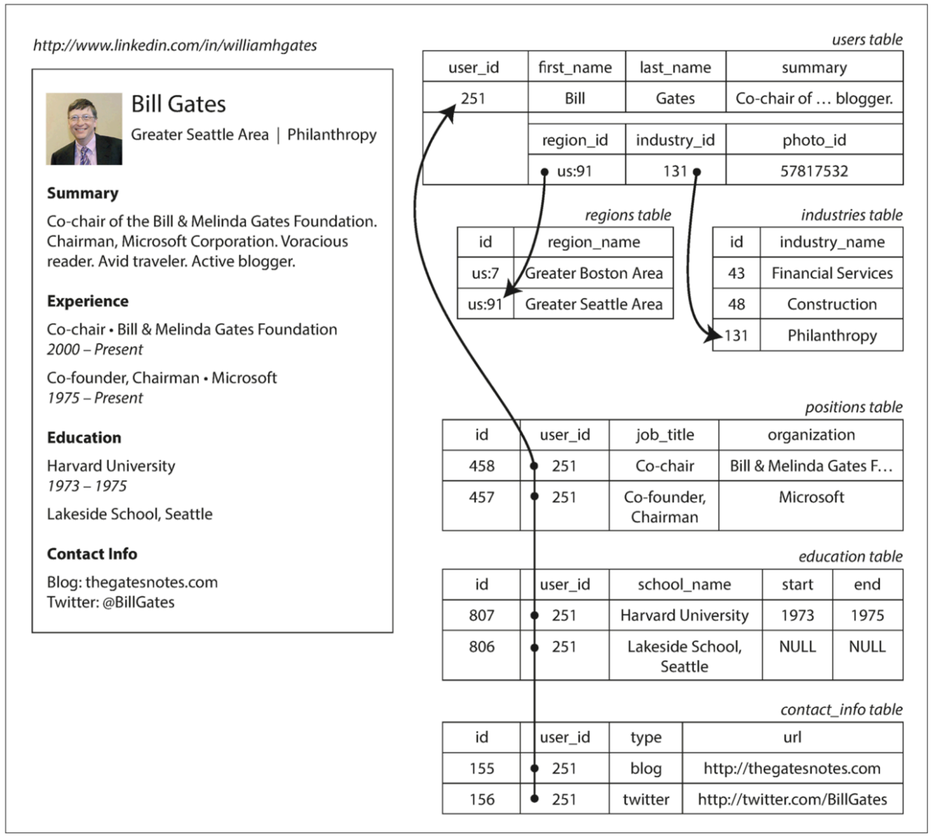


Figure 2-1. Representing a LinkedIn profile using a relational schema. Photo of Bill Gates courtesy of Wikimedia Commons, Ricardo Stuckert, Agência Brasil.

Example 2-1. Representing a LinkedIn profile as a JSON document

```
{
  "user_id":     251,
  "first_name":  "Bill",
  "last_name":   "Gates",
  "summary":     "Co-chair of the Bill & Melinda Gates... Active blogger.",
  "region_id":   "us:91",
  "industry_id": 131,
  "photo_url":   "/p/7/000/253/05b/308dd6e.jpg",

  "positions": [
    {"job_title": "Co-chair", "organization": "Bill & Melinda Gates Foundation"},
    {"job_title": "Co-founder, Chairman", "organization": "Microsoft"}
  ],
  "education": [
    {"school_name": "Harvard University",       "start": 1973, "end": 1975},
    {"school_name": "Lakeside School, Seattle", "start": null, "end": null}
  ],
  "contact_info": {
    "blog":    "http://thegatesnotes.com",
    "twitter": "http://twitter.com/BillGates"
  }
}
```

Materials from Kleppmann, M. Designing Data-Intensive Applications (2017).

Schema Flexibility

- Most document databases do not enforce any scheme on the data in documents.
- No schema (*schemaless*) means that arbitrary keys and values van be added to a document and, when reading, clients have no guarantees as to what fields to expect.
- Although schemaless databases do not impose a schema, code that reads the data usually assumes some kind of structure, thus there is an implicit schema. But not enforced by the database!
- This solution is advantageous when the items in a collection don't have the same structure, or the structure of data is determined by external systems which may change over time.
- Where the schema is enforced is the key issue — schema-on-read (the structure is enforced when data is read, at code level); schema-on-write (the structure is enforced by the database, the traditional relational approach).

Property Graph Model

- When the application has mostly one-to-many relationships (i.e. tree-like structures) or no relationships between records, the document model is appropriate.
- If many-to-many relationships are very common, and the connections within the data are complex, it is more natural to model data as a graph.
- A graph has to kinds of objects: vertices (also known as nodes or entities) and edges (also known as relationships or arcs).
- Graphs are not limited to homogeneous data, an advantage of the graph model is the fact that it provides a consistent way of storing completely different types of data in a single datastore (e.g. people, places, events).
- Typical examples, include: social graphs, web graph, transportation networks.

Graph Example

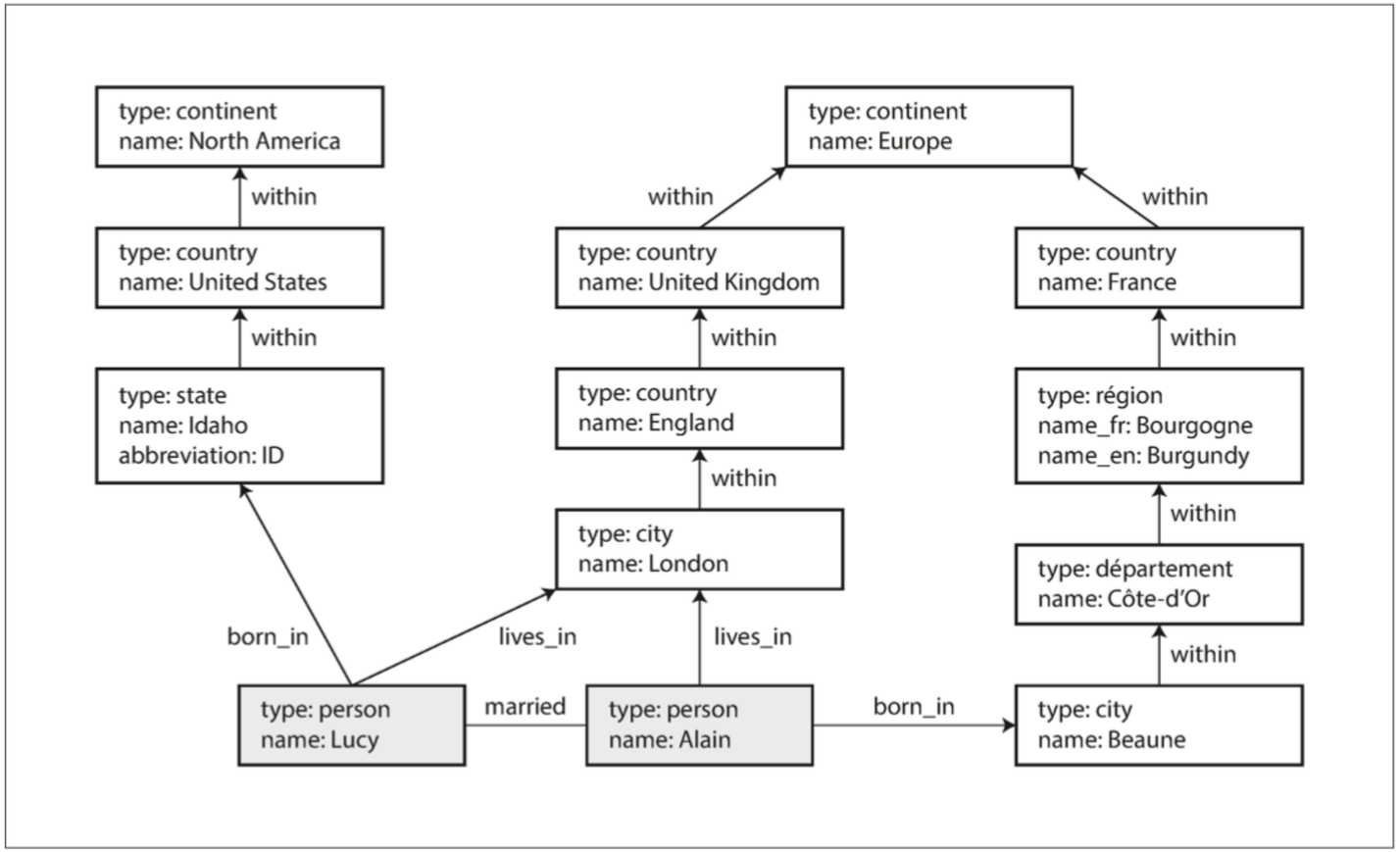


Figure 2-5. Example of graph-structured data (boxes represent vertices, arrows represent edges).

Image from Kleppmann, M. Designing Data-Intensive Applications (2017).

Triple-Store Model

- Similar to the property graph model.
- Important because there are various relevant tools and languages for triple-stores.
- In a triple-store, all information is stored in the form of three-part statements: *(subject, predicate, object)*. For example: *(Maria, enrolled, PRI)*; *(PRI, course, M.EIC)*.
- The subject of a triple is equivalent to a vertex in a graph.
- The object is either, a value (primitive datatype) or another vertex in the graph.
- RDF is a data model for specifying data subject-predicate-object (triples) statements.
- SPARQL is the querying language for triple-stores using the RDF data model.

Data Storage Solutions

- Industry-grade open-source solutions exist for the main paradigms.
- Relational model
  - SQLite, www.sqlite.org
  - PostgreSQL, www.postgresql.org
  - MariaDB, mariadb.org
- Document model
  - MongoDB, www.mongodb.com
  - CouchDB, couchdb.apache.org
- Graph model
  - Neo4j, neo4j.com

Data Storage Summary

- Data models are central in data processing.
- The relational data model is the most widely used approach.
- Non-relational solutions have emerged in recent years under the "NoSQL" umbrella.
- Two main alternative directions — document databases and graph databases.
  - With document databases, data is mapped to self-contained documents and relationships between documents are rare.
  - With graph databases, data is separated in independent units and, potentially, everything is related to everything.
- These three models are widely used and each is good in its respective domain.
- There is no one-size-fits-all solution, that's why different systems exist. Also, database systems support features from "competing" models, e.g. JSON support in relational database systems.

Batch Processing

Interaction Types

- There are three main approaches in building the interaction style in data processing.
- Online systems (services), where a service waits for requests or instructions from a client to arrive. When a request is received, the service tries to handle as quickly as possible and send the response back. Response time is the primary measure of performance.
- Offline systems (batch processing), where a batch processing system takes a large amount of input data, runs a job to process it, and produces some output data. Suitable for long jobs or asynchronous processes.
- Stream processing systems, where a service operates on inputs and produces outputs (rather than responses) as a result of an event happening (rather than periodically at scheduled times).

Batch Processing

- Simple log analysis with Unix tools.
  - ```
    cat /var/log/nginx/access.log | # read the log file
    awk '{print $7}' | # split each line into fields and take the 7th (URL)
    sort | # alphabetically sort each URL
    uniq -c | # filter duplicate URLs and count the number of duplicates
    sort -r -n | # numerically sort based on the number of occurrences
    head -n 5 # print the top 5 lines
    ```
- Powerful and versatile solution that can quickly process gigabytes of log files.
- An alternative approach would be to build a custom program to read each line into a hash table and keep track of each URL occurrence.
- The main difference between the two approaches is the fact that the custom program loads all data into memory, while the Unix approach relies on sorting as a key step in preparing data for counting.
    ```

The Unix Philosophy

- The *Unix Philosophy* encapsulates a set of design principles that became popular among the developers and users of Unix (1960/70s).
  - 1. Make each program do one thing well. To do a new job, build afresh rather than complicate old programs by adding new features.
  - 2. Expect the output of every program to become the input to another, as yet unknown, program. Don’t clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don’t insist on interactive input.
  - 3. Design and build software, even operating systems, to be tried early, ideally within weeks. Don’t hesitate to throw away the clumsy parts and rebuild them.
  - 4. Use tools in preference to unskilled help to lighten a programming task, even if you have to detour to build the tools and expect to throw some of them out after you’ve finished using them.
- Many of this concepts are relevant and are important when designing systems today.

Unix Uniform Interface

- In Unix, everything is a file, i.e. an ordered sequence of bytes.
- Actual filesystem files, TCP connections, device driver, etc., all share the same uniform interface, and thus can be plugged together.
- This makes it easy to build systems that "pipe" together many different blocks.
- Additionally, in Unix there is a decoupling of logic and wiring, in the send that the user can wire up the input and output in whatever way they want. This make it easier to compose small tools into bigger systems.
- One important limitation is that Unix tools run only on a single machine — tools like Hadoop have appeared to overcome this limitation.

Makefiles

Managing Data Pipelines

- A data processing project is typically a fragmented process, involving multiple operations, data storages, files, folders, formats, etc.
- Managing these processes and workflows is a requirement. The goal is to produce "software artifacts" that are maintainable, versionable, testable, and collaborative.
- Solutions include Apache Airflow, Make, Nextflow.
  - https://airflow.apache.org/
  - https://www.gnu.org/software/make/
  - https://www.nextflow.io/
- Make is used in PRI projects.

Make

- Make is a software tool that controls the execution of commands to derive *target files*, based on the existence (or change) of other files (*dependencies*) and the execution of operations (*recipes*).
- Make was originally built, and is still a central piece, in software building processes.
- It can generally be used in scenarios where the execution workflow can be encapsulated as a set of dependencies and execution rules.
- Make is able to automatically decide what to execute based on direct or indirect dependencies to files that have changed.
- Make is language agnostic and follows a declarative programming paradigm.
- Use cases: software building, scientific publishing (figures, tables, LaTeX files, etc), data pipelines.

Makefiles

- Makefile is a text file where Make execution rules and dependencies are defined.
- The default filename is "Makefile", with no extension and starting with an uppercase.
- A makefile consists of a series of rules.
  - Each rule defines a target name (or list) and an optional list of prerequisites.
  - Additionally, a rule can have a list of commands to execute (recipe).
  - Recipe commands must be indented with TABs.
- To build a specific target, "make <target>" is used.
- Manual online: https://www.gnu.org/software/make/manual/make.html

Example Makefile

```
# Makefile

# Vars
URL = "https://en.wikipedia.org/wiki/Alice%27s_Adventures_in_Wonderland"

# Rule 'all' defines how to build everything
all: alice.txt

# Rule to obtain "alice.html"
# The existence of specific tools can be checked on the dependencies (e.g. curl)
alice.html: /usr/bin/curl
  curl -s $(URL) > alice.html

# Rule to generate "alice.txt" from "alice.html"
alice.txt: alice.html
  lynx -dump alice.html > alice.txt

# Rule to remove generated files
clean:
  rm -f alice.txt alice.html
```

```
$ make alice.html # To execute a specific rule

$ make # To build all. If alice.txt already exists, it won't build it again
$ make -s # Silent execution

$ make clean # To clean up
```

Makefiles Highlights

- Makefiles support control structures such as if conditions, execution cycles, etc.
  - https://www.gnu.org/software/make/manual/html_node/Conditionals.html
  - https://www.gnu.org/software/make/manual/html_node/Foreach-Function.html
- Make can build targets in parallel. Use the -j flag to take advantage of multiple cores.
- Automatic variables provide access to values computed for the rule in execution:
  - Examples: $@ (first target); $^ (all prerequisites);
  - https://www.gnu.org/software/make/manual/html_node/Automatic-Variables.html
- It is possible to store the execution of a shell command in a variable
  - PDF_FILES = $(shell find . -type f -name '*.pdf')

Makefiles as Macros

- You can use targets to simply encapsulate multiple commands, e.g.
  - 'make requirements' to setup your environment
  - 'make publish' to upload to a remote server
  - 'make web' to start a local web server and server your project
  - 'make figures' to prepare figures from your data collection
  - ...
- Rules that should always be executed are marked as a 'phony target', with:
  - .PHONY requirements publish

Makefiles - Additional References

- **Makefile Tutorial**, 2021
  https://makefiletutorial.com/
- **Chapter 6 Project Management with Make**, from Data Science at the Command Line, Janssens, J., O'Reilly, 2021
  https://www.datascienceatthecommandline.com/2e/chapter-6-project-management-with-make.html
- **Managing Project with GNU Make**, Robert Mecklenburg. O'Reilly Open Book, 2004
  https://www.oreilly.com/openbook/make3/book/index.csp
- **Why Use Make**, Mike Bostock, 2013
  https://bost.ocks.org/mike/make/
- **Reproducibility with Make**, from The Turing Way Community, Arnold, B. et al., 2019
  https://the-turing-way.netlify.app/reproducible-research/make.html
- **Chapter 36 Automating data-analysis pipelines**, from STAT 545 course, Jenny Bryan
  https://stat545.com/automating-pipeline.html

Data Presentation

Data Presentation

- Data presentation is an important task in data projects.
- Depending on the audience, presentation can differ greatly.
- Presentation can simply use tables to effectively communicate the key aspects.
- Or use more or less complex visualizations, static or interactive.
- We saw that data visualization can be an important tool in data preparation, e.g. to investigate data or evaluate data quality issues. In this case, the audience is mostly specialized.
- For other end-users, presentation solutions will depend, and can range from static images (e.g. in internal documentation), to publicly available websites (e.g. intranet), to interactive applications (e.g. data journalism).

Data Visualization

- Data visualization (*data viz*) refer to algorithmically drawn, easy to regenerate, and aesthetically simples visualizations. As opposed to infographics, which are manually drawn and aesthetically more complex visualizations.
- Data visualizations can be classified according to their complexity considering the number of dimensions they represent, i.e. the number of discrete data types visually encoded in the diagram.
- Visualization can also be divided in two categories, each with a different purpose: exploration and explanation.
  - Exploration is generally done at a higher level of granularity, earlier in the process, and the main goal is to understand what is in the data;
  - Explanation is appropriate when you already know the data and you want to support and present findings.
- In the context of PRI,
  - we are interest in simple (but insightful), static visualization;
  - that help us understand and highlight relevant characteristics of the data;
  - and are programmatically obtained, i.e. use scripts to quickly generate the visualizations.

Napoleon's Russian Campaign (1869)

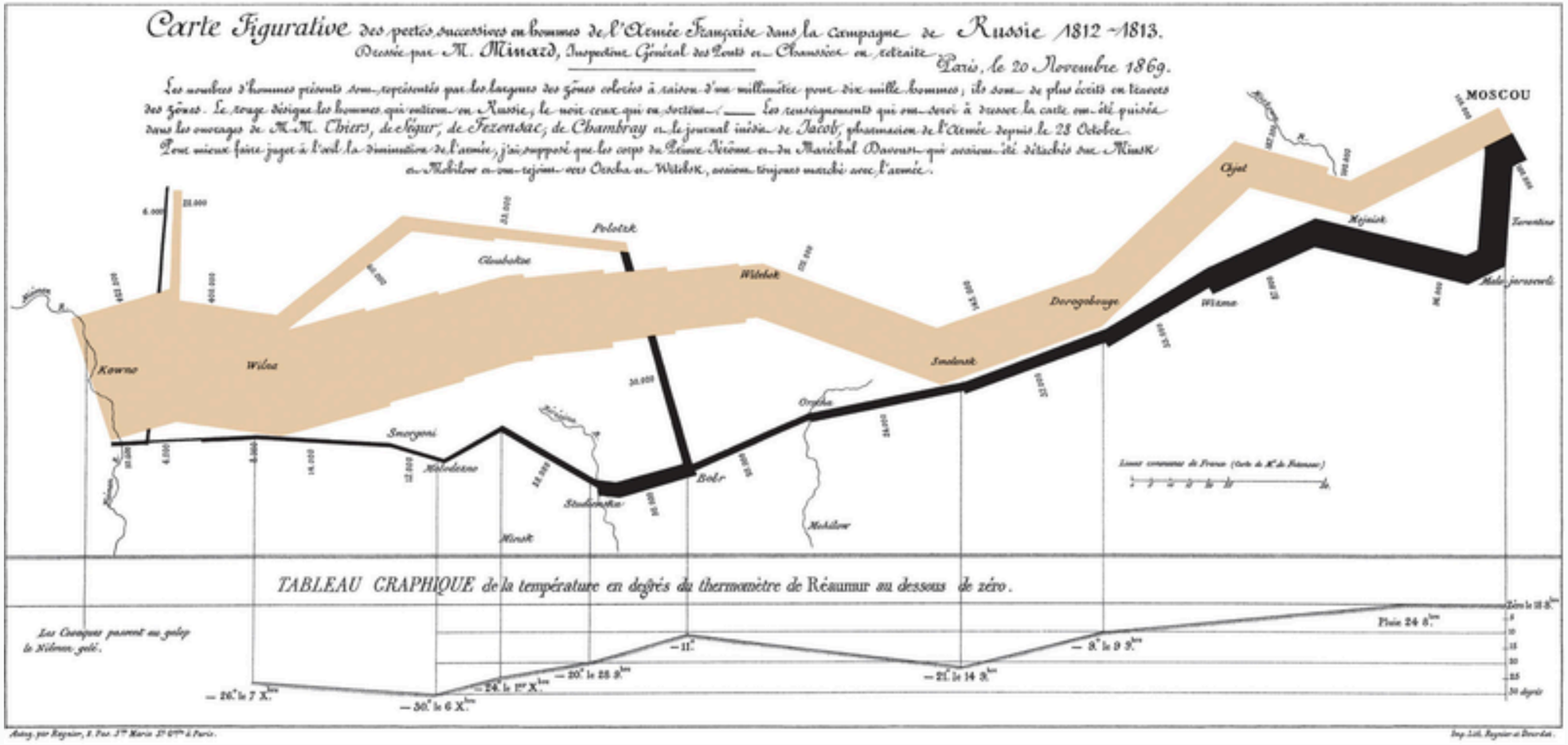


Image from Charles Joseph Minard, Wikipedia. https://en.wikipedia.org/wiki/Charles_Joseph_Minard

Cholera Map (1854)

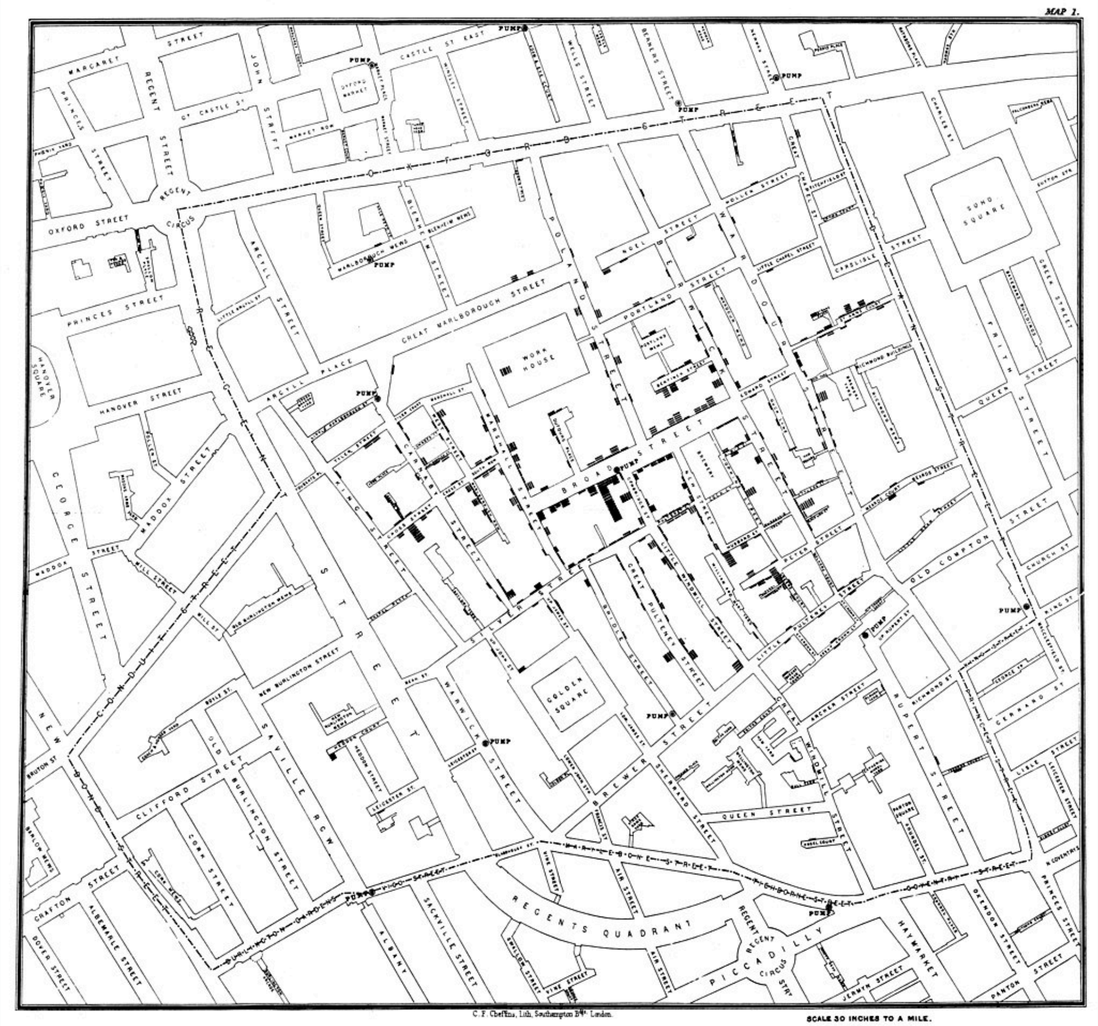


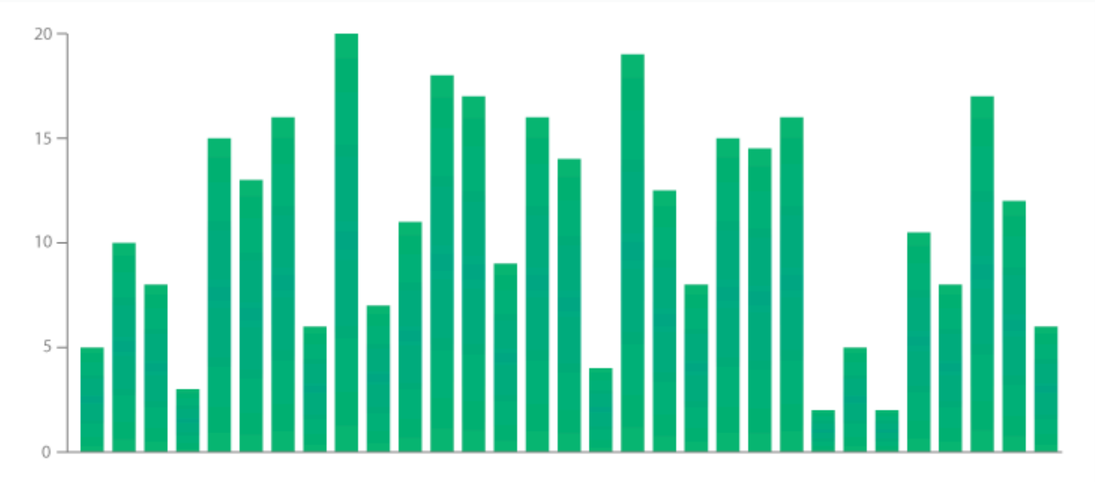
Image from John Snow, Wikipedia. https://en.wikipedia.org/wiki/John_Snow

Elements of Visualization

- Common data visualization solutions include:
  - charts, suitable for numeric data, especially when comparing data sources or groupings;
  - time-series, or timelines, help represent observations over time;
  - maps, focus on geographic properties of data;
  - interactive, where control is given to the user to explore different paths or options;
  - words, representing highlights (e.g. most frequent) of textual descriptions;

- The Data Visualization Catalogue, https://datavizcatalogue.com

Bar Chart



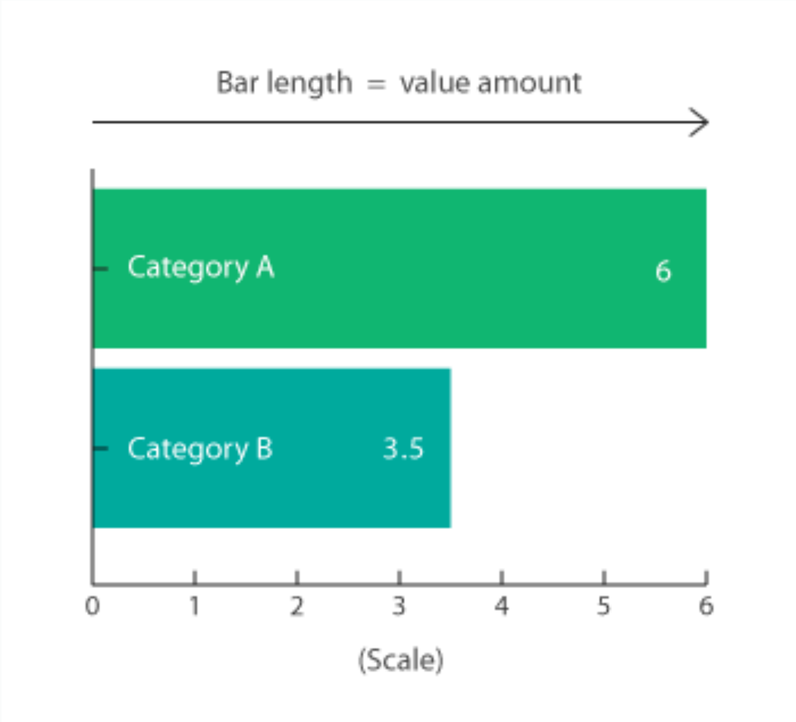
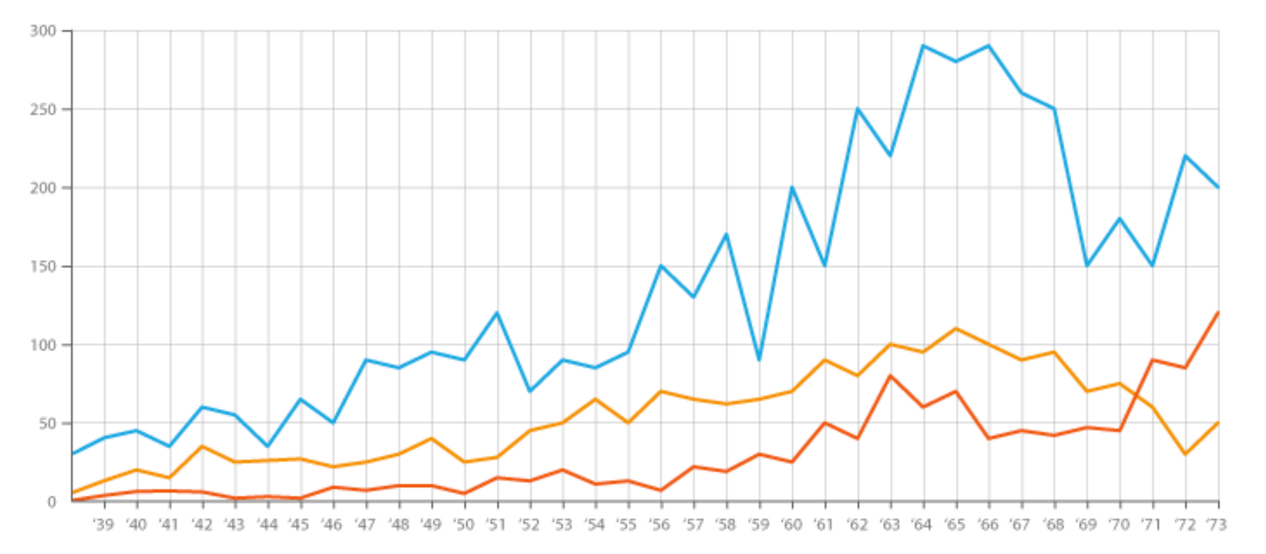


Image from Bar Chart, https://datavizcatalogue.com/methods/bar_chart.html

Line Graph



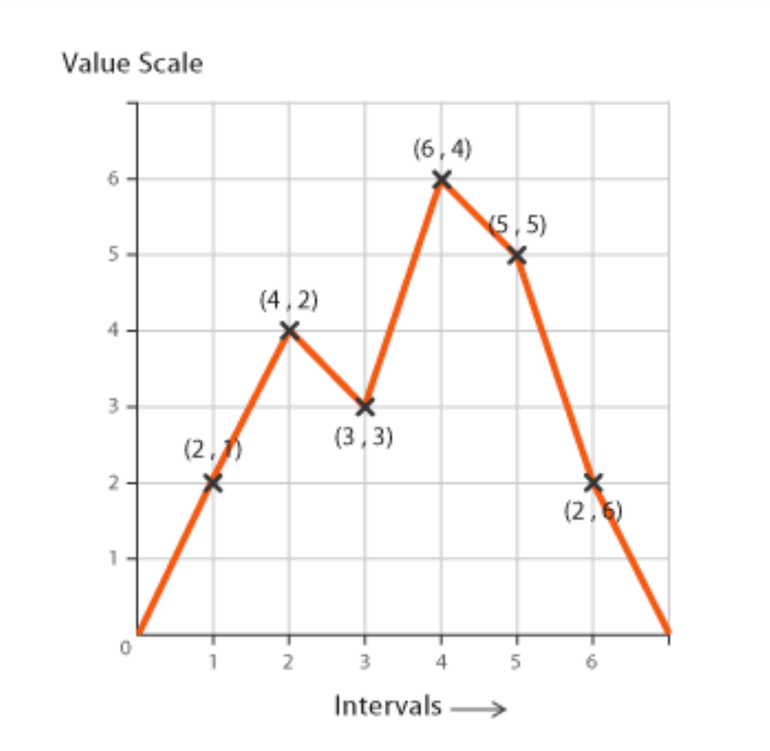
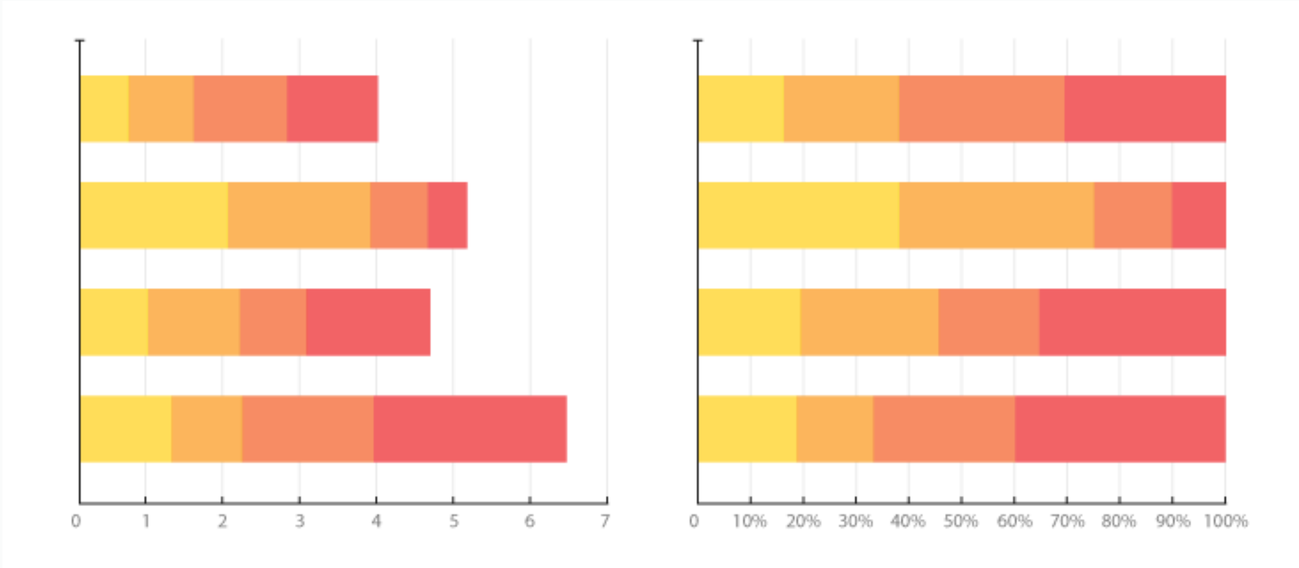


Image from Line Graph, https://datavizcatalogue.com/methods/line_graph.html

Stacked Bar Chart



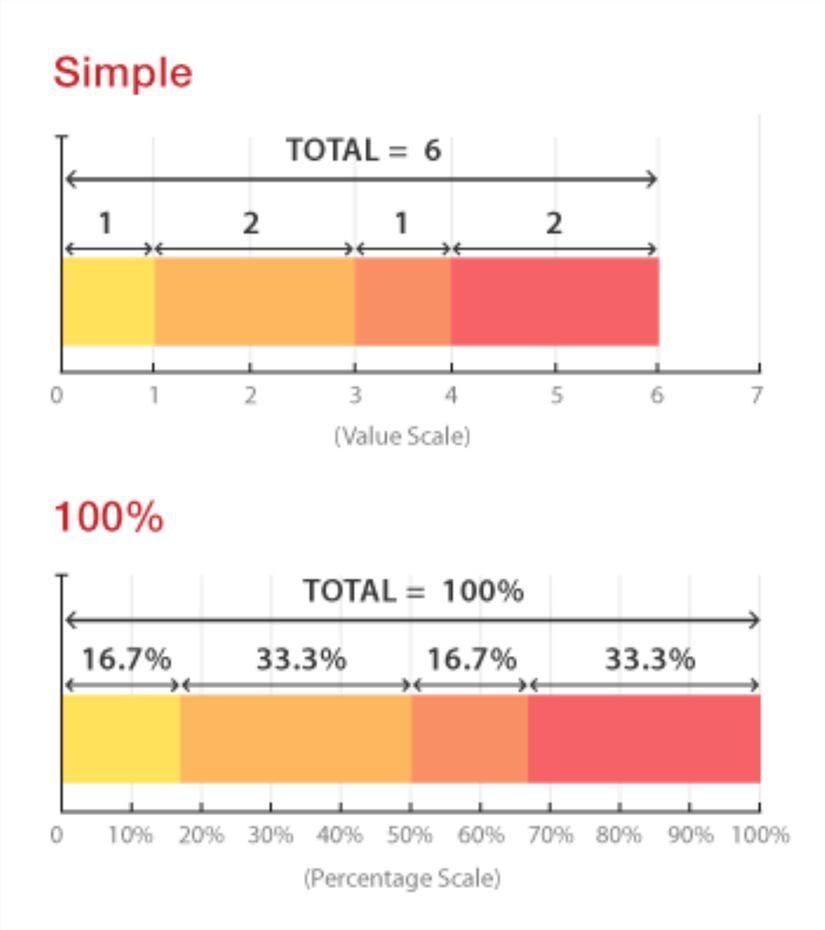
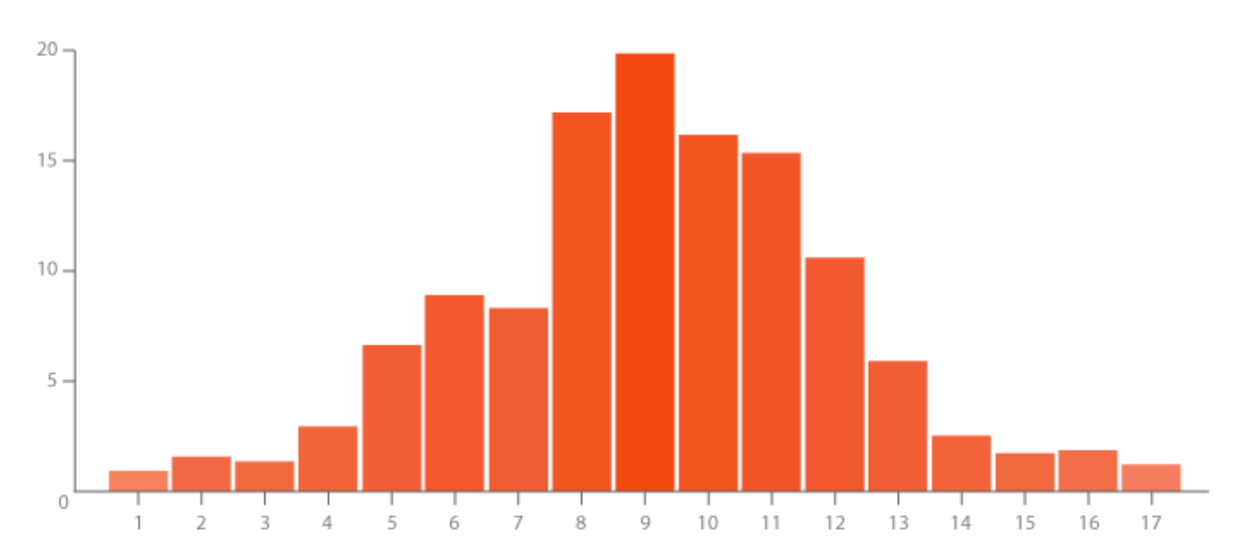


Image from Stacked Bar Chart, https://datavizcatalogue.com/methods/stacked_bar_graph.html

Histogram



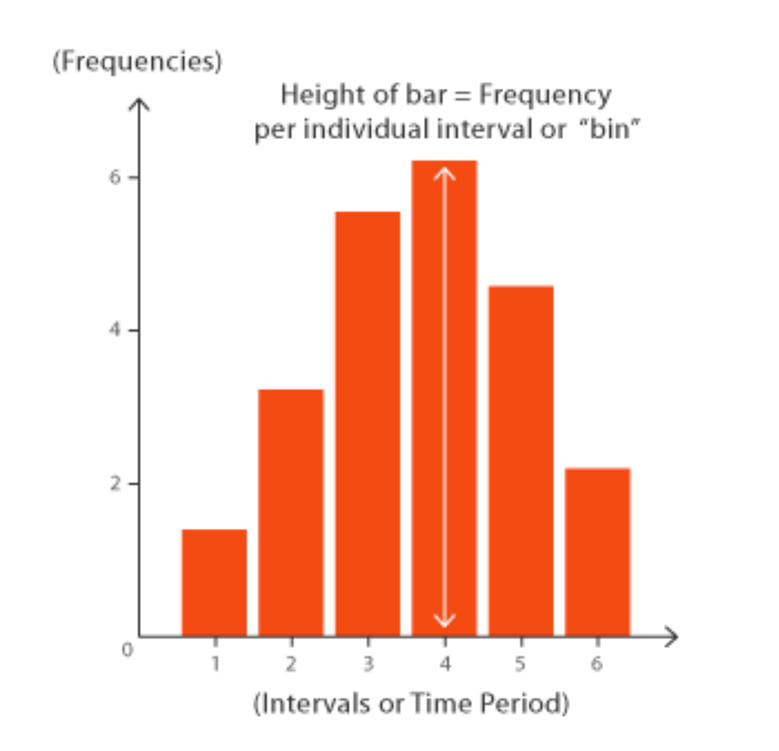
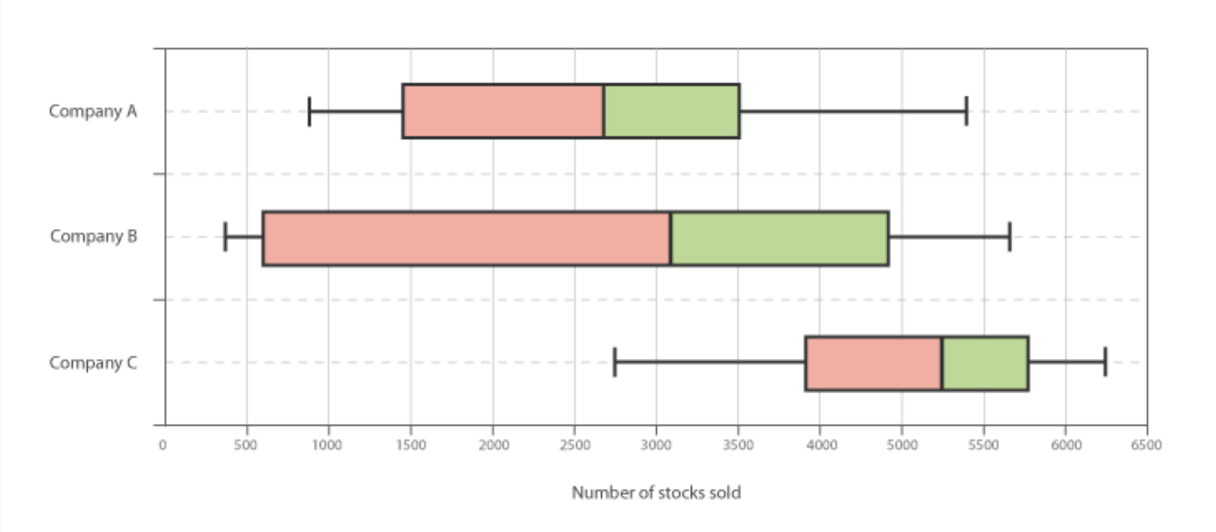


Image from Histogram, https://datavizcatalogue.com/methods/histogram.html

Box Plot



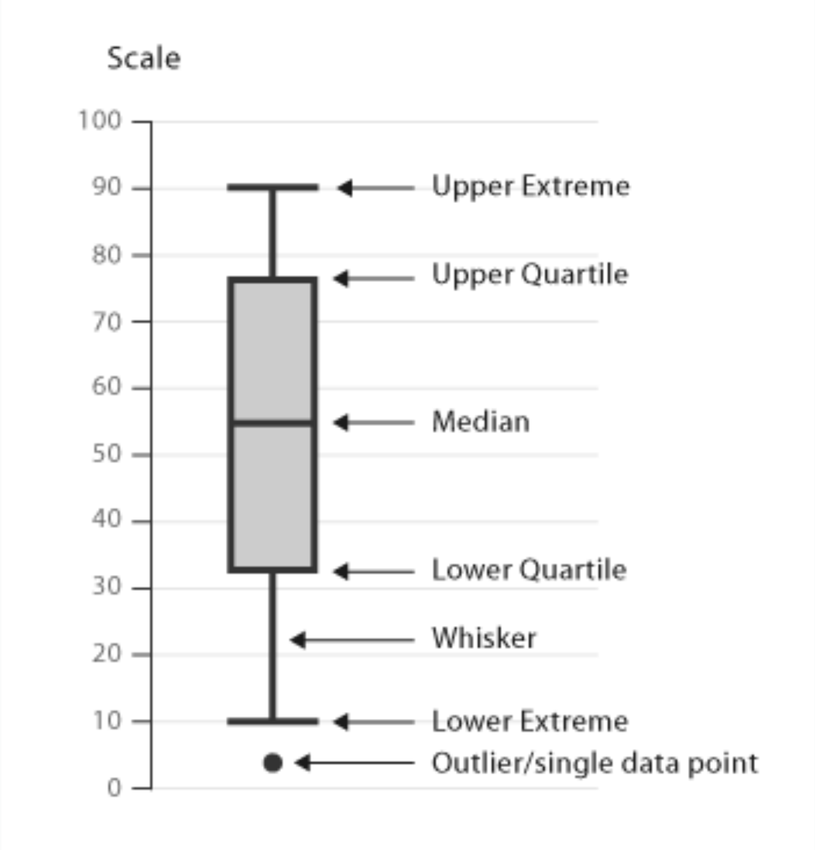
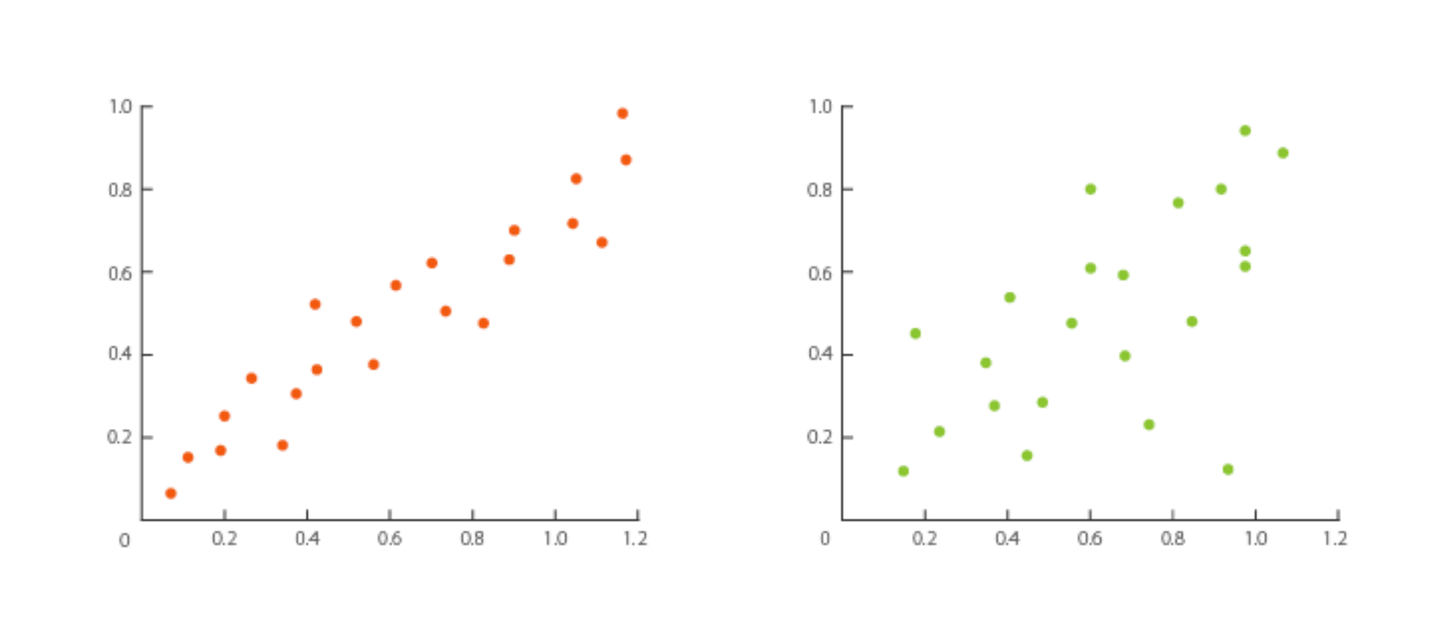


Image from Box Plot, https://datavizcatalogue.com/methods/box_plot.html

Scatter Plot



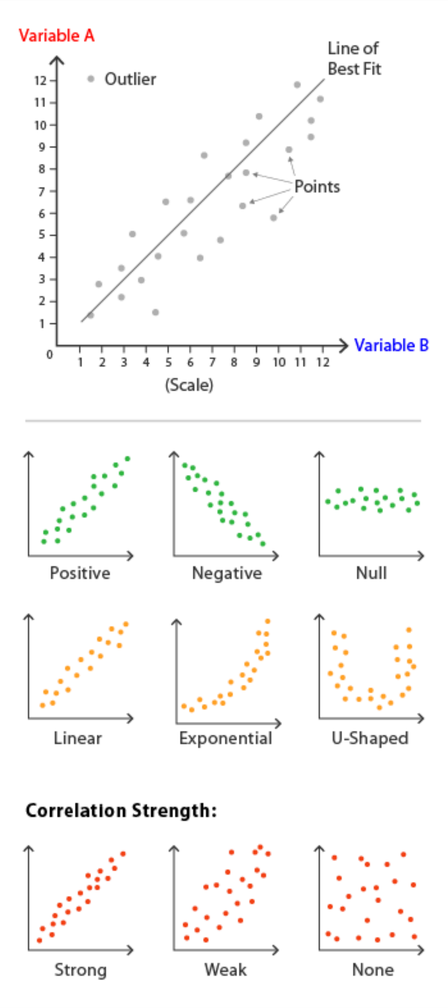


Image from Scatterplot, https://datavizcatalogue.com/methods/scatterplot.html

Word Cloud

Image from Word Cloud, https://datavizcatalogue.com/methods/wordcloud.html

Tools for Data Visualization

- D3.js (Data Driven Documents), d3js.org
- Vega, vega.github.io/vega
- ggplot2 (R), ggplot2.tidyverse.org
- Bokeh (Python), bokeh.org
- p5.js, p5js.org
- Leaflet (maps), leafletjs.com

Bibliography and Further Readings

- **Designing Data-Intensive Applications** [DDIA17]
  Kleppmann, M. O'Reilly, 2017

- **Principles of Data Wrangling** [PDW17]
  Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., and Carreras, C. O'Reilly, 2017

- Unix™ for Poets - https://www.cs.upc.edu/~padro/Unixforpoets.pdf
  Kenneth Ward Church.

Milestone 1

Milestone 1 - Data Preparation

- The first milestone is achieved with the preparation and characterization of the datasets selected for the project.
- Work on these tasks depends on the nature, volume, organization and accessibility of the selected datasets.
- Typical actions (depending on the data source):
  - search repositories for datasets
  - select convenient data subsets
  - assess the authority of the data source and data quality
  - perform exploratory data analysis
  - prepare and document a data processing pipeline
  - characterize the datasets, identifying and describing some of their properties
  - identify the conceptual model for the data domain
  - identify follow-up information needs in the data domain

Project Themes

- Discussion on project themes
- Case studies
  - Electoral programs: break in proposals / sections / paragraphs
  - Sports: search for "entity x" is not sufficient (trivial case)
- Common: combine different data sources
- Different starting points, e.g. already prepared datasets to raw unstructured data
  - Important: value added by the data processing pipeline must be clear
  - Prepare data for the next task - search system.

Data Processing

- Wikipedia and Wikidata are key resources in the data ecosystem.
- Wikipedia is often used as a source of text about entities.
  - An API is available - https://en.wikipedia.org/w/api.php
  - Organized in language-specific projects, e.g. https://pt.wikipedia.org
  - DBpedia produces structured snapshots of Wikipedia subsets - https://www.dbpedia.org
- Wikidata is used as a hub to link different collections, e.g. exploring references as entry points to distinct collections / domains.
  - Luís Vaz de Camões — https://www.wikidata.org/wiki/Q590
  - Portugal — https://www.wikidata.org/wiki/Q45

Domain Data Model

- The data preparation report includes a presentation of the domain data model, i.e. a representation of the domain data concepts and their interrelationships.
- Conceptual modeling is the process of discovering the entity types that represent the things and concepts, their relationships, pertinent to the system.
- It establishes a vocabulary and a shared vision for the system.
- Use a UML class diagram to document the domain data model.

Data Processing Pipeline

- The data preparation report includes a detailed presentation of the data processing pipeline.
- Document the pipeline describing the main decisions and key processes.
- We interested on the physical perspective, i.e. focus on implementation details - including data formats, libraries, storage solutions, software tools, etc.
- Include a data flow diagram (DFD). Key elements are:
  - External entities, which are sources or destinations of data.
  - Processes, that represent data processing operations (automatic or manual).
  - Data flows, that describe how and what data moves around.
  - Data stores, representing data storage structures (files, databases, etc).

Information Needs

- Prospective search tasks should be described considering "information needs" and the document types to be returned by the search system to be developed.
- Information needs represents the underlying motivation of a user to obtain information to satisfy a need, i.e. use a search system.
- Example of an information need:
  - Information on whether drinking red wine is more effective at reducing your risk of heart attacks than drinking white wine.
- This might be translated into a query such as:
  - [wine red white heart attack effective]
- A search system is successful if retrieved documents address the stated information need, not because they just contain all the words in the query.

LaTeX

- LaTeX should be used to prepare the final manuscript.
- LaTeX is a document processing and preparation system.
- Focus is on the content, separating writing from presentation and styling.
- Commonly used in academia to prepare scientific manuscripts and publications.
- You can either use a local installation or a web-based collaborative application such as Overleaf (overleaf.com).
- Use a two-column format, such as ACM LaTeX templates.
- Each milestone report is limited to 4 pages, excluding references.

M1 Submission

- Submissions are due next week, in Moodle.
- Deadline is on the day before each lab class (before 18h).
- Materials to submit: report (pdf) + presentation (pdf) + makefile (txt).
- See project rules for details on page limits and formats.

M1 Workshop

- The lab class is organized as a workshop for presentation and discussion.
- Each group has a 10 minute slot: 5 min for presentation, plus 5 min for questions.
- Questions are part of the 10% individual assessment for this milestone.
  - *The grade for deliveries has an individual component, that is positive if the student contributes to the workshop with questions or comments, and negative if the student is not present or shows unprofessional behavior.*
- Planned schedule and additional notes available in Moodle.

This Week's Lab Class (#3)

- Before class
  - Describe the data processing pipeline (diagram)
  - Describe prospective search tasks for milestone 2
- In class
  - Review project submission guidelines
  - Review workshop guidelines
  - Discuss data processing guideline
  - Discuss search tasks
  - Work on data extraction and characterization
  - Work on tutorials: makefiles and characterization
- After class
  - Prepare milestone 1 report and presentation

Questions or comments?

Information Retrieval

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Chapter 1 from Modern Information Retrieval, Baeza-Yates, R. et al. (2011)
Based on Chapters 1 and 2 from Search Engines in Practice, Croft, B. et al. (2008)

Project Milestone 1 — Data Preparation

Project M1: Data Preparation

- M1 presentations are this week.

- Deadline for submitting your materials is the day before the lab class (until 18h).
- Presentation schedules are tight.
  - Be there on time.
  - 5 min presentations + 5 min discussion.
- Don’t forget the individual participation component (10%) — ask questions!

Project Overview

- M1:
  - **from…** selected data sources
  - **to…** an organized (and studied) collection of documents
  - **plus…** and a set of prospective search tasks

- M2:
  - **from…** a collection of documents
  - **to…** the first version of a search system

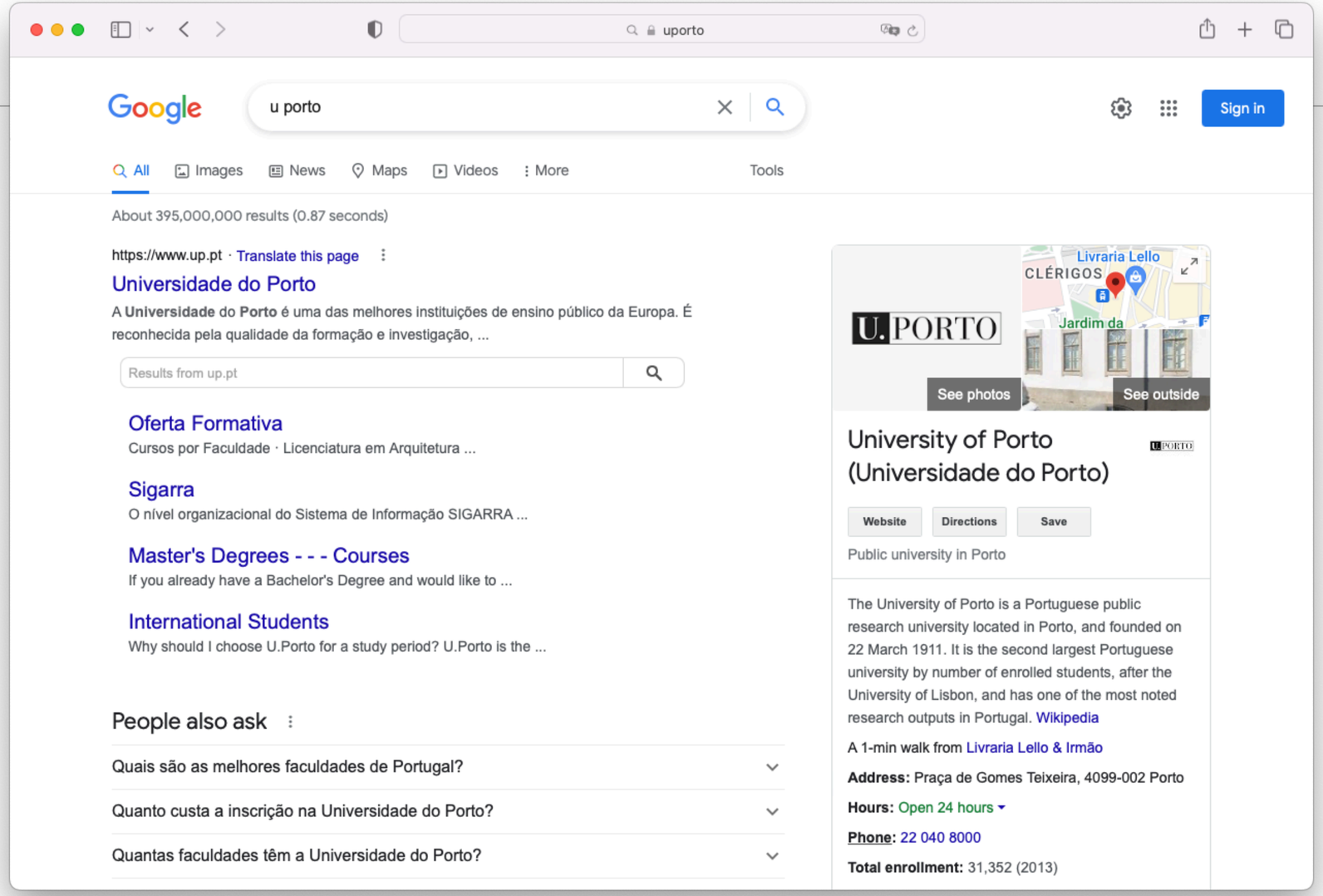
Today's Lecture Outline

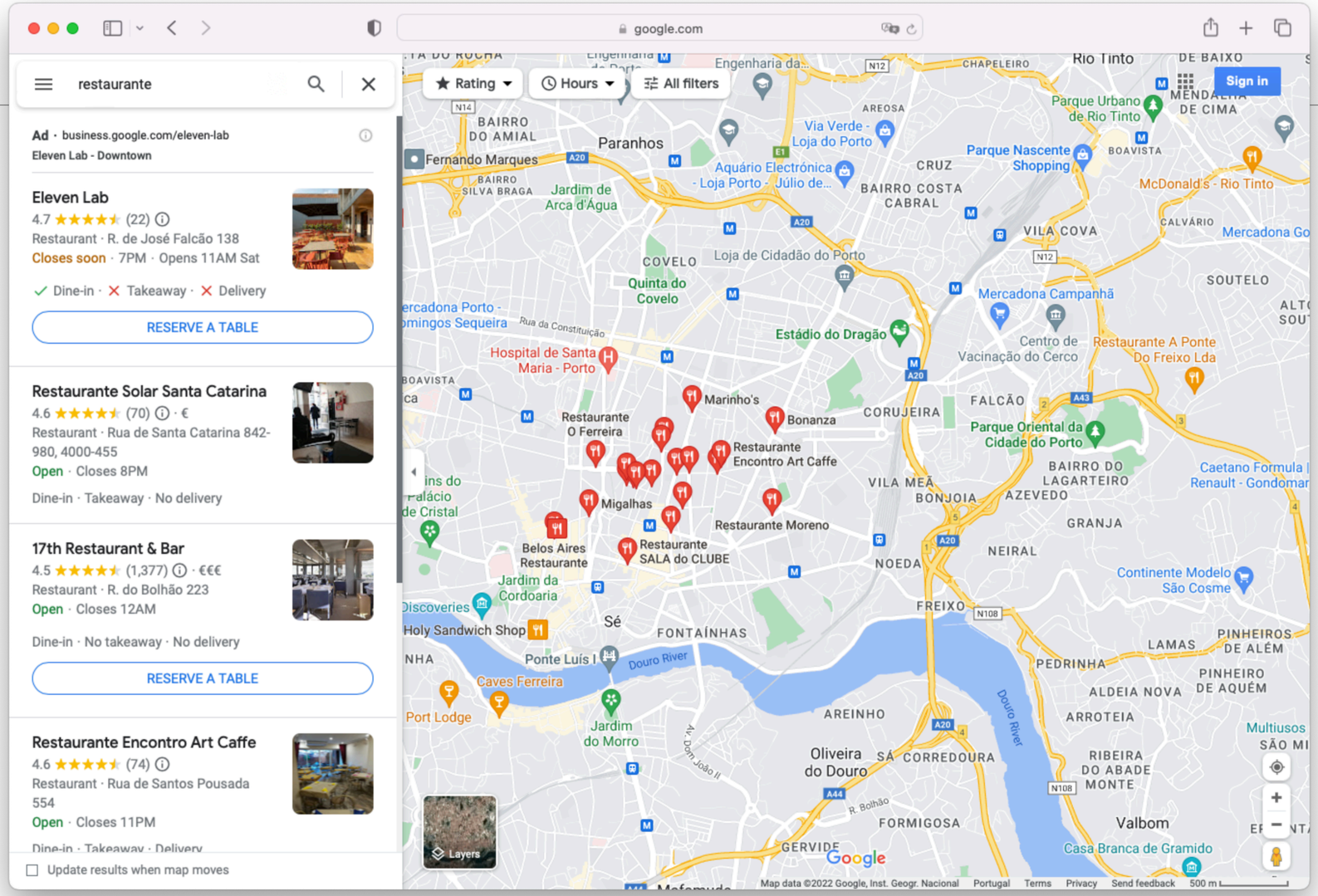
- Search systems
- Information retrieval concepts
  - Collections, documents, queries, results
  - Tasks, test collections, measures
  - Users, information needs, evaluation
- Architecture of information retrieval systems
  - Components of an information retrieval system
- History of information retrieval
  - Relation to text and data mining, machine learning, natural language processing

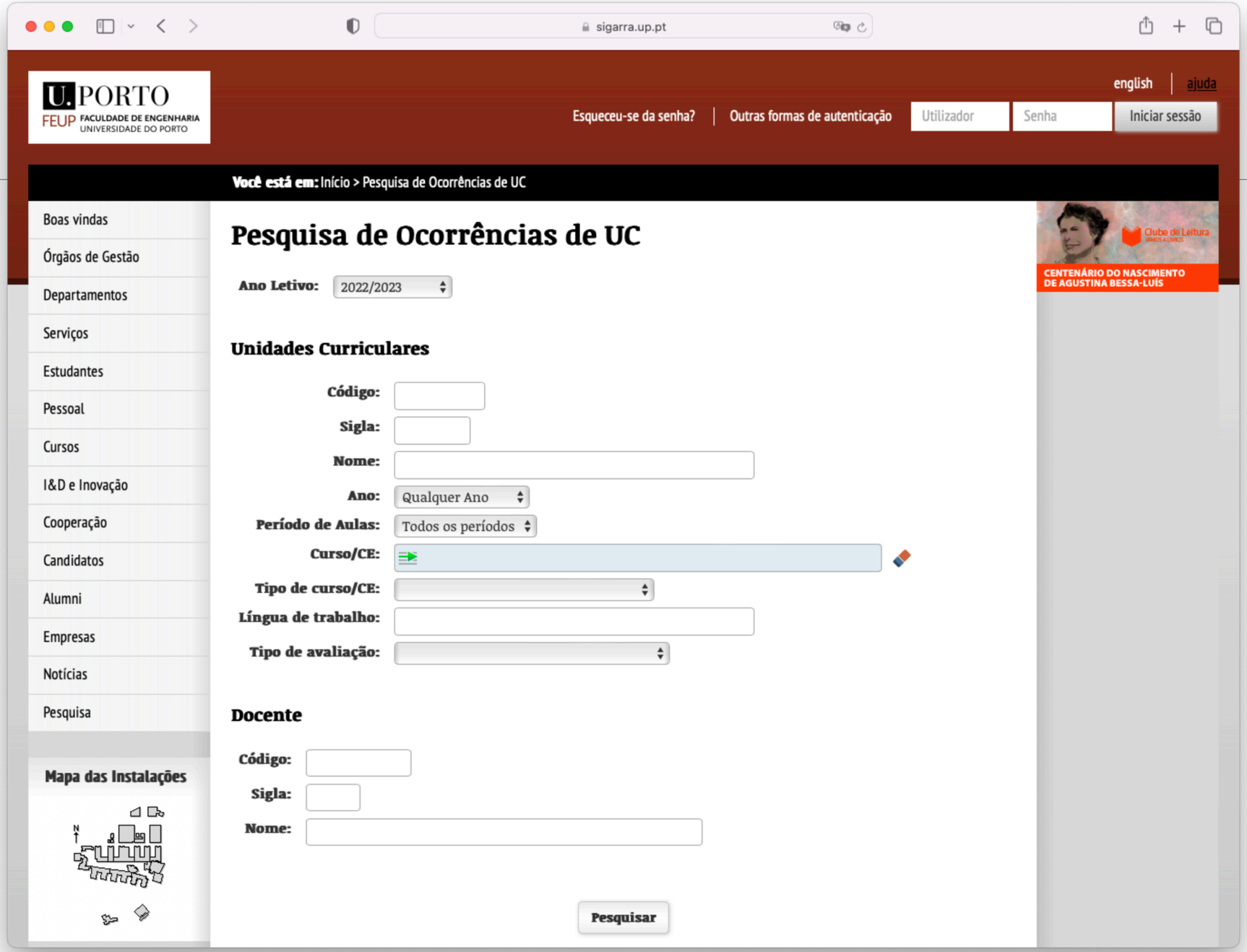
Search systems

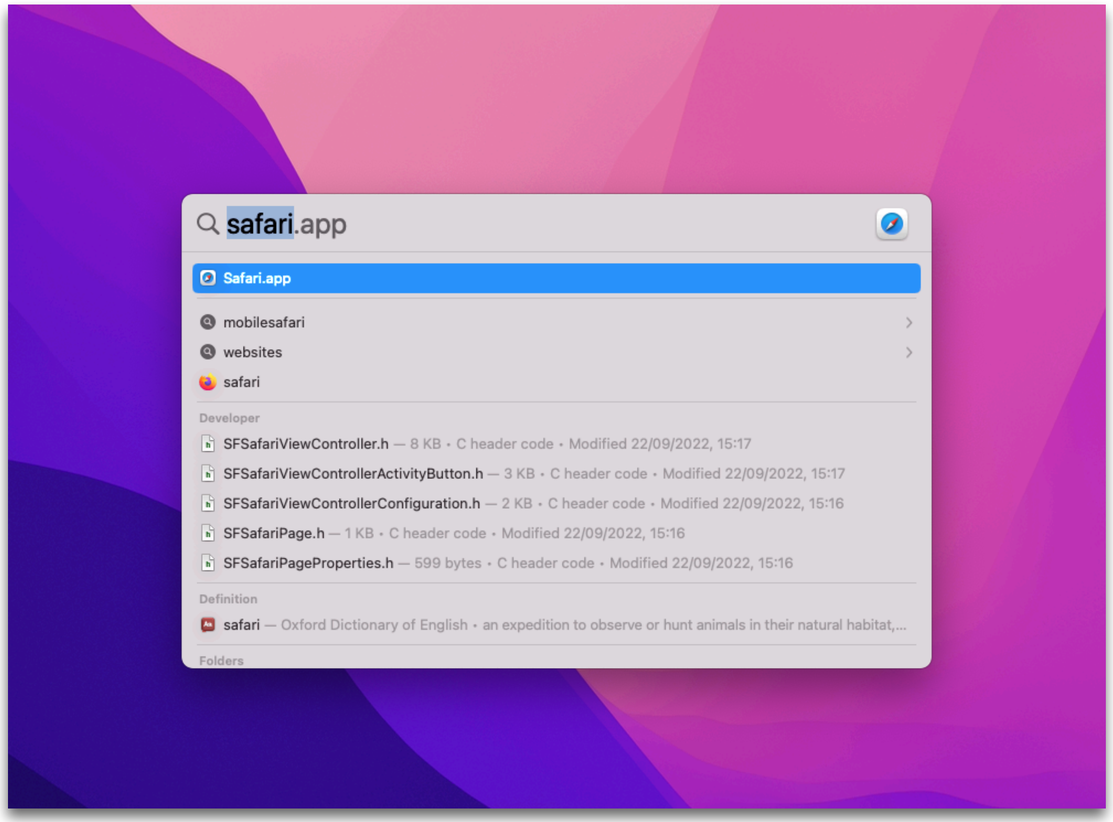
Search and Information Retrieval

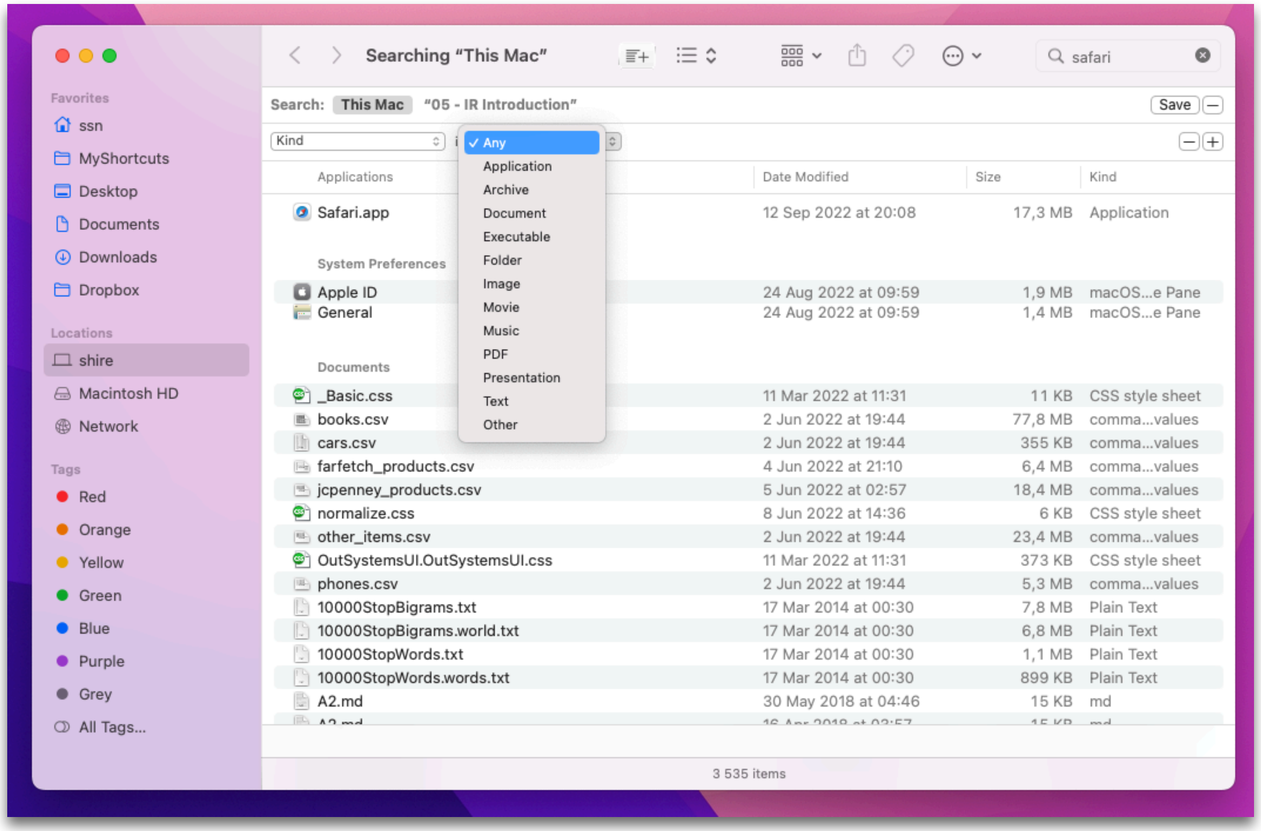
- ➔ Search is a daily activity for every computer, web, smart phone, … user.
- ➔ Search exists, *and is expected*, in every application.
- ➔ The web has brought the concept of keyword-based search to the center stage.
- ➔ Search contexts: web, site-specific, drive, computer, folders, email, messaging system.
- ➔ Many different types: web search, maps search, image similarity search, social search, etc.

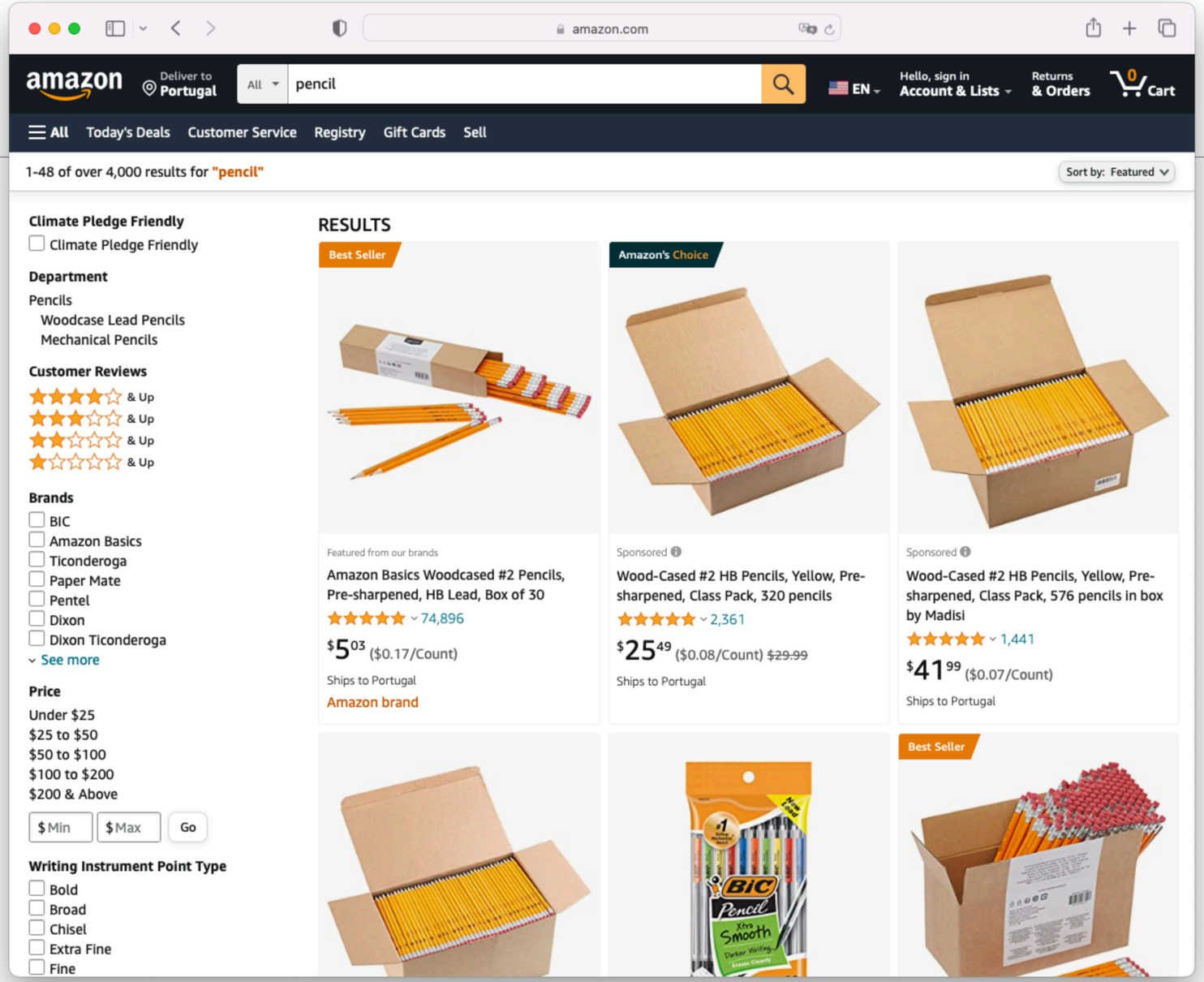

uporto
Google
u porto
Sign in
All
Images
News
Maps
Videos
More
Tools
About 395,000,000 results (0.87 seconds)
https://www.up.pt · Translate this page
Universidade do Porto
A Universidade do Porto é uma das melhores instituições de ensino público da Europa. É reconhecida pela qualidade da formação e investigação, ...
Results from up.pt
Oferta Formativa
Cursos por Faculdade · Licenciatura em Arquitetura ...
Sigarra
O nível organizacional do Sistema de Informação SIGARRA ...
Master's Degrees - - - Courses
If you already have a Bachelor's Degree and would like to ...
International Students
Why should I choose U.Porto for a study period? U.Porto is the ...
People also ask
Quais são as melhores faculdades de Portugal?
Quanto custa a inscrição na Universidade do Porto?
Quantas faculdades têm a Universidade do Porto?
Livraria Lello
CLÉRIGOS
Jardim da
U.PORTO
See photos
See outside
University of Porto (Universidade do Porto)
Website
Directions
Save
Public university in Porto
The University of Porto is a Portuguese public research university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of Lisbon, and has one of the most noted research outputs in Portugal. Wikipedia
A 1-min walk from Livraria Lello & Irmão
Address: Praça de Gomes Teixeira, 4099-002 Porto
Hours: Open 24 hours
Phone: 22 040 8000
Total enrollment: 31,352 (2013)


google.com
restaurante
Ad · business.google.com/eleven-lab
Eleven Lab - Downtown
Eleven Lab
4.7 (22)
Restaurant · R. de José Falcão 138
Closes soon · 7PM · Opens 11AM Sat
Dine-in · Takeaway · Delivery
RESERVE A TABLE
Restaurante Solar Santa Catarina
4.6 (70) · €
Restaurant · Rua de Santa Catarina 842-980, 4000-455
Open · Closes 8PM
Dine-in · Takeaway · No delivery
17th Restaurant & Bar
4.5 (1,377) · €€€
Restaurant · R. do Bolhão 223
Open · Closes 12AM
Dine-in · No takeaway · No delivery
RESERVE A TABLE
Restaurante Encontro Art Caffe
4.6 (74)
Restaurant · Rua de Santos Pousada 554
Open · Closes 11PM
Update results when map moves
Rating
Hours
All filters
Sign in
Paranhos
Fernando Marques
Jardim de Arca d'Água
Aquário Electrónica - Loja Porto - Júlio de...
Via Verde - Loja do Porto
Parque Urbano de Rio Tinto
Parque Nascente Shopping
McDonald's - Rio Tinto
Loja de Cidadão do Porto
Quinta do Covelo
Estádio do Dragão
Mercadona Campanhã
Centro de Vacinação do Cerco
Restaurante A Ponte Do Freixo Lda
Hospital de Santa Maria - Porto
Marinho's
Bonanza
Restaurante O Ferreira
Restaurante Encontro Art Caffe
Parque Oriental da Cidade do Porto
Migalhas
Restaurante Moreno
Belos Aires Restaurante
Restaurante SALA do CLUBE
Jardim da Cordoaria
Continente Modelo São Cosme
Holy Sandwich Shop
Sé
Ponte Luís I
Douro River
Caves Ferreira
Port Lodge
Jardim do Morro
Oliveira do Douro
Valbom
Casa Branca de Gramido
Google
Map data ©2022 Google, Inst. Geogr. Nacional
Portugal
Terms
Privacy
Send feedback
500 m

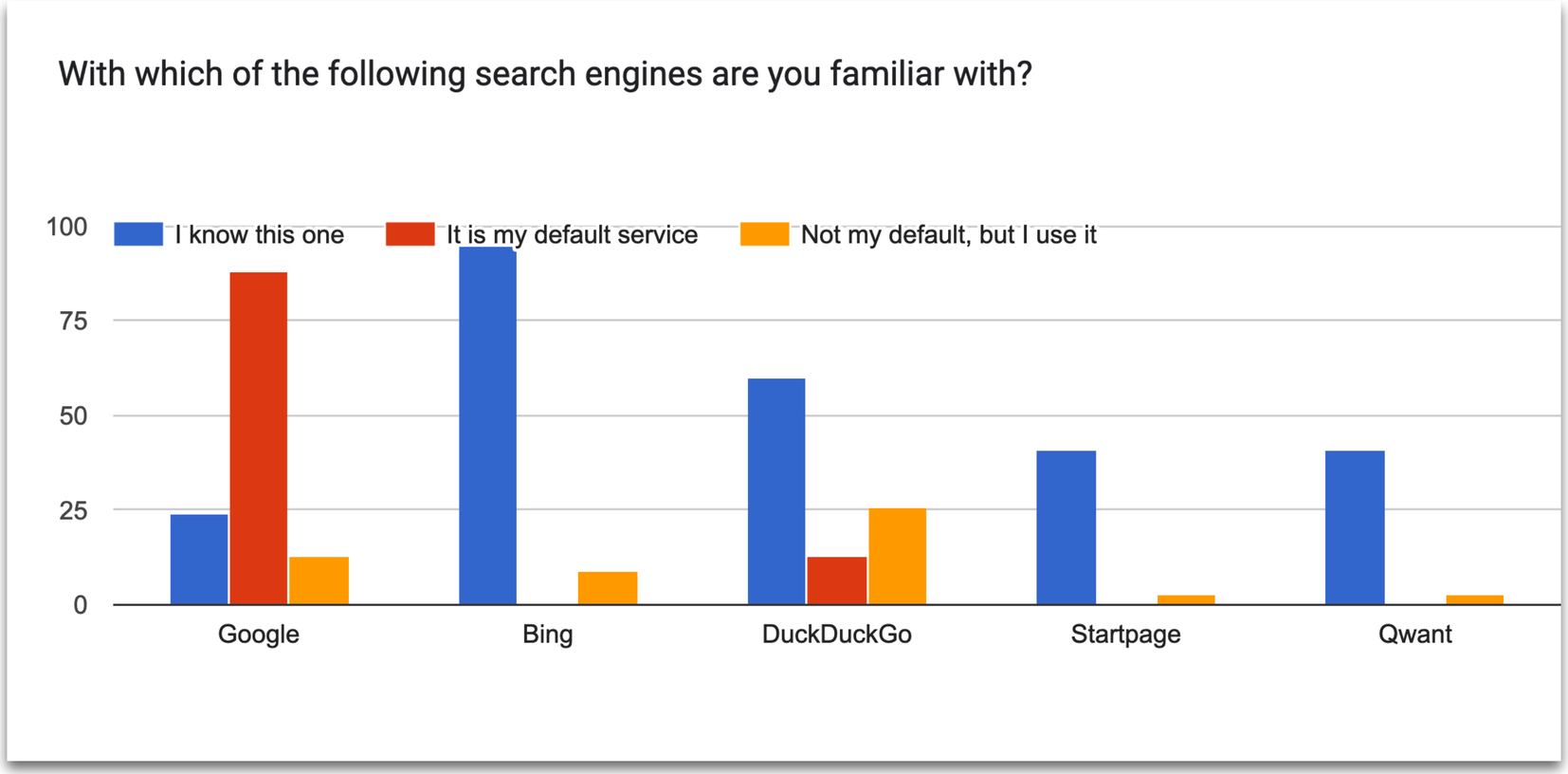

sigarra.up.pt
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
english
ajuda
Esqueceu-se da senha?
Outras formas de autenticação
Utilizador
Senha
Iniciar sessão
Você está em: Início > Pesquisa de Ocorrências de UC
Boas vindas
Órgãos de Gestão
Departamentos
Serviços
Estudantes
Pessoal
Cursos
I&D e Inovação
Cooperação
Candidatos
Alumni
Empresas
Notícias
Pesquisa
Mapa das Instalações
Pesquisa de Ocorrências de UC
Ano Letivo: 2022/2023
Unidades Curriculares
Código:
Sigla:
Nome:
Ano: Qualquer Ano
Período de Aulas: Todos os períodos
Curso/CE:
Tipo de curso/CE:
Língua de trabalho:
Tipo de avaliação:
Docente
Código:
Sigla:
Nome:
Pesquisar
Clube de Leitura
CENTENÁRIO DO NASCIMENTO DE AGUSTINA BESSA-LUÍS


safari.app
Safari.app
mobilesafari
websites
safari
Developer
SFSafariViewController.h — 8 KB • C header code • Modified 22/09/2022, 15:17
SFSafariViewControllerActivityButton.h — 3 KB • C header code • Modified 22/09/2022, 15:17
SFSafariViewControllerConfiguration.h — 2 KB • C header code • Modified 22/09/2022, 15:16
SFSafariPage.h — 1 KB • C header code • Modified 22/09/2022, 15:16
SFSafariPageProperties.h — 599 bytes • C header code • Modified 22/09/2022, 15:16
Definition
safari — Oxford Dictionary of English • an expedition to observe or hunt animals in their natural habitat,...
Folders


Searching "This Mac"
safari
Favorites
ssn
MyShortcuts
Desktop
Documents
Downloads
Dropbox
Locations
shire
Macintosh HD
Network
Tags
Red
Orange
Yellow
Green
Blue
Purple
Grey
All Tags...
Search: This Mac "05 - IR Introduction"
Save
Kind
Any
Application
Archive
Document
Executable
Folder
Image
Movie
Music
PDF
Presentation
Text
Other
Applications
Date Modified
Size
Kind
Safari.app 12 Sep 2022 at 20:08 17,3 MB Application
System Preferences
Apple ID 24 Aug 2022 at 09:59 1,9 MB macOS...e Pane
General 24 Aug 2022 at 09:59 1,4 MB macOS...e Pane
Documents
_Basic.css 11 Mar 2022 at 11:31 11 KB CSS style sheet
books.csv 2 Jun 2022 at 19:44 77,8 MB comma...values
cars.csv 2 Jun 2022 at 19:44 355 KB comma...values
farfetch_products.csv 4 Jun 2022 at 21:10 6,4 MB comma...values
jcpenney_products.csv 5 Jun 2022 at 02:57 18,4 MB comma...values
normalize.css 8 Jun 2022 at 14:36 6 KB CSS style sheet
other_items.csv 2 Jun 2022 at 19:44 23,4 MB comma...values
OutSystemsUI.OutSystemsUI.css 11 Mar 2022 at 11:31 373 KB CSS style sheet
phones.csv 2 Jun 2022 at 19:44 5,3 MB comma...values
10000StopBigrams.txt 17 Mar 2014 at 00:30 7,8 MB Plain Text
10000StopBigrams.world.txt 17 Mar 2014 at 00:30 6,8 MB Plain Text
10000StopWords.txt 17 Mar 2014 at 00:30 1,1 MB Plain Text
10000StopWords.words.txt 17 Mar 2014 at 00:30 899 KB Plain Text
A2.md 30 May 2018 at 04:46 15 KB md
3 535 items


amazon.com
amazon
Deliver to Portugal
All
pencil
EN
Hello, sign in
Account & Lists
Returns
& Orders
0
Cart
All
Today's Deals
Customer Service
Registry
Gift Cards
Sell
1-48 of over 4,000 results for "pencil"
Sort by: Featured
Climate Pledge Friendly
Climate Pledge Friendly
Department
Pencils
Woodcase Lead Pencils
Mechanical Pencils
Customer Reviews
& Up
& Up
& Up
& Up
Brands
BIC
Amazon Basics
Ticonderoga
Paper Mate
Pentel
Dixon
Dixon Ticonderoga
See more
Price
Under $25
$25 to $50
$50 to $100
$100 to $200
$200 & Above
$ Min
$ Max
Go
Writing Instrument Point Type
Bold
Broad
Chisel
Extra Fine
Fine
RESULTS
Best Seller
Featured from our brands
Amazon Basics Woodcased #2 Pencils, Pre-sharpened, HB Lead, Box of 30
74,896
$5.03 ($0.17/Count)
Ships to Portugal
Amazon brand
Amazon's Choice
Sponsored
Wood-Cased #2 HB Pencils, Yellow, Pre-sharpened, Class Pack, 320 pencils
2,361
$25.49 ($0.08/Count) $29.99
Ships to Portugal
Sponsored
Wood-Cased #2 HB Pencils, Yellow, Pre-sharpened, Class Pack, 576 pencils in box by Madisi
1,441
$41.99 ($0.07/Count)
Ships to Portugal
Best Seller
BIC
Pencil
Xtra Smooth

Student Survey



https://www.youtube.com/watch?v=0eKVizvYSUQ

Information retrieval concepts

Information Retrieval

- In computer science, the field that is most involved in search problems and developments is
  - Information Systems > **Information Retrieval**, and includes sub-areas such as:
    - Document representation
    - Information retrieval query processing
    - Users and interactive retrieval
    - Retrieval models and ranking
    - Retrieval tasks and goals
    - Evaluation of retrieval results
    - Search engines architectures and scalability

Defining Information Retrieval

- “Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information”, Salton (1968)
  - General definition that can be applied to many types of search applications.
  - Primary focus since the 50s has been on text and documents.

- “Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)”, Manning et al. (2008)

Documents

- Documents can be web pages, email messages, books, news articles, research papers, computer files (PDF, Word, text), text messages, etc.

- Common properties:
  - Significant text content.
  - Some structure, e.g. title, body, date, tags, author.

- From PRI projects:
  - linux packages, songs lyrics, cooking recipes, MSc dissertations, game plays, etc.

Different from Databases

- “A database management system is a software system that enables the creation, maintenance, and use of large amounts of data” (Abiteboul et al., 1995)
- Information retrieval and database systems have a lot in common.
- But emphasize different aspects of information management.
  - Databases contains highly structured data, queried using formal query languages, and the results are set-based — results are true or false for a given condition.
  - Information retrieval systems support interactive processes, support natural language queries, and results are rank-based — results are more or less relevant for a given information need. Not just text matching!

Information Retrieval and Databases

Table 1.1 Comparison of database systems and information retrieval, based on [40]

	Database systems	Information retrieval
Data type	Numbers, short strings	Text
Foundation	Algebraic/logic based	Probabilistic/statistics based
Search paradigm	Boolean retrieval	Ranked retrieval
Queries	Structured query languages	Free text queries
Evaluation criteria	Efficiency	Effectiveness (user satisfaction)
User	Programmer	Nontechnical person

Table from Entity-Oriented Search, Balog (2018).

Documents vs. Records (tuples) Queries

- Example **database queries**
  - All students enrolled in course “ABC” in year 22/23.
  - All bank account transferences between account A and account B, within a given date interval, ordered by transferred amount.

- Example **search engine queries**
  - [feup enrolled students ABC 22/23]
  - [weather porto site:pt]

Information Retrieval Tasks

- Information retrieval applications and research are typically structured around tasks.
- Tasks address specific contexts, and specific information access problems.
- Examples of classical information retrieval tasks:
  - Ad-hoc search, find relevant documents for an arbitrary text query.
  - Filtering, identify relevant user profiles for a new document.
  - Classification, identify relevant labels for documents.
  - Question answering, give specific answer to a question.
- Yearly evaluation initiatives with classical and novel tasks at TREC
  - https://trec.nist.gov/pubs/call2022.html

Dimensions of Information Retrieval

Content	Applications	Tasks
Text	Web search	Ad hoc search
Images	Vertical search	Filtering
Video	Enterprise search	Classification
Scanned docs	Desktop search	Question answering
Audio	Forum search	
Music	P2P search	
	Literature search	

Image from Search Engines: Information Retrieval in Practice, Croft et al. (2009)

Central issues in information retrieval

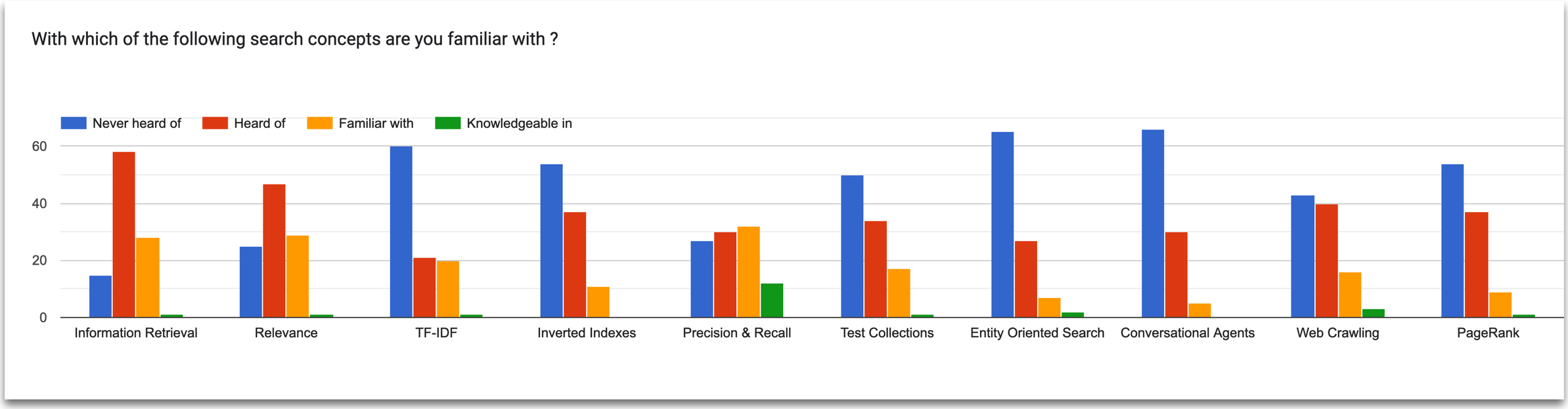
Relevance

- The concept of relevance is a key issue in information retrieval.
- Many factors impact a person's decision of what is relevant,
  e.g. task, context, previous knowledge, etc.
- Simple definition of relevance,
  - a relevant document contains the information that a person was looking for when submitting a query to a search engine.

Evaluation

- Evaluation setups are based on the use of annotated test collections of documents, queries, and relevance judgements.
- Evaluation in information retrieval dates back to the 1960s (Cranfield paradigm).
- IR evaluation methodologies have been adopted by many fields.
  - Recall and precision are two examples of widely used measures.
- Information needs drive the search process and result from the underlying task users' have.

Student Survey - IR Concepts



Information Retrieval and Search Engines

- ➔ Search engines are practical applications of information retrieval techniques.
- ➔ Existing open-source and commercial solutions that support development and research:
  - ➔ Apache Solr, Elasticsearch, Open Search, Terrier.

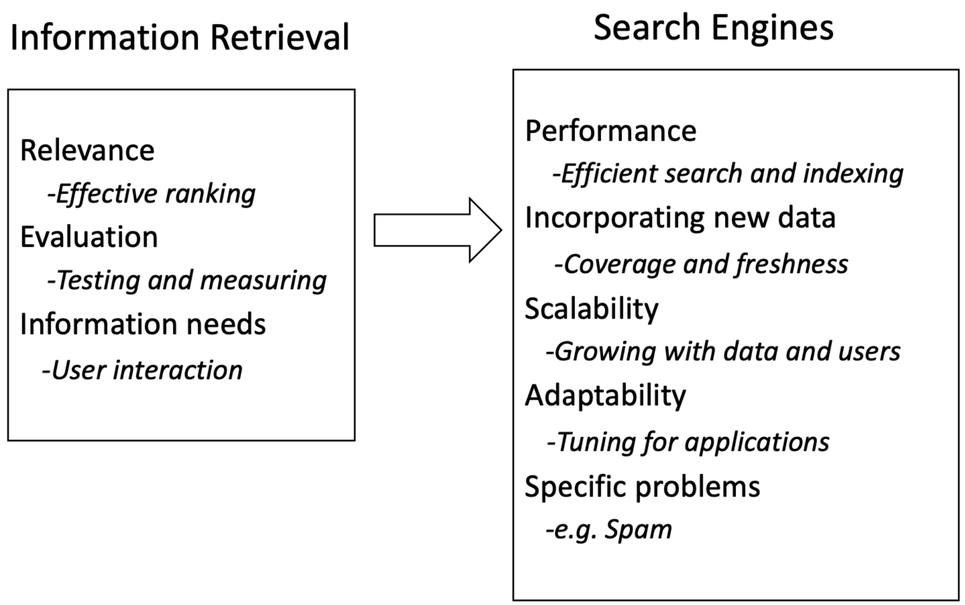
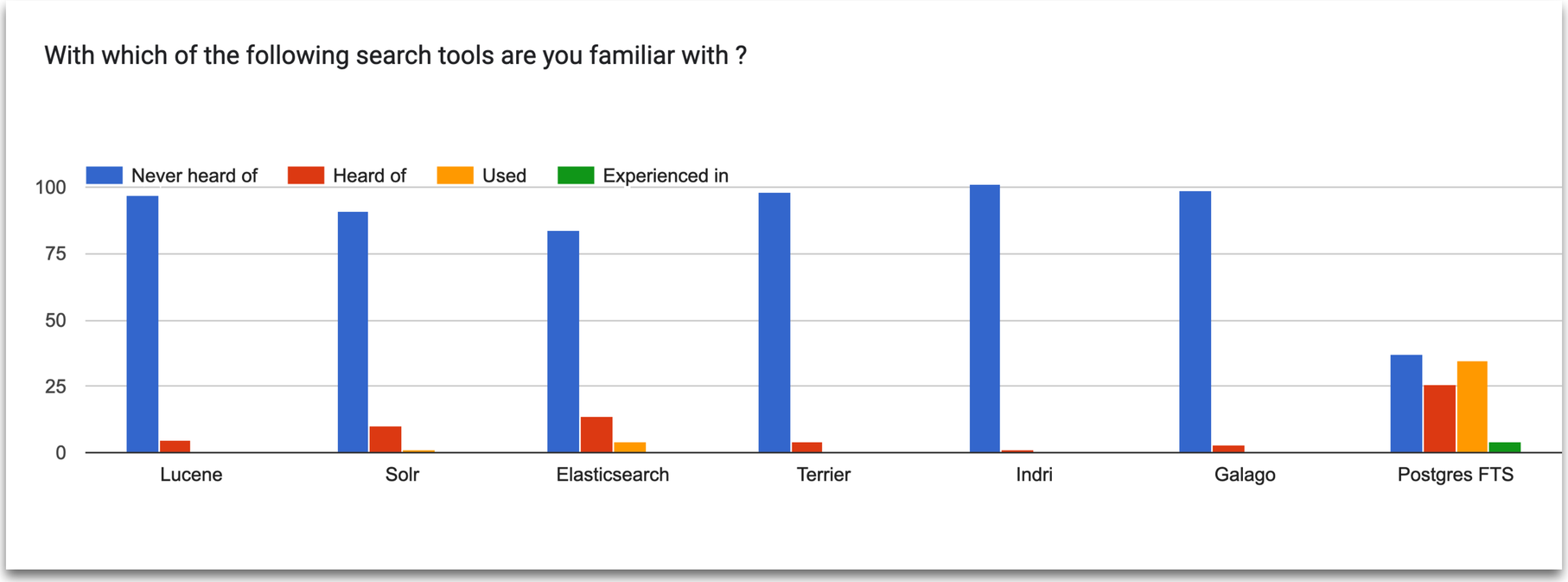


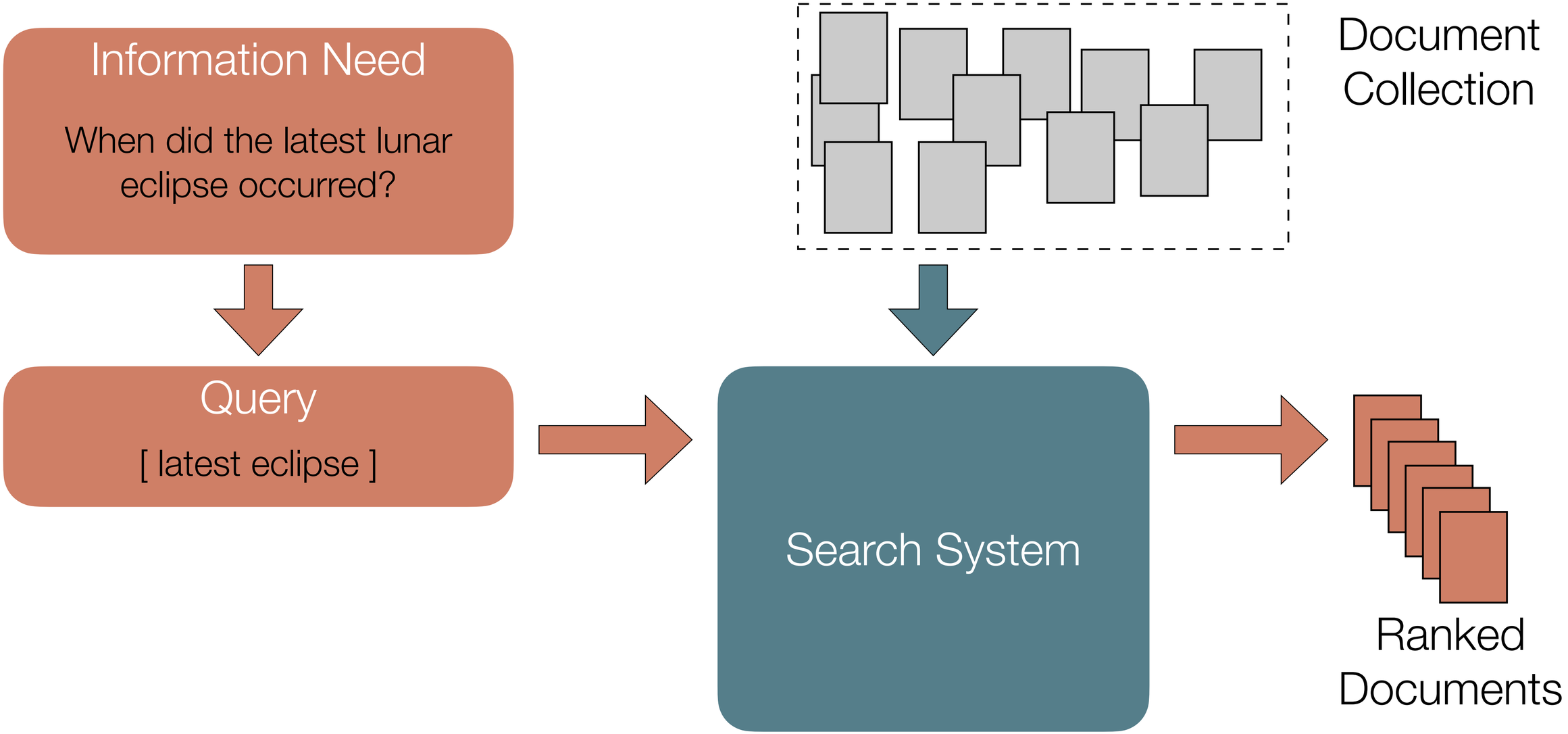
Image from Search Engines: Information Retrieval in Practice, Croft et al. (2009)

Student Survey - IR Technologies



Architecture of information retrieval systems

Classic Information Retrieval



Architecture of an Information Retrieval System

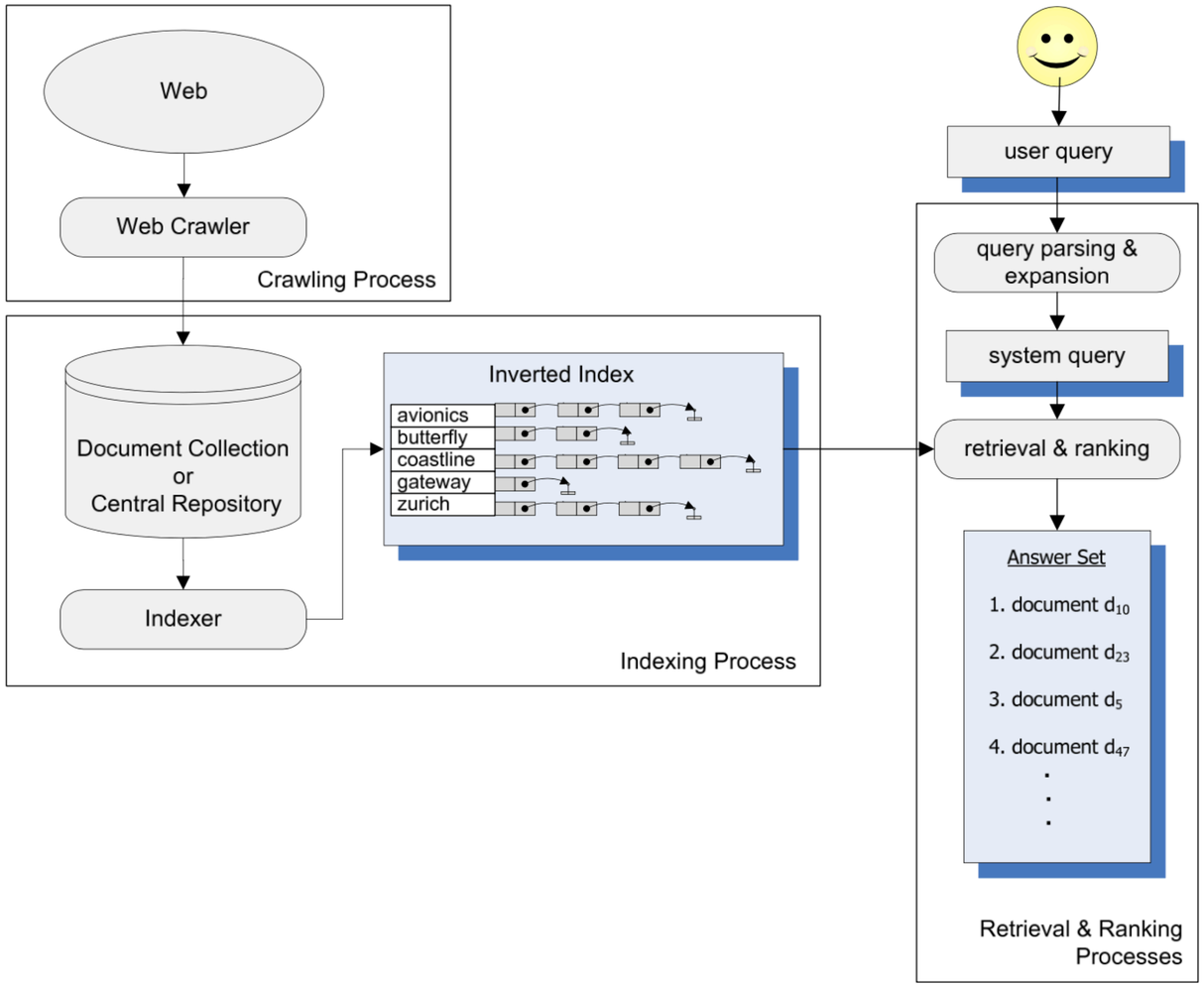


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

Indexing Process

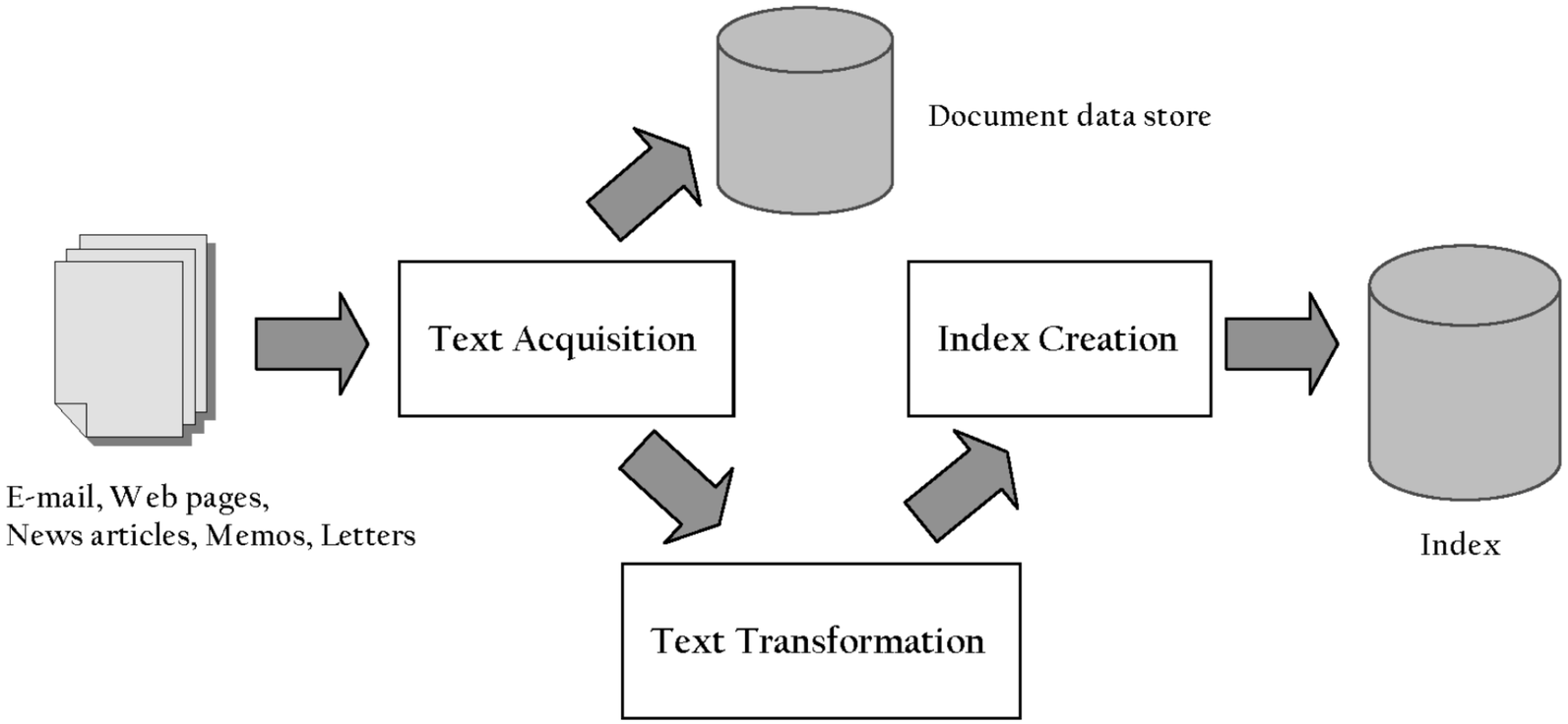


Image from Search Engines: Information Retrieval in Practice, Croft et al. (2009)

Query Process

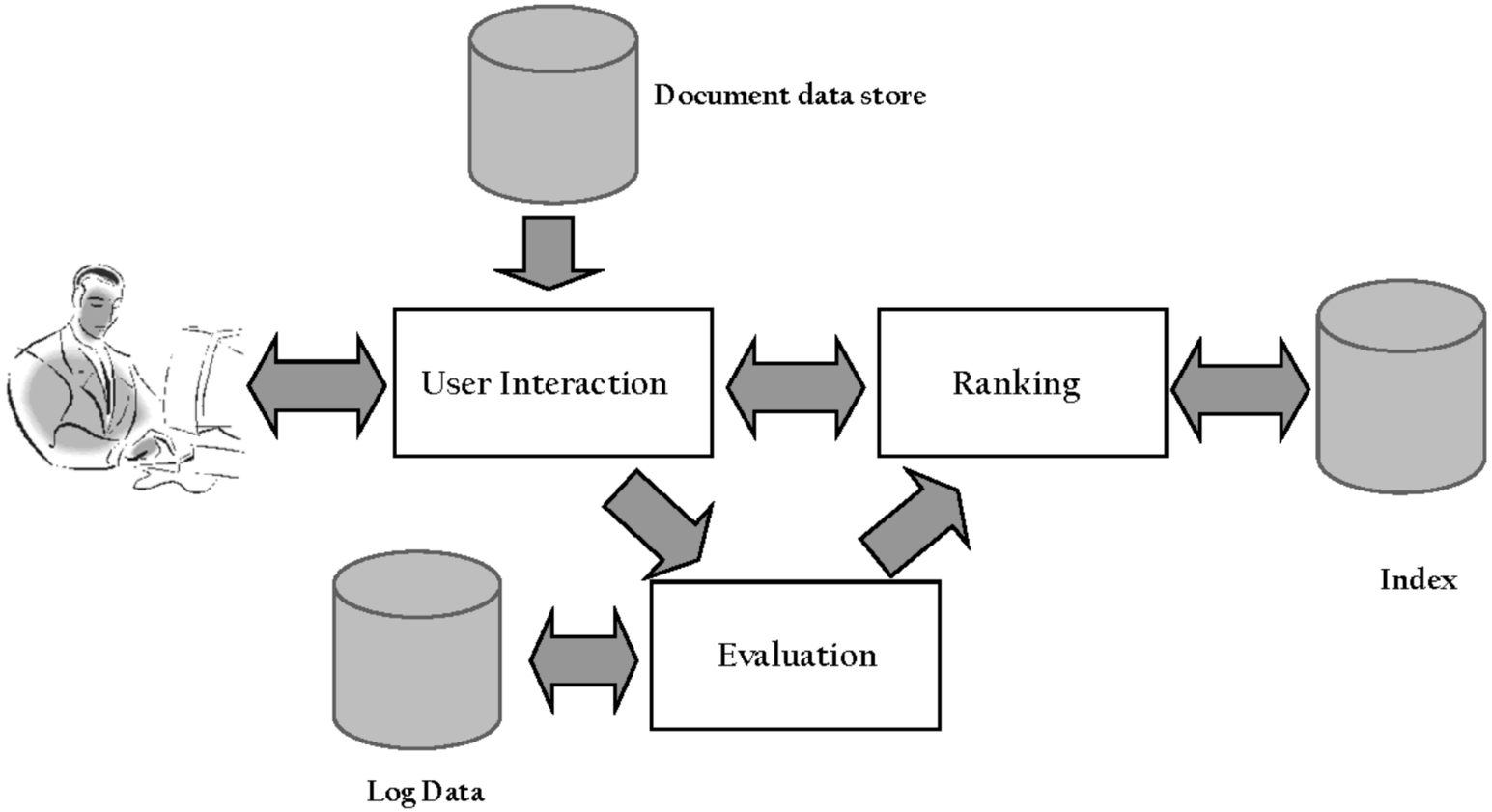


Image from Search Engines: Information Retrieval in Practice, Croft et al. (2009)

History of information retrieval

Historic Context

- At the end of World War II, Vannebar Bush, who headed the R&D office during the war, reflected on applications of new technologies in peace times.
- He wrote a landmark essay entitled As We May Think in 1945, where a new device, the Memex, capable of managing and interlinking knowledge is described.
  - https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/
- This essay greatly influenced the next generation of innovators.
  - Douglas Engelbart and the development of the computer mouse ("the mother of demos"), 1968.
  - Ted Nelson and the development of the concept of hypertext (Project Xanadu), 1960s.
  - Tim Berners-Lee and the World Wide Web, in 1989.
  - Web search engines, such as Lycos (94), Altavista (95), and Google (98).

Searching the Web

- ➔ The web represented a unique search challenge.
- ➔ A unique document collection, millions of documents, connected through hyperlinks, and distributed through multiple repositories (required crawling).
- ➔ The size of the collection and the user base, millions of documents and users. Performance and scalability became a key issue.
- ➔ Predicting relevance in such a large and heterogeneous collection is complex. New sources of evidence appeared (e.g. hyperlinks).
- ➔ The web is also a business platform, thus users needs are not limited to text search.
- ➔ Web spam is a real problem on the web, sometimes referred to as “adversarial information retrieval”.

Information Retrieval Summary

- ➔ Information Retrieval dates back to 1940s.
- ➔ Well established field in Computer Science.
- ➔ Deals with a wide range of information access problems.
- ➔ Organized in tasks: web search, filtering, question answering, podcast search, conversational, enterprise / intranet search, entity-oriented search
- ➔ Strong links with information science, NLP, machine learning, databases, etc.

- ➔ Core concepts: information need, index, query, relevance.

Course Plan

Course Plan

- ➔ Information Retrieval Overview (**today**)
- ➔ Evaluation
- ➔ Models and Indexing
- ➔ Retrieval and Web Retrieval
- ➔ Query Processing
- ➔ Entity-Oriented Search
- ➔ Neural IR

Lab Classes

- Tutorials on Solr and IR Evaluation

- Develop M2: Information Retrieval
  - Index
  - Retrieve
  - Evaluate

References and Further Reading

- Modern Information Retrieval (2nd Edition) (2011)
  Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Addison-Wesley Professional.
  http://www.mir2ed.org

- Introduction to Information Retrieval (2008)
  Christopher Manning, Prabhakar Raghavan, Hinrich Schütze. Cambridge University Press.
  http://nlp.stanford.edu/IR-book

- Search Engines: Information Retrieval in Practice (2009)
  W. Bruce Croft, Donald Metzler, Trevor Strohman. Pearson.
  http://ciir.cs.umass.edu/downloads/SEIRiP.pdf

- Entity-Oriented Search (2018)
  Krisztian Balog. Springer One.
  https://link.springer.com/book/10.1007/978-3-319-93935-3

Evaluation in Information Retrieval

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Chapter 8 from Introduction to Information Retrieval, Manning, C. et al. (2008)

PRI Status

- Week Plan
  - Lecture: Information Retrieval Evaluation
  - Labs: Solr tutorial; Collection indexing;

- Start of Milestone #2
  - Define documents
  - Index documents
  - Retrieval testing

Outline for Today

- Information Retrieval Evaluation
  - Review of main IR concepts
  - Test collections
  - Offline evaluation metrics
  - Online evaluation strategies

- Solr tutorial

Notes on PRI Project

Milestone #1

Notes on Milestone #1 Reports

- Scientific writing is expected
  - Scientific writing is technical writing for communication in science
  - Factual, concise, evidence-based
  - Avoid creative, verbose prose
  - Attention to expressions with rigorous meanings in CS, Math or Engineering, e.g. "significant" (statistics), "exponential growth" (math, CS).
  - Do not use "decorative" figures
  - Abstracts are not an introduction to the document, they should act as 'mini-documents' with all the main elements: why, what, how, results.

Notes on Milestone #1 Reports

- Other aspects
  - Missing information needs — necessary for milestone 2 evaluation
  - Missing analysis of textual fields — important to understand the data

- Follow-up deliveries build upon the previous
  - correct the main problems reported in the evaluation.
  - use \parts (Latex) for each milestone;

Milestone #2

Milestone #2

- The second milestone is achieved with the implementation and use of an information retrieval tool on the project datasets and its exploration with free-text queries.
  - choose the information retrieval tool (Solr, Elasticsearch, Lucene, Terrier, …);
  - analyze the documents and identify their indexable components;
  - use the selected tool to build the indexes;
  - use the selected tool to configure and execute the queries;
  - demonstrate the indexing and retrieval processes;
  - manually evaluate the returned results;
  - evaluate the results obtained for the defined information needs.

Milestone #3

Milestone #3

- The 3rd milestone is achieved with the development of the final version of the search system.
- This version is an **improvement over the previous milestone**, making use of features and techniques with the goal of improving the quality of the search results.
- For this milestone, each group is expected to explore innovative approaches and ideas, and will heavily depend on the context and data of each group.
- An **extended evaluation of the results** and **a comparison with the previous version** of the search system is expected.
- As examples of topics to explore, groups **may**:
  - Incorporate new information retrieval algorithms;
  - Expand the information available for each document by adding and linking new datasets;
  - Work on user interfaces by developing a frontend for the search system (**not sufficient**);

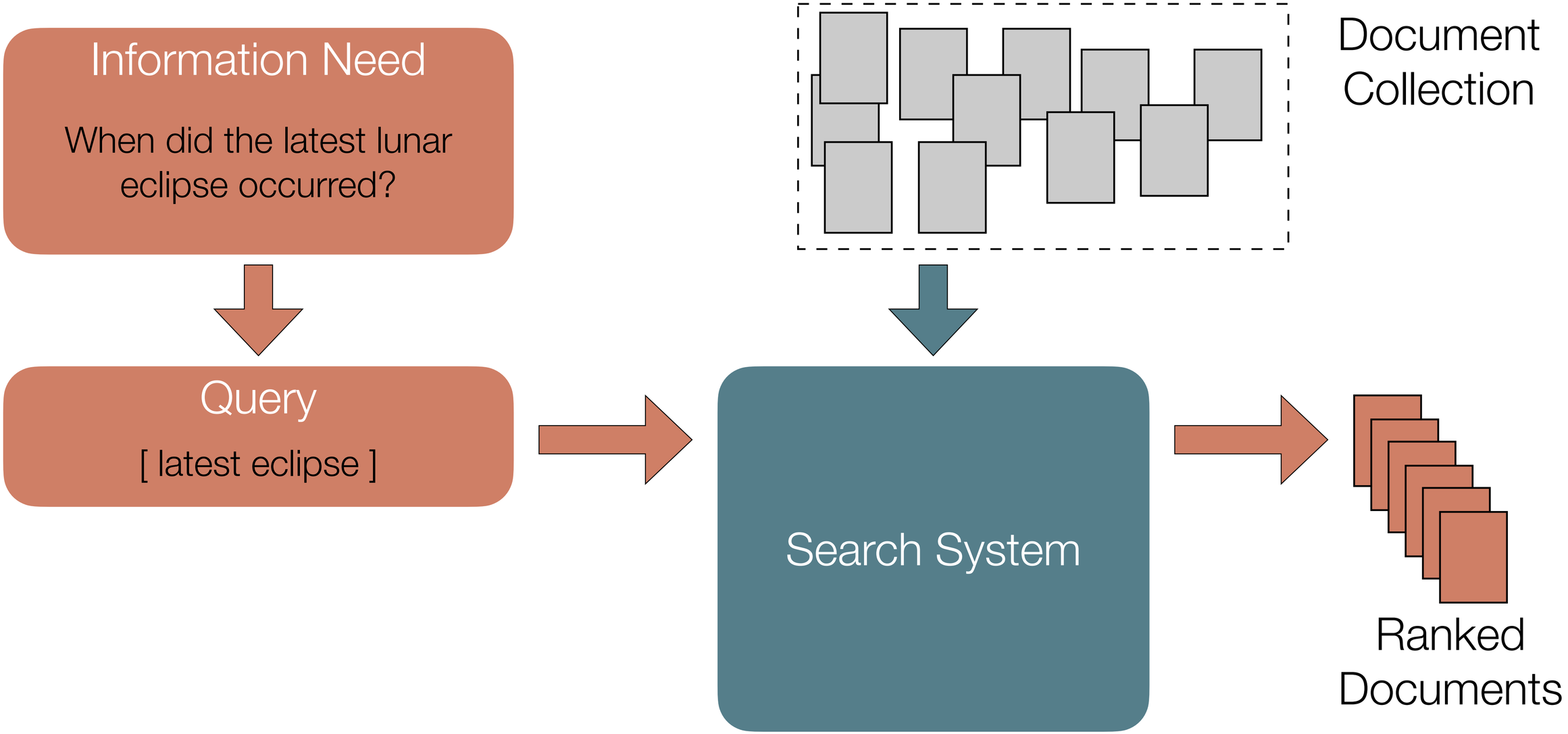
Information Retrieval Recap

Information Retrieval Recap

- ➔ Information Retrieval dates back to 1940s.
- ➔ Well established field in Computer Science.
- ➔ Deals with a wide range of information access problems.
- ➔ Organized in tasks: web search, filtering, question answering, podcast search, conversational, enterprise / intranet search, entity-oriented search
- ➔ Strong links with information science, NLP, machine learning, databases, etc.

- ➔ Core concepts: information need, index, query, relevance.

Classic Information Retrieval



Architecture of an Information Retrieval System

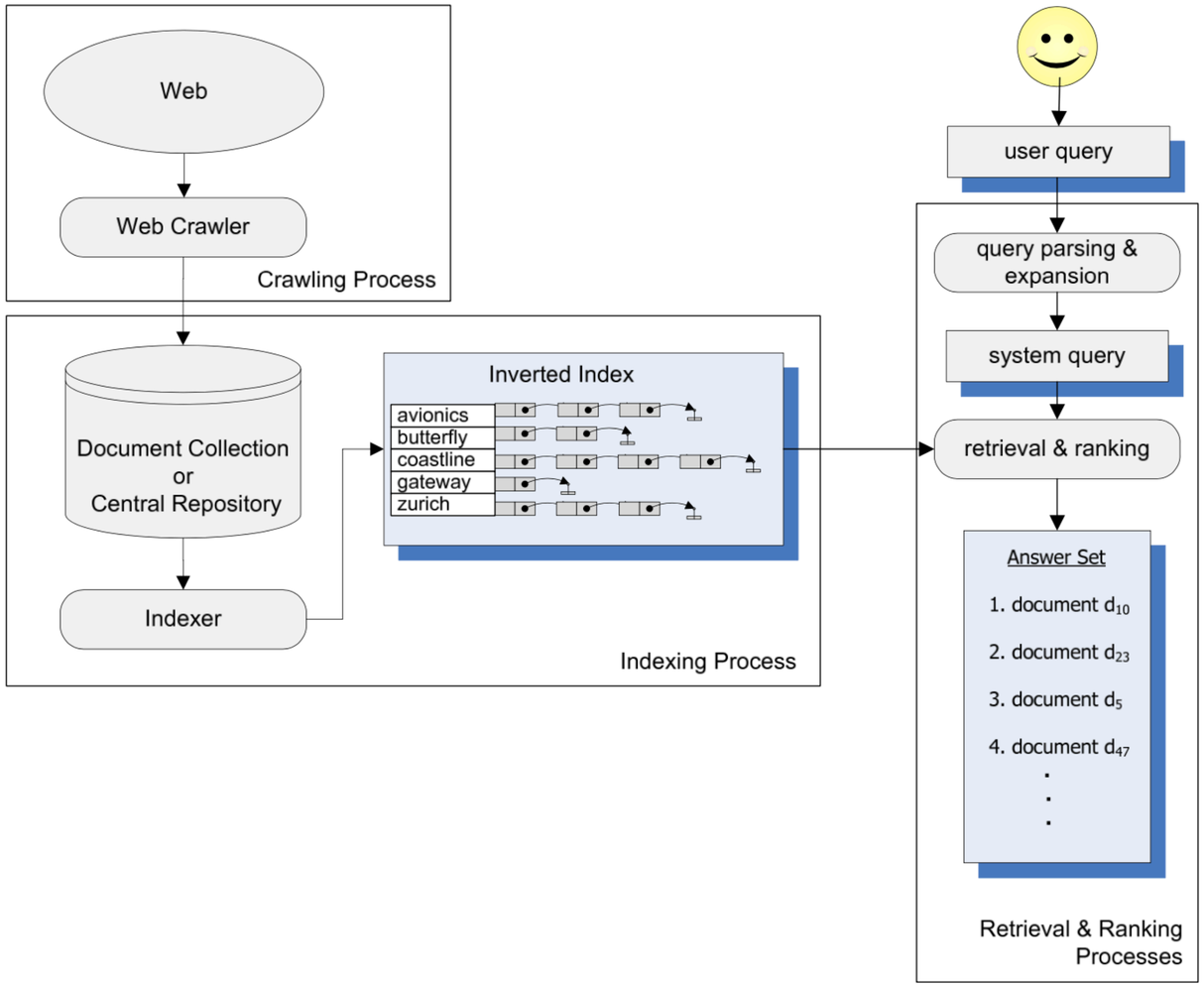


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

Evaluation in Information Retrieval

www.google.com/search?q=latest+eclipse

Google

latest eclipse

Tudo | Imagens | Notícias | Vídeos | Compras | Mais | Ferramentas

Cerca de 295 000 000 resultados (0,33 segundos)

https://www.eclipse.org › downloads · Traduzir esta página

Eclipse Downloads | The Eclipse Foundation

The **Eclipse** Installer 2021-09 R now includes a JRE for macOS, Windows and Linux. **Eclipse**. Get **Eclipse** IDE 2021-09. Install your favorite desktop IDE packages.

- **Eclipse IDE for Java Developers** — The essential tools for any Java developer, including a Java IDE ...
- **Eclipse IDE for Java EE ...** — Tools for Java developers creating Java EE and Web applications, ...
- **Eclipse Installer 2021-09 R** — 5 Steps to Install Eclipse · 1. Download the Eclipse Installer ...
- **Packages** — The Eclipse Installer 2021-09 R now includes a JRE for macOS ...

Mais resultados de eclipse.org »

https://www.eclipse.org › eclipseide · Traduzir esta página

Eclipse IDE 2021-09 | The Eclipse Foundation

LATEST JAVA™ SUPPORT. Supports Java 17, including Pattern Matching for Switch (Preview), Sealed Classes, and more via **Eclipse** Marketplace ...

Enabling Open Innovation & Collaboration | The Eclipse ...

The **Eclipse** Foundation - home to a global community, the **Eclipse** IDE, Jakarta EE and over 375 open source projects, including runtimes, ...

Eclipse Project Downloads

On this page you can find the **latest** builds produced by the **Eclipse** Project. To get started, run the program and go through the user and developer ...

As pessoas também perguntam

- What is the latest version of Eclipse?
- Does Eclipse support Java 17?
- What is the latest version of Eclipse Mars?
- Does Eclipse support Java 15?

Feedback

latest eclipse

latest eclipse

All | Images | Videos | News | Maps | Settings

Portugal · Safe search: moderate · Any time

https://www.eclipse.org › downloads

Eclipse Downloads | The Eclipse Foundation

Eclipse GlassFish provides a complete application server which serves as a compatible implementation for the Jakarta EE specification. **Eclipse** Jetty provides a web server and javax.servlet container. **Eclipse** Jetty provides a web server and javax.servlet container. **Eclipse** Equinox is an implementation of the OSGi core framework specification.

- **Packages** — Eclipse Modeling Tools. 446 MB ; 9,183 DOWNLOADS; The Modeling package...
- **Download 64 Bit** — Download 64 Bit - Eclipse Downloads | The Eclipse Foundation
- **Eclipse Installer** — 5 Steps to Install Eclipse. We've recently introduced the Eclipse...
- **Eclipse Downloads** — Eclipse Downloads - Eclipse Downloads | The Eclipse Foundation
- **Eclipse Desktop & Web IDEs** — The Eclipse IDE is famous for our Java Integrated Development...
- **Projects** — The Eclipse Foundation - home to a global community, the Eclipse IDE,...

https://www.eclipse.org › eclipseide

Eclipse IDE 2021-09 | The Eclipse Foundation

The **Eclipse** Foundation - home to a global community, the **Eclipse** IDE, Jakarta EE and over 375 open source projects, including runtimes, tools and frameworks.

https://www.wizcase.com › download › eclipse-ide

Eclipse IDE Download for Free - 2021 Latest Version

Eclipse, an integrated development environment (IDE) software, is the market leader for Java coding. Its rise in popularity can be attributed to top-notch features, such as autocomplete and auto-correction tools and an easy-to-use workstation.

https://eclipse.gsfc.nasa.gov › lunar.html

NASA - Lunar Eclipse Page

Share Feedback

Evaluation of Information Retrieval Systems

- Evaluation depends on the task, collection, information need type.
- Collections: news articles, web pages, scientific articles, etc.
- Different information needs:
  - What is the homepage of the Dept. of Informatics Engineering at FEUP?
  - Who are the teachers of information systems courses at FEUP?
  - Open source information retrieval frameworks?
- Informational vs. navigational; Number of relevant documents; Answers needed.

Evaluation of Information Retrieval Systems

- Evaluation is at the heart of Information Retrieval.
- Evaluation is important to:
  - Understand the use of a system by its users.
  - Make decisions on new designs and features to implement.
- A primary distinction must be made between <u>effectiveness</u> (*eficácia*) and <u>efficiency</u> (*eficiência*).
  - Effectiveness measures the ability of a search system to find the right information.
  - Efficiency measures how quickly a search system provides an answer.
- User satisfaction encapsulates these and other aspects (ux, coverage, effort, etc).

Information Retrieval System Evaluation

- To measure the effectiveness of a search system in the standard way, we need three things:
  - A document collection;
  - A test suite of information needs, expressible as queries;
  - A set of relevance judgements, typically a binary assessment of either relevant or non-relevant for each query-document pair.
- The standard approach to IR system evaluation revolves around the notion of relevant and non-relevant documents.
- With respect to a user information need, a document in the test collection is given a binary classification as either relevant or non-relevant (gold standard or ground truth).

Information Need

- Relevance is assessed relative to an information need, not a query.
- An information need might be:
  - Information on whether drinking red wine is more effective at reducing your risk of heart attacks than drinking white wine.
- This might be translated into a query such as:
  - [wine red white heart attack effective]
- A document is relevant if it addresses the stated information need, not because it just happens to contain all the words in the query. This distinction is often misunderstood in practice, because the information need is not clear.

The Cranfield Paradigm

- ➔ Evaluation of Information Retrieval systems is the result of early experimentation initiated in the 50's by Cyril Cleverdon.
- ➔ The insights derived from these experiments provide a foundation for the evaluation of IR systems.
- ➔ These experiments culminated in the metrics of Precision and Recall.
- ➔ Cyril Cleverdon introduced the notion of test reference collections, composed of documents, queries, and relevance judgements.
- ➔ Reference collections allows using the same set of documents and queries to evaluate different ranking systems.

Illustration of Cranfield Evaluation Methodology

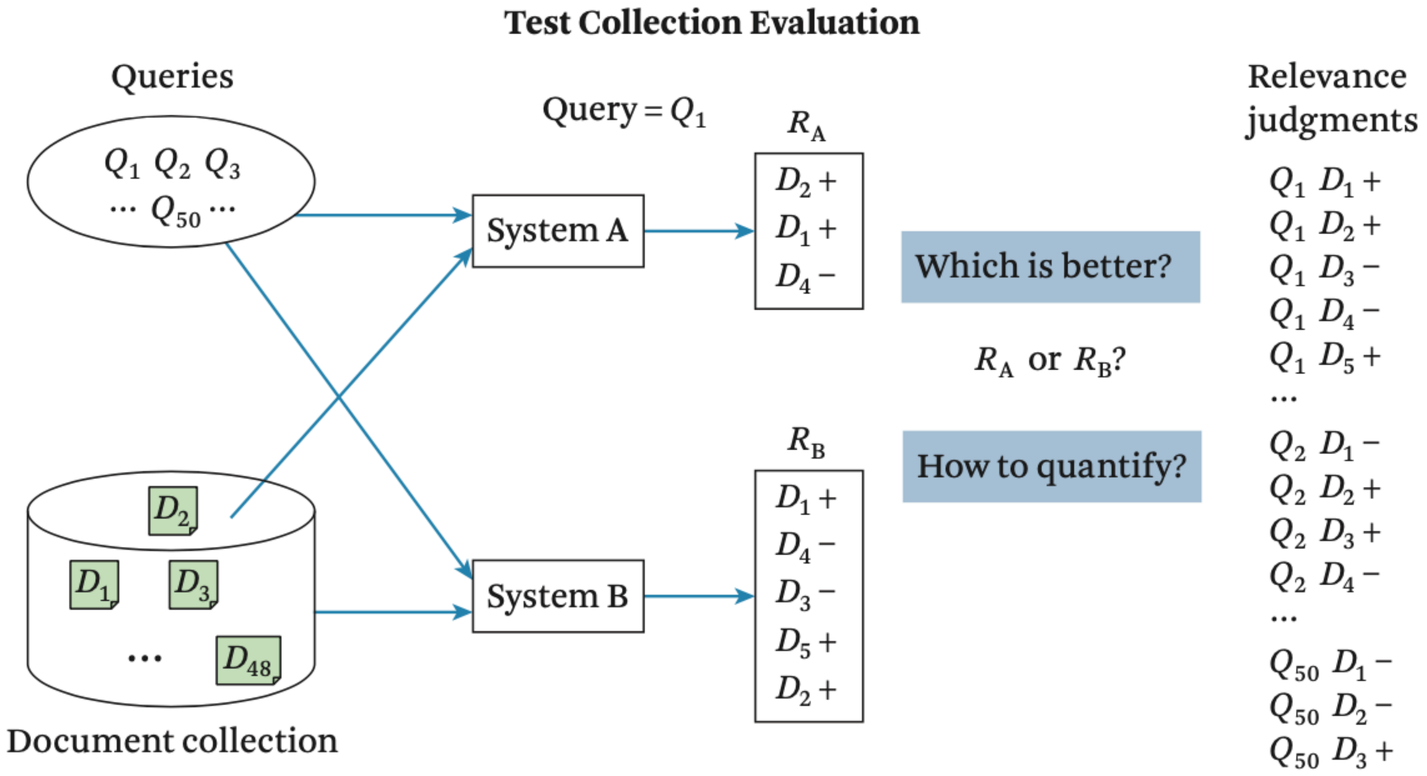


Figure 9.1 **Illustration of Cranfield evaluation methodology.**

From: Zhai, ChengXiang, Sean Massung. Text data management and analysis: a practical introduction to information retrieval and text mining. ACM and Morgan & Claypool, 2016.

TREC Topic Example

<top>
<num> Number: 794

<title> pet therapy

<desc> Description:
How are pets or animals used in therapy for humans and what are the benefits?

<narr> Narrative:
Relevant documents must include details of how pet- or animal-assisted therapy is or has been used. Relevant details include information about pet therapy programs, descriptions of the circumstances in which pet therapy is used, the benefits of this type of therapy, the degree of success of this therapy, and any laws or regulations governing it.

</top>

From: W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009.

Example Test Collections

- ➔ CACM: Titles and abstracts from the Communications of the ACM from 1958–1979. Queries and relevance judgments produced by computer scientists.
- ➔ AP: Associated Press newswire documents from 1988–1990 (from TREC disks 1–3). Queries are the title fields from TREC topics 51–150. Topics and relevance judgments produced by government information analysts.
- ➔ GOV2: Web pages crawled from websites in the .gov domain during early 2004. Queries are the title fields from TREC topics 701–850. Topics and relevance judgments produced by government analysts.

From: W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009.

Example Test Collections

Collection	Number of documents	Size	Average number of words/doc.
CACM	3,204	2.2 MB	64
AP	242,918	0.7 GB	474
GOV2	25,205,179	426 GB	1073

Table 8.1. Statistics for three example text collections. The average number of words per document is calculated without stemming.

From: W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009.

Example Test Collections

Collection	Number of queries	Average number of words/query	Average number of relevant docs/query
CACM	64	13.0	16
AP	100	4.3	220
GOV2	150	3.1	180

Table 8.2. Statistics for queries from example text collections

From: W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009.

Evaluation of Unranked Retrieval

- The two most frequent and basic measures for information retrieval effectiveness are Precision and Recall.
- Precision is the fraction of retrieved documents that are relevant.
  - Precision (P) = #(relevant items retrieved) / #(retrieved items)
- Recall is the fraction of relevant documents that are retrieved.
  - Recall (R) = #(relevant items retrieved) / #(relevant items)
- Precision and Recall are set-based measures.

Precision and Recall

Not retrieved for q1

Results for query q1

#1 #2 #3 #4 #5 #6

Precision

P(q1) = 4 relevant documents retrieved / 6 documents retrieved = 0.67

Recall

R(q1) = 4 relevant docs retrieved / 8 existing relevant docs = 0.5

Contingency Table

	relevant	**not relevant**
retrieved	true positives (tp)	false positives (fp)
not retrieved	false negatives (fn)	true negatives (tn)

- Precision = true positives / (true positives + false positives)
- Recall = true positives / (true positives + false negatives)

- Accuracy, the fraction of classifications that are correct (not useful for IR).
  - Accuracy = #(true positives + true negatives) / #(tp + fp + fn + tn)

F measure

- A measure that trades-off Precision versus Recall is the F measure (or F score), which is the weighted harmonic mean of precision and recall.

- By default a balanced harmonic mean is used (α = 1/2) resulting in a balanced F measure defined by:

$$F = (1 + \beta^2) \times \frac{P \times R}{\beta^2 \times P + R} \qquad F_{\beta=1} = \frac{2PR}{P + R}$$

- It is possible to emphasize Precision ($\beta < 1$) or Recall ($\beta > 1$).

- Often criticized because it is opaque to interpretation.

Evaluation of Ranked Retrieval

- ➔ Precision and Recall are computed over unordered sets of documents.
- ➔ These measures need to be extended to evaluate the ranked lists of results common in search engines.
- ➔ In ranked retrieval contexts, appropriate sets of retrieved documents are naturally given by the top k retrieved documents. For each set, Precision and Recall values can be computed.
- ➔ These values can be plotted to obtain a precision-recall curve.

Precision-Recall Curves

- Consider the ordered set of relevant (R) and non-relevant (N) results from a search system A:
  - S_a = R R N R N N R N R N
- In this ranking, the first result is relevant and corresponds to 20% of all (available) relevant documents.
  - We say that we have 100% precision at 20% recall.
- At position 4, three documents out of four are relevant, and three documents of a total of five relevant document have been retrieved.
  - We say that we have 75% precision at 60% recall.
- Using this data, we can plot a precision-recall curve.

Precision-Recall Curves

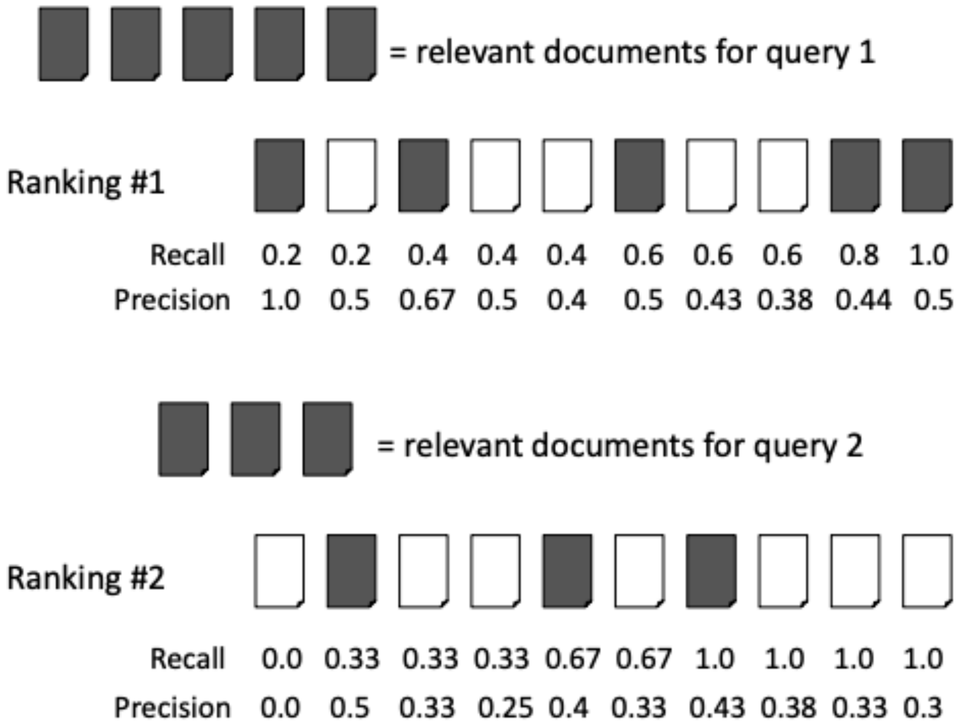


Fig. 8.3. Recall and precision values for rankings from two different queries

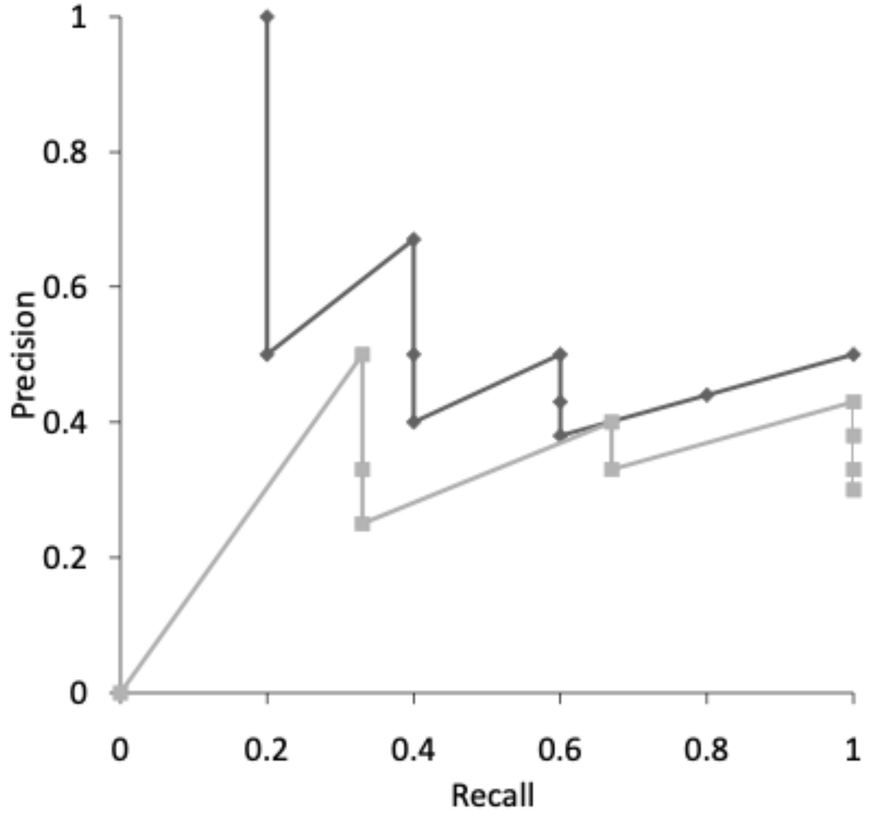


Fig. 8.4. Recall-precision graphs for two queries

From: W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009.

Precision-Recall Curves (interpolated)

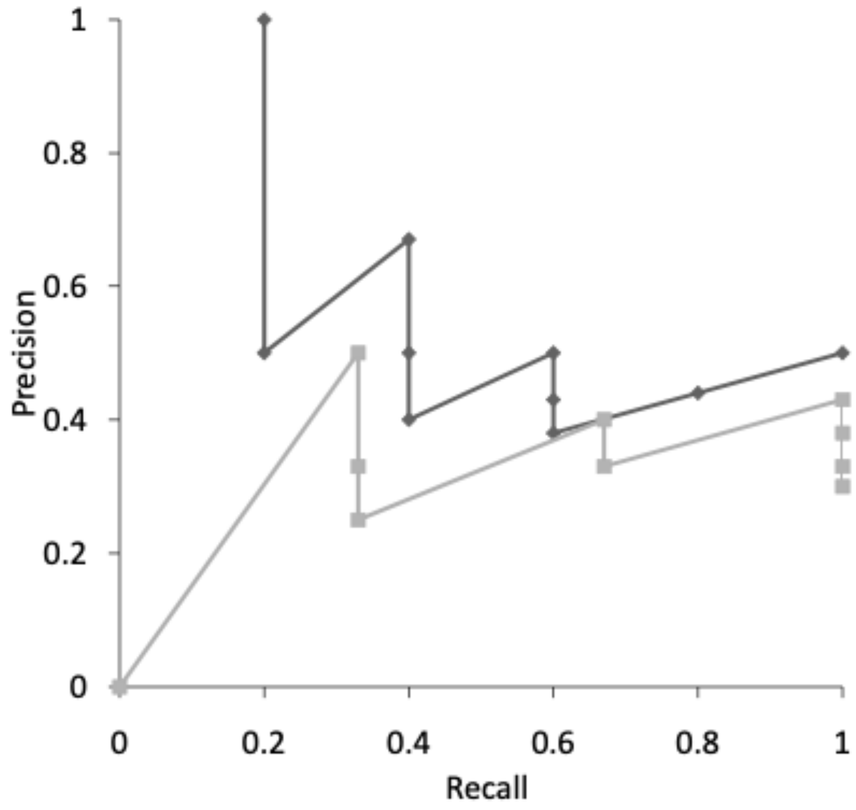


Fig. 8.4. Recall-precision graphs for two queries

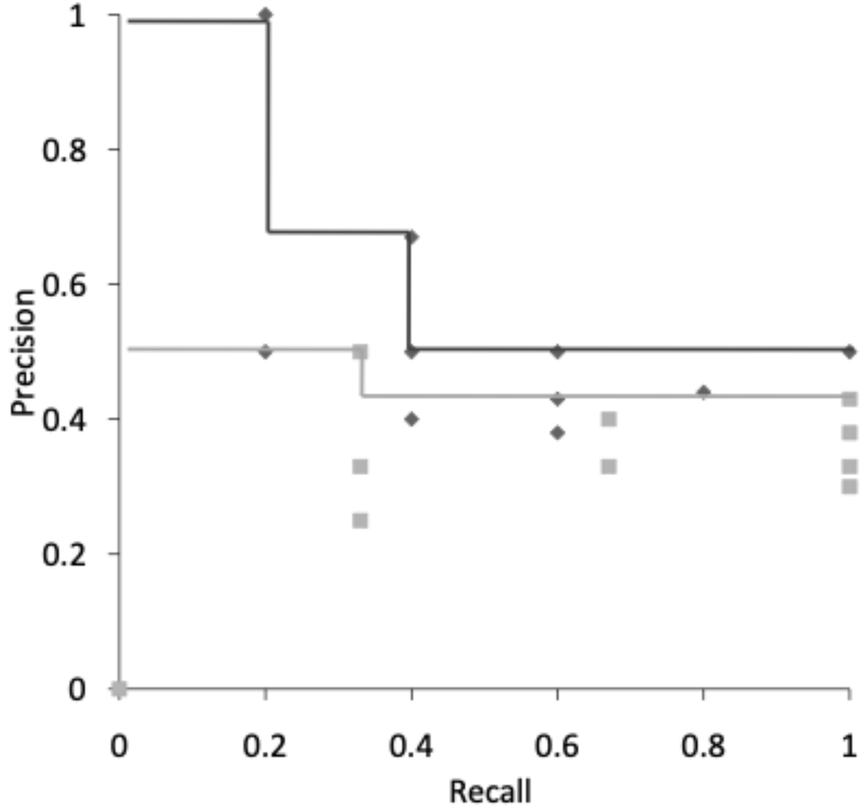


Fig. 8.5. Interpolated recall-precision graphs for two queries

From: W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009.

Comparing Systems

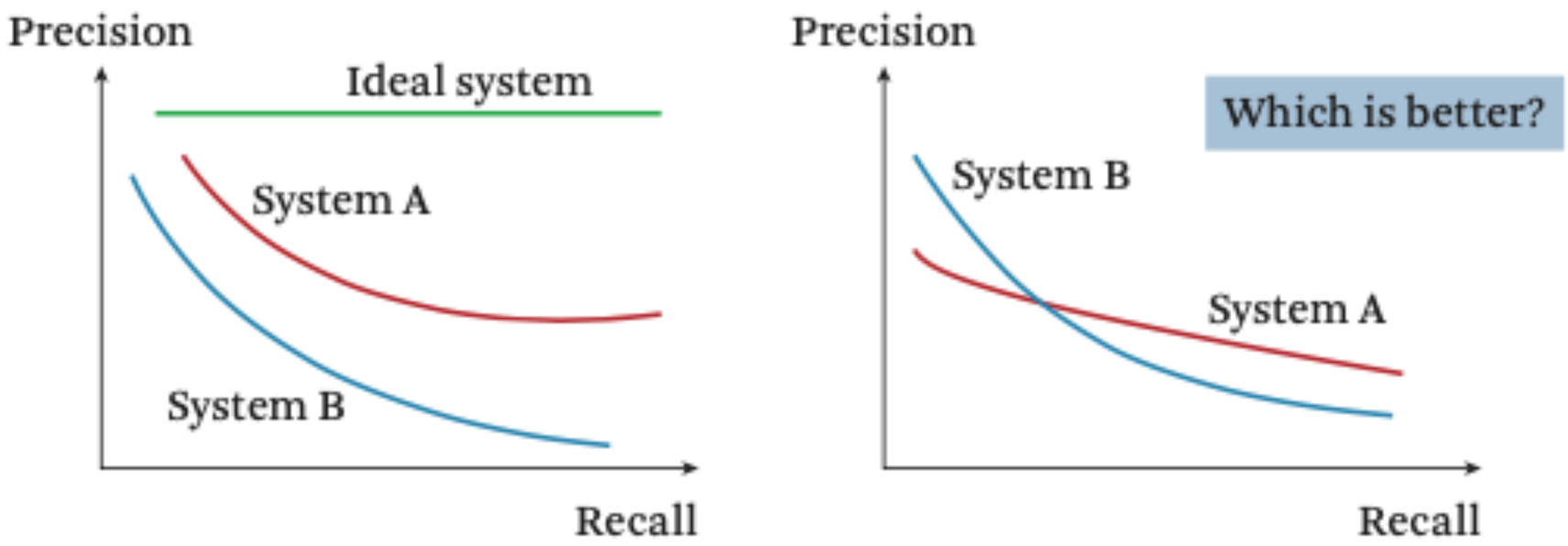


Figure 9.5 **Comparison of two PR curves.** (Courtesy of Marti Hearst)

From: Zhai, ChengXiang, Sean Massung. Text data management and analysis: a practical introduction to information retrieval and text mining. ACM and Morgan & Claypool, 2016.

Precision at k (P@k)

- In the case of web search, the majority of users do not require high recall.
- What matters are high quality results on the first page. This leads to measuring precision at fixed low levels of retrieved results.
- For example, "precision at 5" (P@5) or "precision at 10" (P@10).
- Considering the following ranking for a given query:
  - R R N N R N R R R R
- P@5 = 0.6; P@10 = 0.7

Mean Average Precision

- Average Precision (AvP) provides a single-figure measure of quality across recall levels for a single query.
- For a single information need, average precision is the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved.
- Given a set of queries, the Mean Average Precision (MAP) is the mean over the AvP values. This is one of the most commonly used measures in IR.

Average Precision

➔ Ranking #1

	X		**X**	**X**	**X**	**X**				**X**
R	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
P	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

➔ Ranking #2

		X			**X**	**X**	**X**		**X**	**X**
R	0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
P	0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

➔ AvP (R#1) = (1 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6) / 6 = 0.78

➔ AvP (R#2) = (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6) / 6 = 0.52

➔ MAP = (0.78 + 0.52) / 2 = 0.65

User-Centered and Online Evaluation

- Instead of using standard collection, systems may be evaluated using real user interaction data. This requires a working system or prototype with a user base.
- In user-centered evaluation, different techniques may be used
  - Observational studies, observe the user in controlled sessions;
  - Observe server-side query logs, to measure performance;
  - Use client-side logging tools, to study interaction patterns;
- Online, large-scale evaluation experiments can be made with A/B tests. Common practice in industry - run multiple experiments with different versions of the system.

Measuring Efficiency

Example Efficiency Metrics

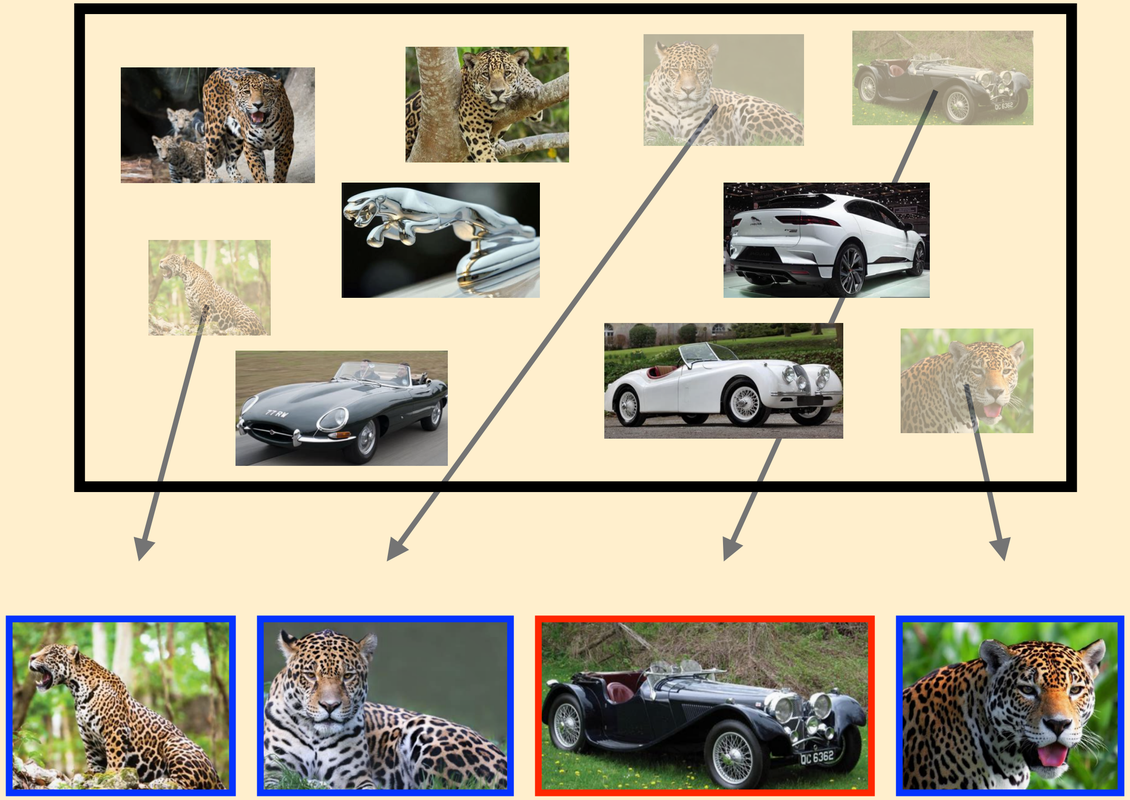
Metric name	Description
Elapsed indexing time	Measures the amount of time necessary to build a document index on a particular system.
Indexing processor time	Measures the CPU seconds used in building a document index. This is similar to elapsed time, but does not count time waiting for I/O or speed gains from parallelism.
Query throughput	Number of queries processed per second.
Query latency	The amount of time a user must wait after issuing a query before receiving a response, measured in milliseconds. This can be measured using the mean, but is often more instructive when used with the median or a percentile bound.
Indexing temporary space	Amount of temporary disk space used while creating an index.
Index size	Amount of storage necessary to store the index files.

Table 8.5. Definitions of some important efficiency metrics

Exercises

Calculate Precision and Recall

➔ Consider the following scenario and a search for [jaguar] (the animal).
What is the precision and recall of this system?



	relevant	not
retrieved	3	1
not	2	4

Retrieved documents

Calculate Precision and Recall

	relevant	not
retrieved	3	1
not	2	4

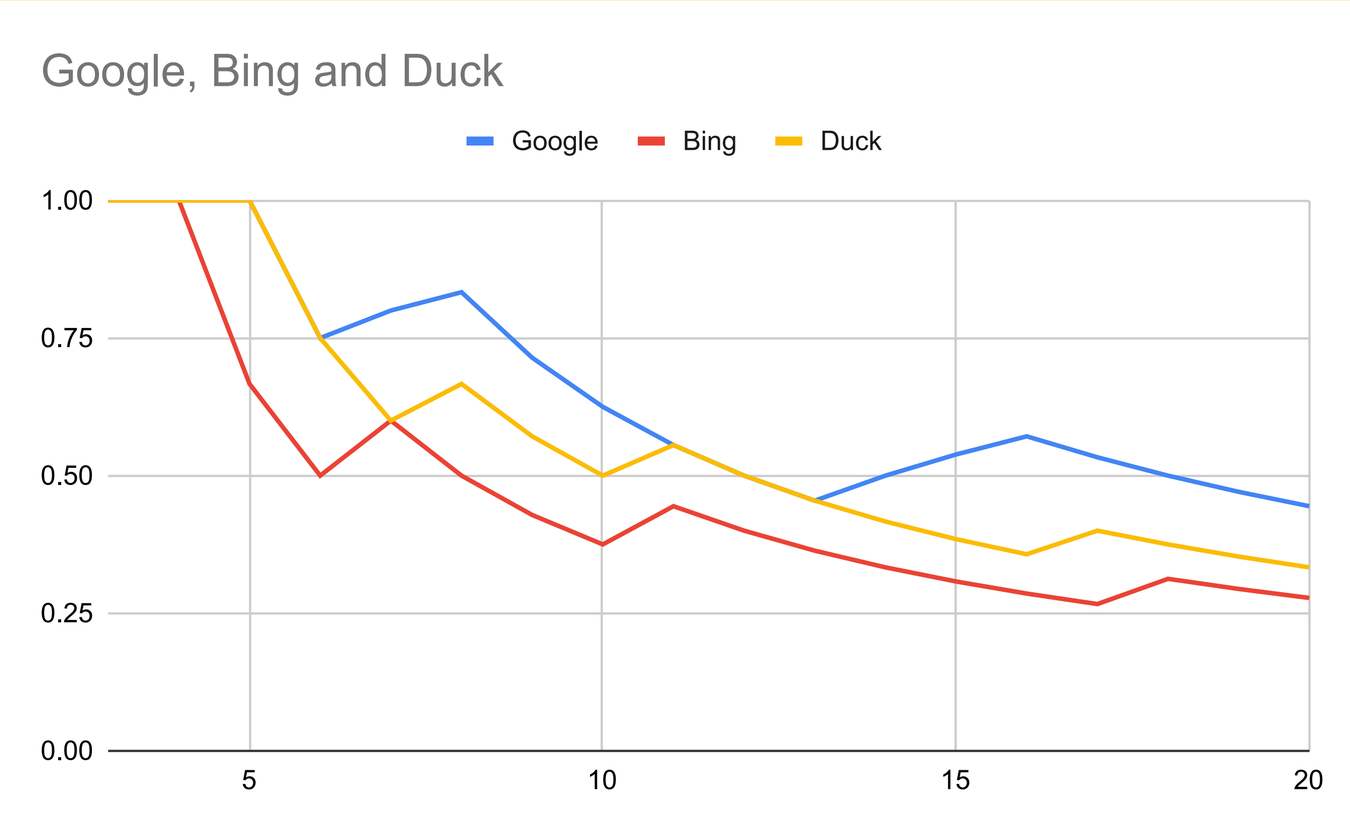
- ➔ Precision = #(relevant items retrieved) / #(retrieved items) = 3 / 4 = 0.75
- ➔ Recall = #(relevant items retrieved) / #(relevant items) = 3 / 5 = 0.6

Calculate Precision and Recall

- Compare Google, Bing, DuckDuckGo with the following queries:
  - [eclipse]
  - [eclipse -lunar]
- For these two queries, for each search engine,
  and considering "eclipse the editor" the underlying information need:
  - Plot P@ values (x-axis) for each search engine for the first 20 results returned. Use a spreadsheet.
  - Calculate AvP for each query for each search engine.
  - Calculate MAP values for each search engine.

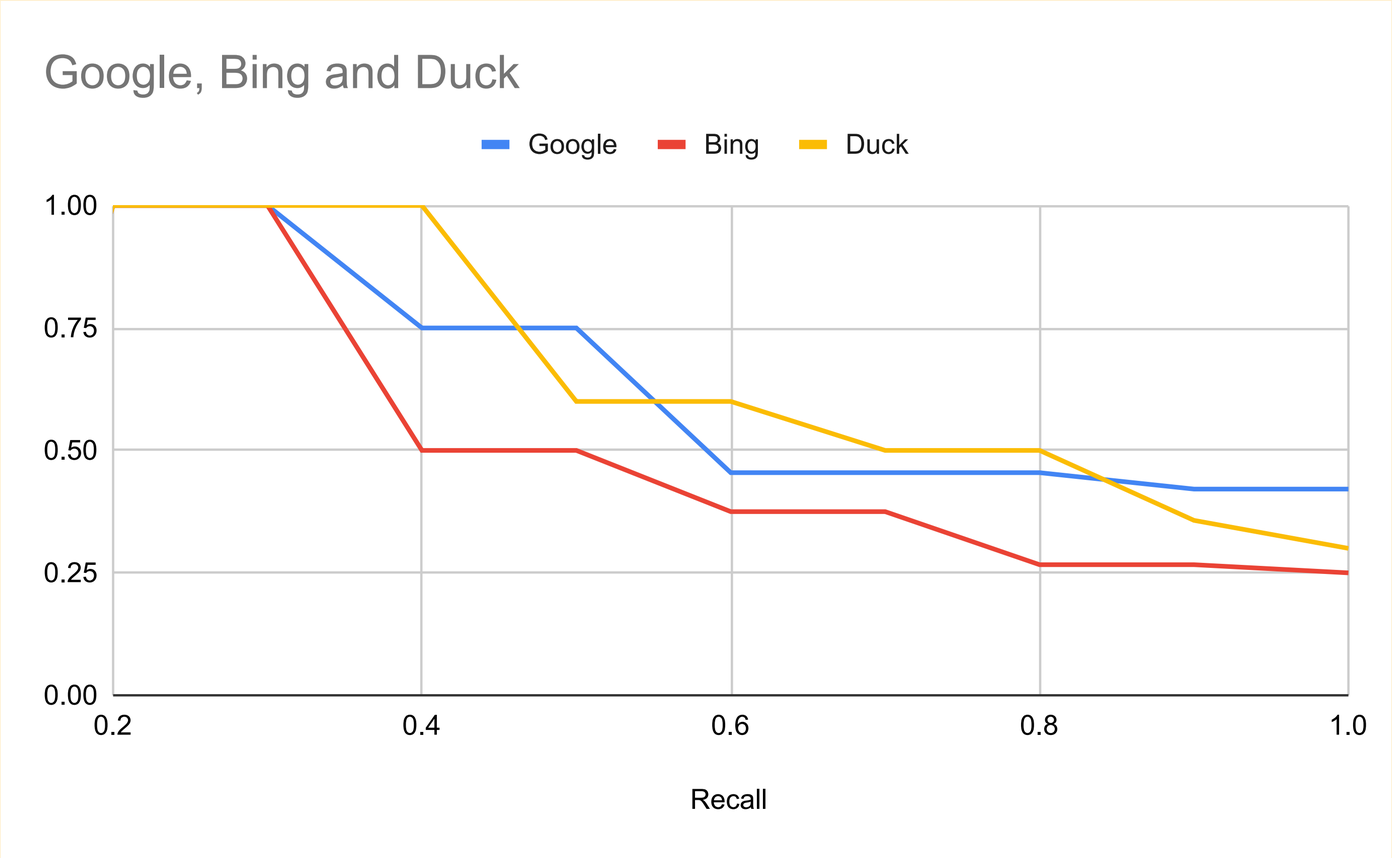
P@ Plot

➔ Q = [eclipse]



Precision-Recall Graph

➔ Q = [eclipse]



Average Precision Values

➔ Q = [eclipse]

Rank	Google		Bing		Duck	
1	**x**	1/1=1	**x**	1	**x**	1
2	**x**	2/2=1			**x**	1
3					**x**	1
4			**x**	2/4=0.5		
5			**x**	3/5=0.6		
6					**x**	4/6=0.67
7	**x**	3/7=0.43				
8						
9			**x**	4/9=0.44	**x**	5/9=0.56
10						
AP		0.81		0.635		0.846

Next Steps

- Work on Milestone #2
  - Define information needs
  - Index documents
  - Retrieve documents — explore different rankings

Information Retrieval Overview

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Chapters 1, 2, 6, 11, 12 from Introduction to Information Retrieval, Manning, C. et al. (2008)
Based on Chapter 3 from Modern Information Retrieval, Baeza-Yates et al. (2011)

PRI Status

- Lecture: IR techniques and models.
- Lab:
  - 1. Solr practice (last week);
  - 2. Indexing and retrieval of project collection (**this week**);
  - 3. Evaluation of results (*two weeks* from now);
  - 4. M2 workshop.
- Next week: FEUP week / no classes.

Background Concepts

Incidence Matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	…
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
…							

Figure 1.1 A term-document incidence matrix. Matrix element (t, d) is 1 if the play in column d contains the word in row t, and is 0 otherwise.

Boolean Model

- In the Boolean Retrieval Model queries are represented in the form of a Boolean expression of terms.
- E.g.: [Brutus AND Caesar AND NOT Calpurnia]
  - => 110100 AND 110111 AND 101111 = 100100
- The model views each document as a set of words.
- Bag of Words (BoW) model, where the exact ordering of terms in a document is ignored and only their presence is considered.

Inverted Index

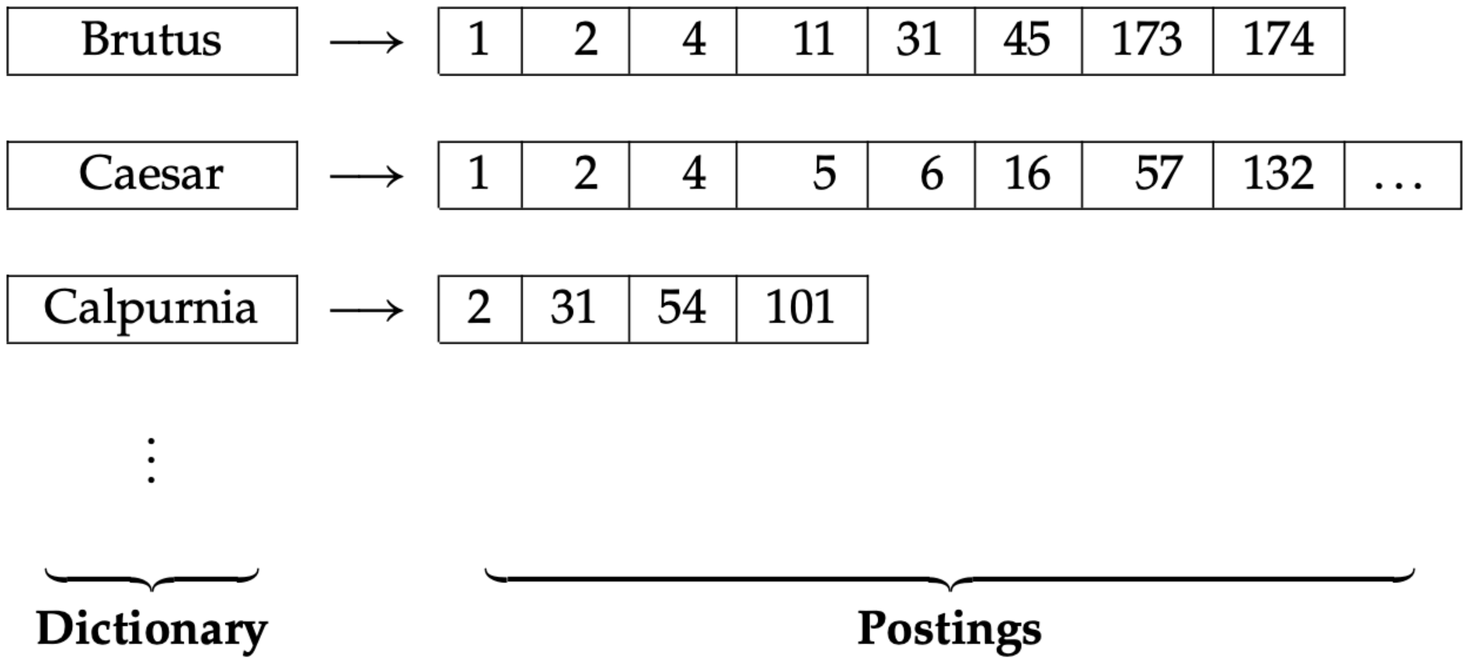


Figure 1.3 The two parts of an inverted index. The dictionary is commonly kept in memory, with pointers to each postings list, which is stored on disk.

Inverted Index (Basic)

Vocabulary	n_i	Occurrences as inverted lists
to	2	[1,4],[2,2]
do	3	[1,2],[3,3],[4,3]
is	1	[1,2]
be	4	[1,2],[2,2],[3,2],[4,2]
or	1	[2,1]
not	1	[2,1]
I	2	[2,2],[3,2]
am	2	[2,2],[3,1]
what	1	[2,1]
think	1	[3,1]
therefore	1	[3,1]
da	1	[4,3]
let	1	[4,2]
it	1	[4,2]

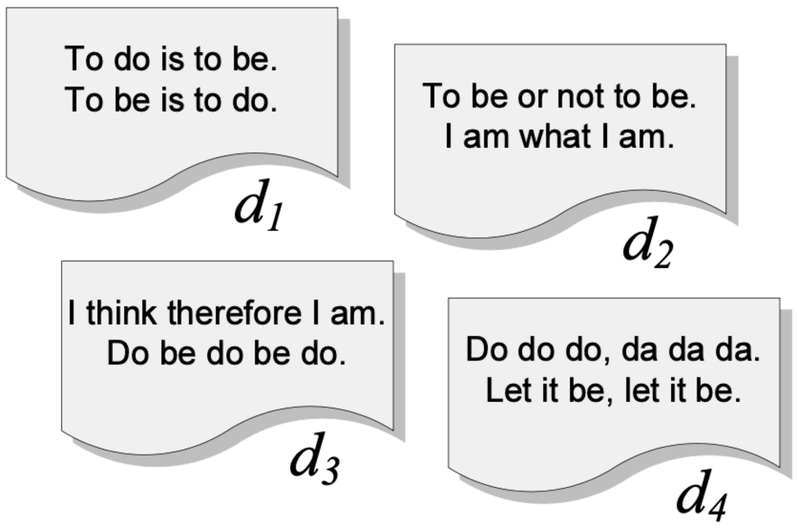


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

Inverted Index with Positional Information

Vocabulary	n_i	Occurrences as full inverted lists
to	2	[1,4,[1,4,6,9]],[2,2,[1,5]]
do	3	[1,2,[2,10]],[3,3,[6,8,10]],[4,3,[1,2,3]]
is	1	[1,2,[3,8]]
be	4	[1,2,[5,7]],[2,2,[2,6]],[3,2,[7,9]],[4,2,[9,12]]
or	1	[2,1,[3]]
not	1	[2,1,[4]]
I	2	[2,2,[7,10]],[3,2,[1,4]]
am	2	[2,2,[8,11]],[3,1,[5]]
what	1	[2,1,[9]]
think	1	[3,1,[2]]
therefore	1	[3,1,[3]]
da	1	[4,3,[4,5,6]]
let	1	[4,2,[7,10]]
it	1	[4,2,[8,11]]

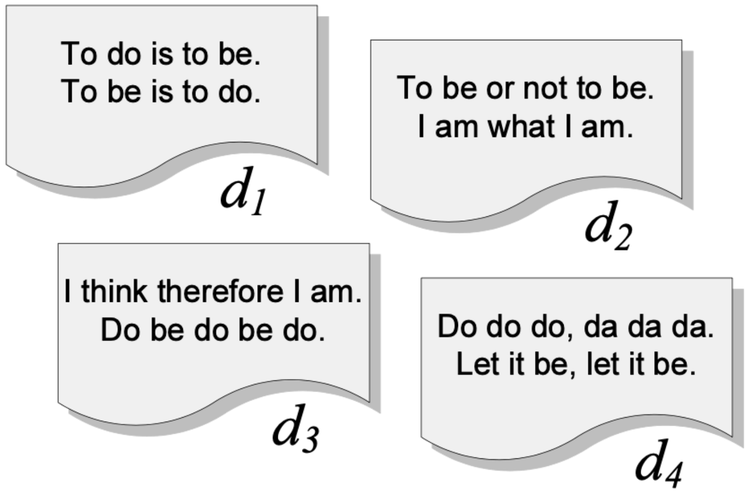


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

Index Construction

- Choosing the document unit for indexing and the index granularity are important first steps.

- There is a precision / recall tradeoff in this decision.

  - "If the units get too small (e.g. sentences), we are likely to miss important passages because terms were distributed over several mini-documents, whereas if units are too large (e.g. books) we tend to get spurious matches and the relevant information is hard for the user to find."

Tokenization

- "Given a character sequence and a defined document unit, tokenization is the task of chopping it into pieces, called tokens."

Input: Friends, Romans, Countrymen, lend me your ears;

Output: | Friends | Romans | Countrymen | lend | me | your | ears |

- "A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing."

- "A type is the class of all tokens containing the same character sequence."

Stop Words

- ➔ Stop words are words considered of little value in helping select documents in the retrieval process, e.g. a word that exists in all documents.
- ➔ The strategy to determine stop words is by looking the collection frequency of a term (*cf*), i.e. the total number of times a term appears in the document collection.
- ➔ A stop list is a (commonly) hand-picked list of terms to be discarded during the indexing process.

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Figure 2.5 A stop list of twenty-five semantically nonselective words that are common in Reuters-RCV1.

Token Normalization

- "Token normalization is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens."
- Some forms of normalization commonly employed are: accents, capitalization, and stemming and lemmatization for dealing with different forma of a word (e.g. watch, watches, watching).
- Stemming refers to a heuristic process that chops off the ends of words in the hope of reducing inflectional forms.
- Lemmatization refers to the process of reducing inflectional forms by using vocabularies and the morphological analysis of words to find its lemma.

From full text to index terms

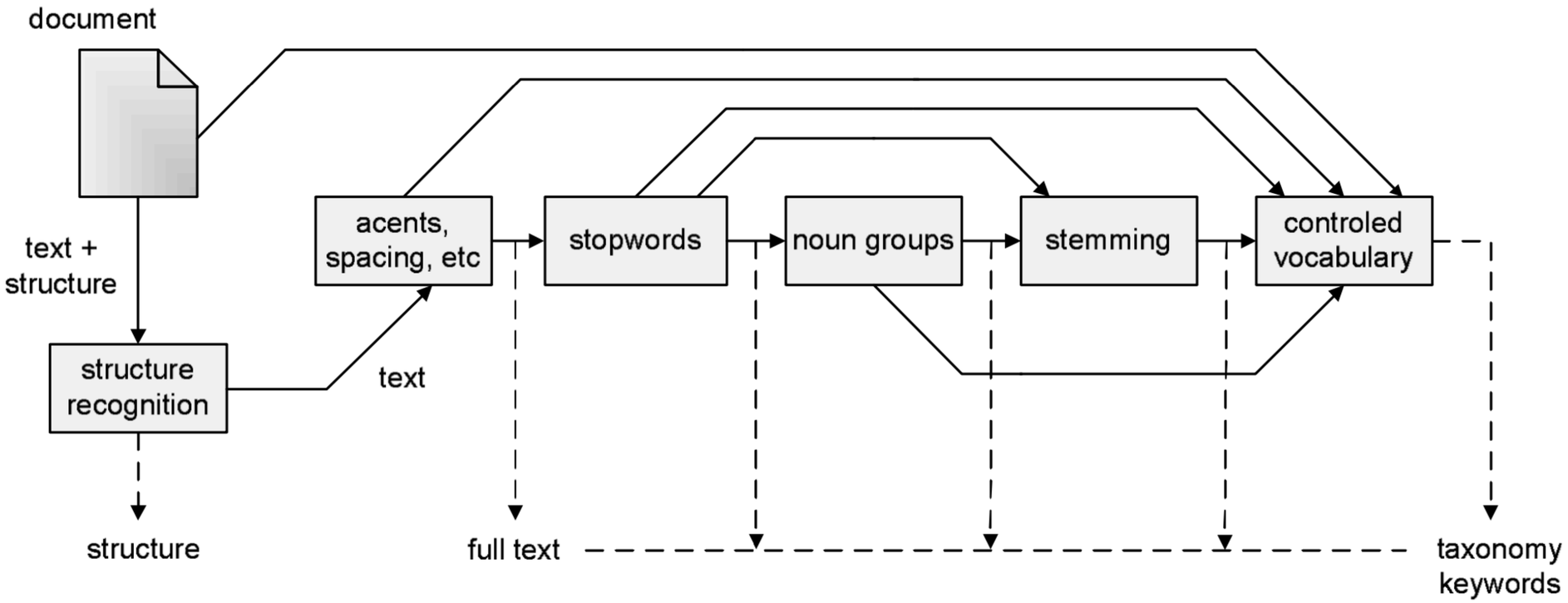
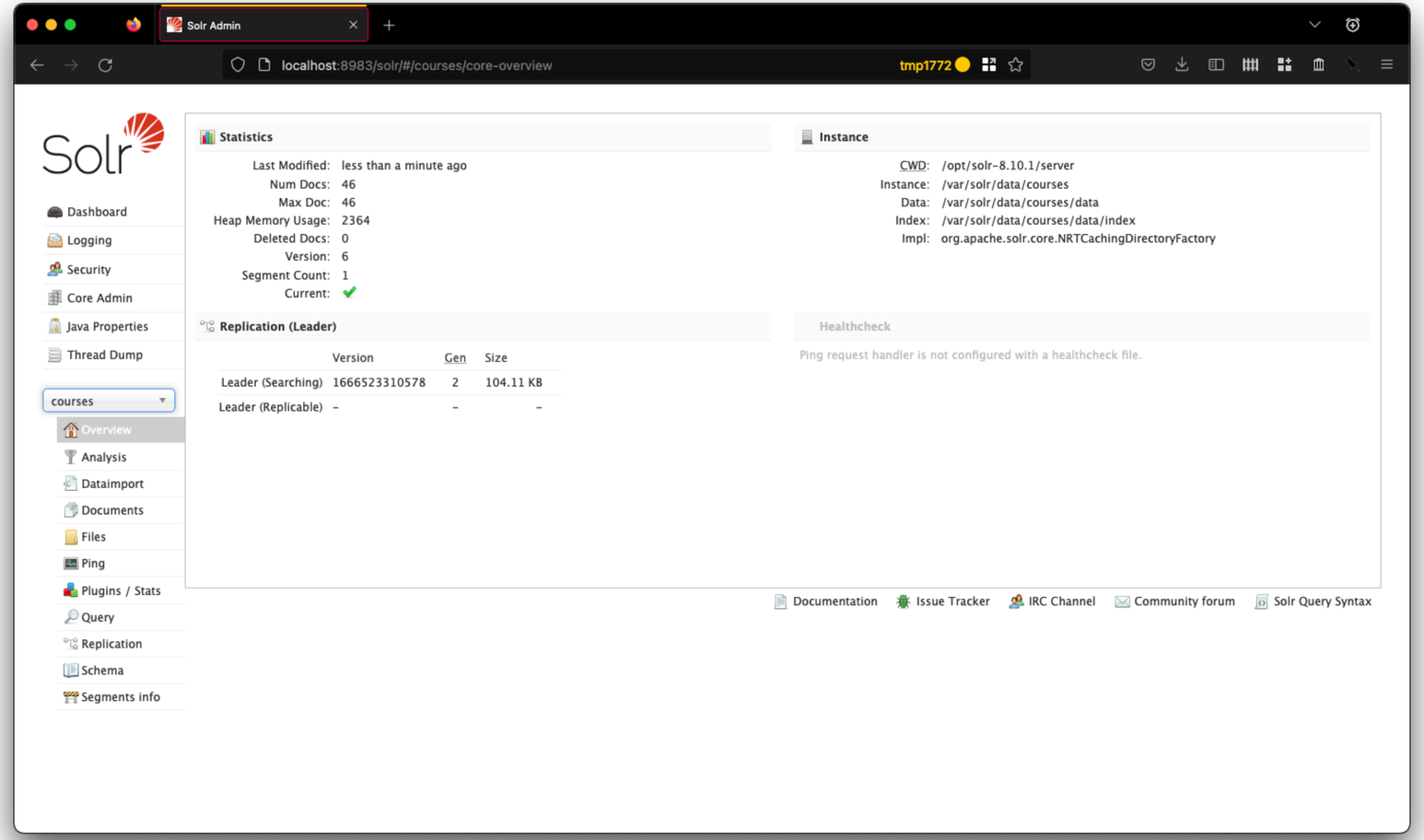
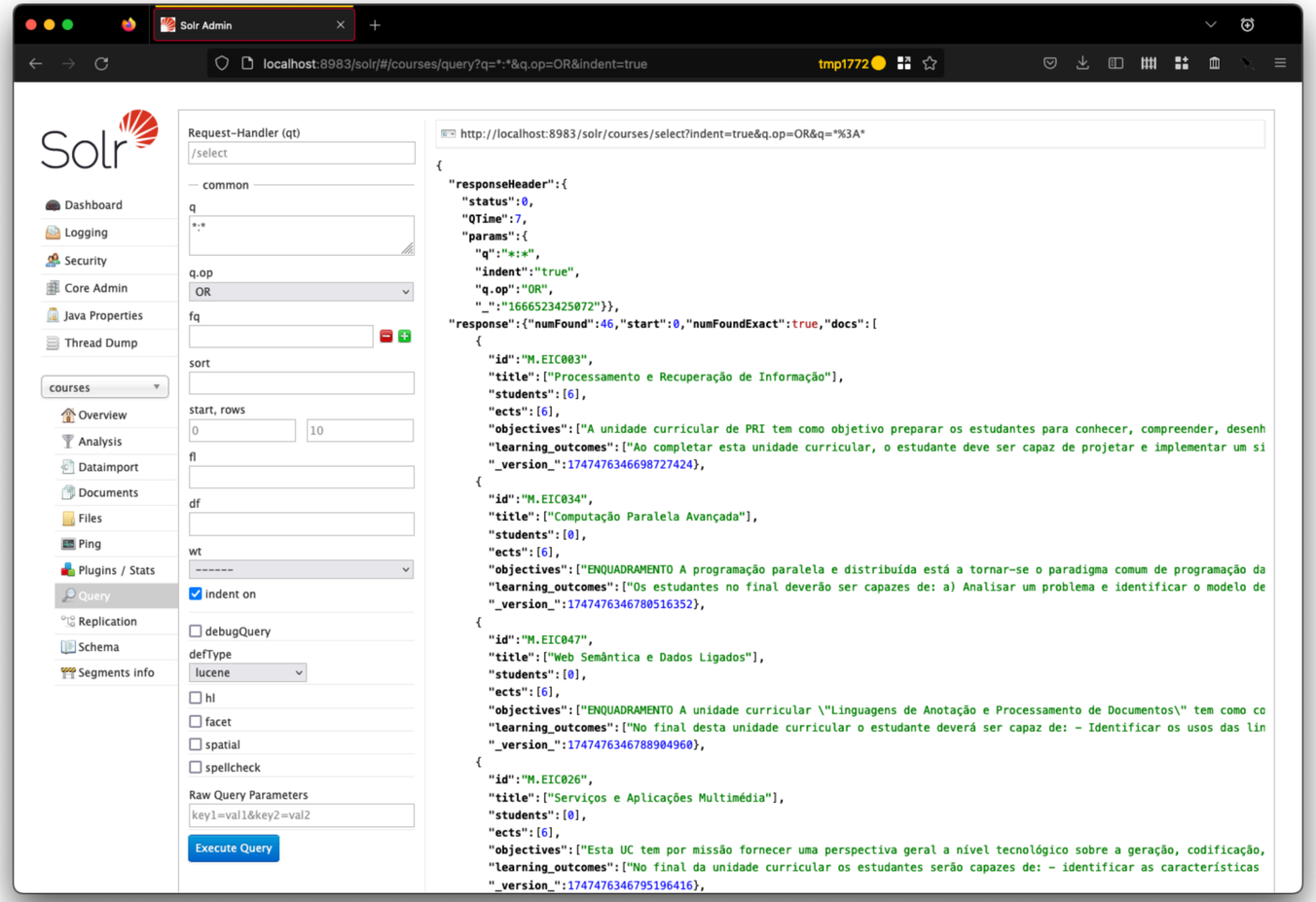


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

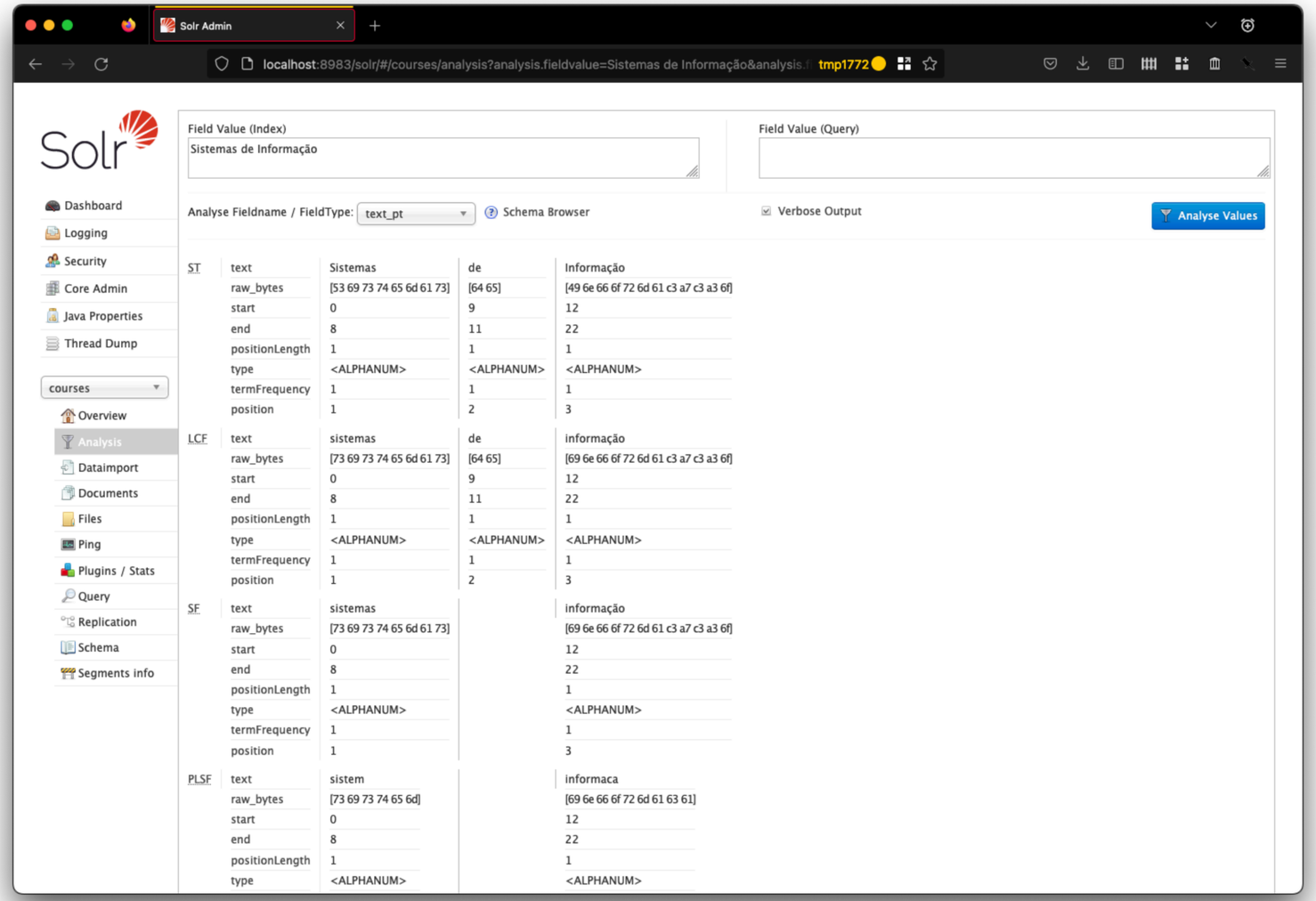
Solr Example



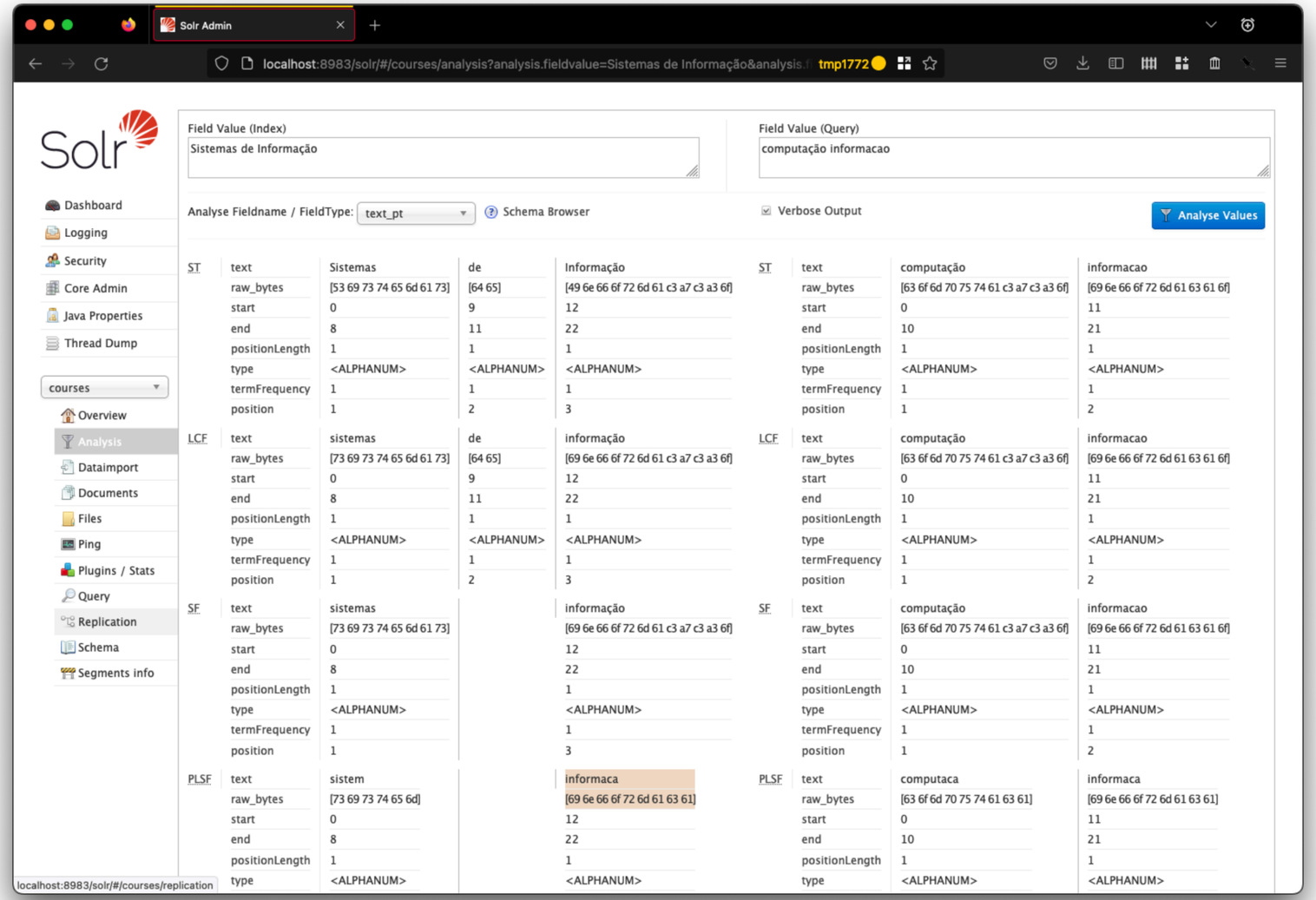
Solr Query Tool



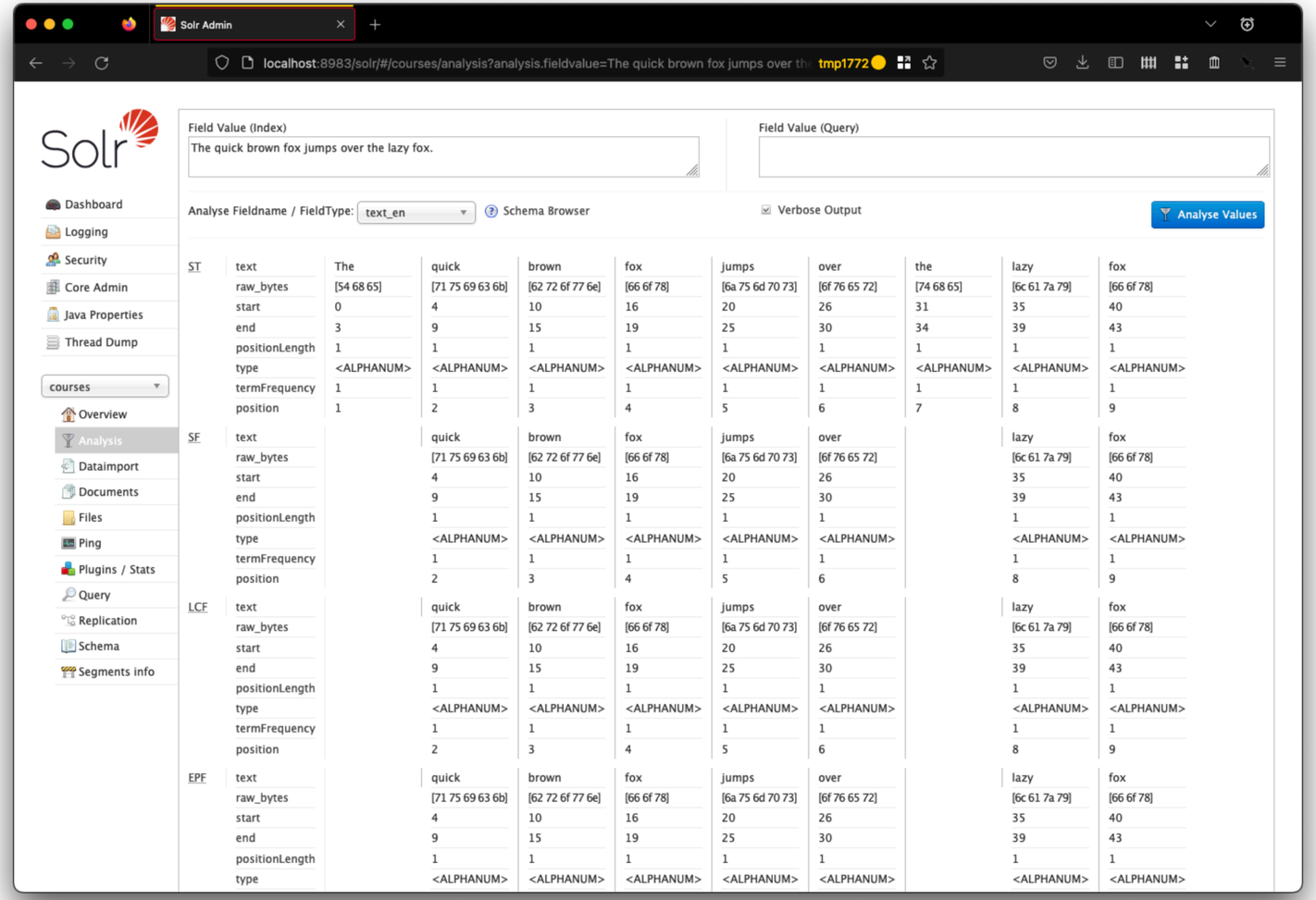
Solr Analysis Screen (pt)



Solr Match


Solr Admin
localhost:8983/solr/#/courses/analysis?analysis.fieldvalue=Sistemas de Informação&analysis.f
tmp1772
Solr
Dashboard
Logging
Security
Core Admin
Java Properties
Thread Dump
courses
Overview
Analysis
Dataimport
Documents
Files
Ping
Plugins / Stats
Query
Replication
Schema
Segments info
Field Value (Index)
Sistemas de Informação
Field Value (Query)
computação informacao
Analyse Fieldname / FieldType: text_pt
Schema Browser
Verbose Output
Analyse Values
ST text Sistemas de Informação
raw_bytes [53 69 73 74 65 6d 61 73] [64 65] [49 6e 66 6f 72 6d 61 c3 a7 c3 a3 6f]
start 0 9 12
end 8 11 22
positionLength 1 1 1
type <ALPHANUM> <ALPHANUM> <ALPHANUM>
termFrequency 1 1 1
position 1 2 3
LCF text sistemas de informação
raw_bytes [73 69 73 74 65 6d 61 73] [64 65] [69 6e 66 6f 72 6d 61 c3 a7 c3 a3 6f]
start 0 9 12
end 8 11 22
positionLength 1 1 1
type <ALPHANUM> <ALPHANUM> <ALPHANUM>
termFrequency 1 1 1
position 1 2 3
SF text sistemas informação
raw_bytes [73 69 73 74 65 6d 61 73] [69 6e 66 6f 72 6d 61 c3 a7 c3 a3 6f]
start 0 12
end 8 22
positionLength 1 1
type <ALPHANUM> <ALPHANUM>
termFrequency 1 1
position 1 3
PLSF text sistem informaca
raw_bytes [73 69 73 74 65 6d] [69 6e 66 6f 72 6d 61 63 61]
start 0 12
end 8 22
positionLength 1 1
type <ALPHANUM> <ALPHANUM>
ST text computação informacao
raw_bytes [63 6f 6d 70 75 74 61 c3 a7 c3 a3 6f] [69 6e 66 6f 72 6d 61 63 61 6f]
start 0 11
end 10 21
positionLength 1 1
type <ALPHANUM> <ALPHANUM>
termFrequency 1 1
position 1 2
LCF text computação informacao
raw_bytes [63 6f 6d 70 75 74 61 c3 a7 c3 a3 6f] [69 6e 66 6f 72 6d 61 63 61 6f]
start 0 11
end 10 21
positionLength 1 1
type <ALPHANUM> <ALPHANUM>
termFrequency 1 1
position 1 2
SF text computação informacao
raw_bytes [63 6f 6d 70 75 74 61 c3 a7 c3 a3 6f] [69 6e 66 6f 72 6d 61 63 61 6f]
start 0 11
end 10 21
positionLength 1 1
type <ALPHANUM> <ALPHANUM>
termFrequency 1 1
position 1 2
PLSF text computaca informaca
raw_bytes [63 6f 6d 70 75 74 61 63 61] [69 6e 66 6f 72 6d 61 63 61]
start 0 11
end 10 21
positionLength 1 1
type <ALPHANUM> <ALPHANUM>
localhost:8983/solr/#/courses/replication

Solr Analysis Screen (eng)



Solr Tokenizers and Filters

- **Tokenizers**
  https://solr.apache.org/guide/solr/latest/indexing-guide/tokenizers.html
  - *"Tokenizers are responsible for breaking field data into lexical units, or tokens."*
  - E.g. standard tokenizer, classic tokenizer, letter tokenizer, n-gram

- **Filters**
  https://solr.apache.org/guide/solr/latest/indexing-guide/filters.html
  - *"Filters examine a stream of tokens and keep them, transform them, or discard them depending on the filter type being used."*
  - E.g. stop filter, synonym filter (numbers), porter stem filter.

Term Weighting

Ranked Retrieval

- ➔ With the Boolean model, a document either matches or does not match a query.
- ➔ In large document collections this is not feasible, thus it is essential for a search system to rank-order the documents.
- ➔ Note that there are scenarios where recall is determinant — i.e., all documents need to be analyzed (e.g., patent search), and thus Boolean search is used.

Parametric and Zone Indexes

- ➔ Although we have considered documents to be a simple sequence of terms, most documents have additional structure (e.g. email message). Additionally, metadata is often associated with a document (e.g. date, authors, title).

- ➔ Parametric indexes are inverted indexes built for specific parameters, or fields, that support parametric search (e.g. "all documents from author Z containing word Y").

- ➔ Zones are a similar concept applied to arbitrary free text (e.g. portion of a document). For example, a document's abstract can be associated to a specific zone index.

Zone Indexes

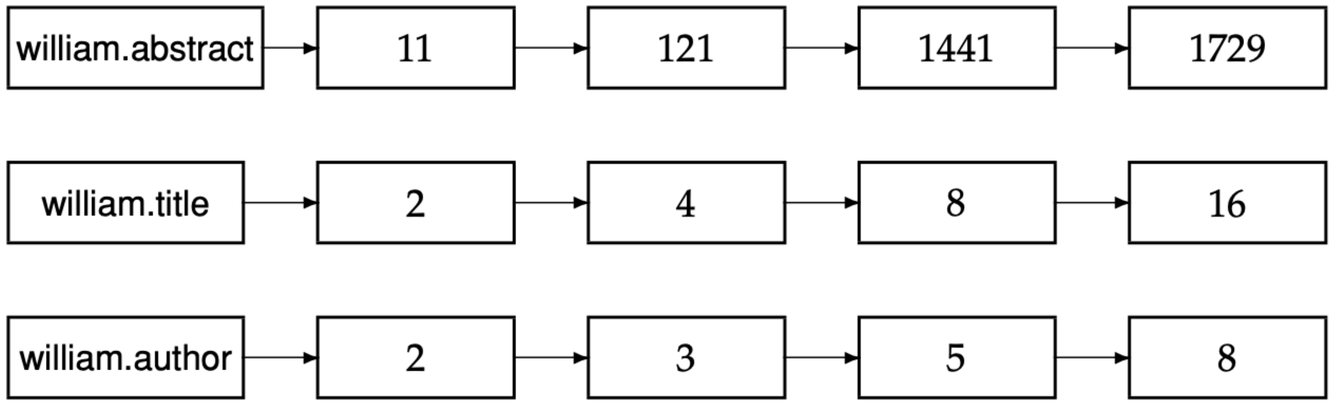


Figure 6.2 Basic zone index; zones are encoded as extensions of dictionary entries.

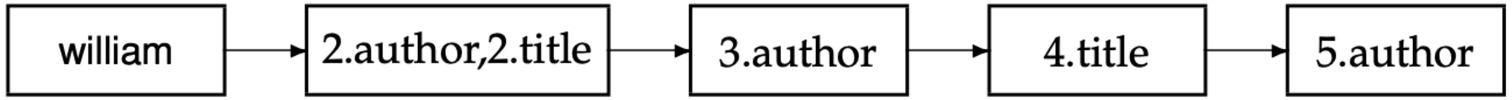


Figure 6.3 Zone index in which the zone is encoded in the postings rather than the dictionary.

Ranked Boolean Retrieval

- ➔ Zones (or fields) can be weighted differently to compute each document's relevance simply using a linear combination of zone scores, where each zone of the document contributes a Boolean value.

- ➔ Zone weights can be specified by an expert (commonly the end user) but are usually learned by the system based on training examples. This method is known as "machine-learned relevance" or "learning to rank".

Term Frequency (tf)

- ➔ The next step is not to consider only the presence (or not) of a query term in a zone, but the number of mentions of the term (i.e., term frequency).
- ➔ The simplest form of weighting terms differently is simply to assign the weight of a term to the term's frequency.
- ➔ The term frequency of a term in a document is denoted $tf_{t,d}$.
- ➔ In the bag of words model, the ordering is ignored but the number of occurrences of each term is key (in contrast with Boolean retrieval).

Document Frequency (df)

- Raw term frequency suffers from a problem: all terms are considered equally important when assessing a query, when in fact some terms are of little use in determining relevance.
  - For example, in a collection of thesis dissertations, the term "dissertation" is less likely to be of value since this term probably exists in every document.
- An important measure to incorporate the <u>discriminative power of a term</u> in a collection is the document frequency of a term (df), i.e. the number of documents that contain a term.

Inverse Document Frequency (idf)

- The document frequency of a term is incorporated in the weight of a term by using the concept of <u>inverse document frequency</u> (idf).

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

- Where N is the total number of documents in the collection.

- The rarer the term in a collection, the higher its idf.

tf-idf

- Combining term frequency (tf) with inverse document frequency (idf) results in a classic measure in Information Retrieval, the tf-idf weighting scheme.

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

- tf-idf $_{t,d}$ assigns a term t a weight in a document d that is:
  - highest when t occurs many times within a small number of documents;
  - lower when the term occurs fewer times in a document, or occurs in many documents;
  - lowest when the term occurs in virtually all documents.

Information Retrieval Models

Modeling in IR

- Modeling in IR aims at **producing a raking function**,
  - i.e. assign scores to documents with regard to a given query.
- The process consists of **two main tasks**:
  - Conception of a logical framework for representing documents and queries.
  - Definition of a ranking function that quantifies the similarities between documents and queries.

Information Retrieval Process

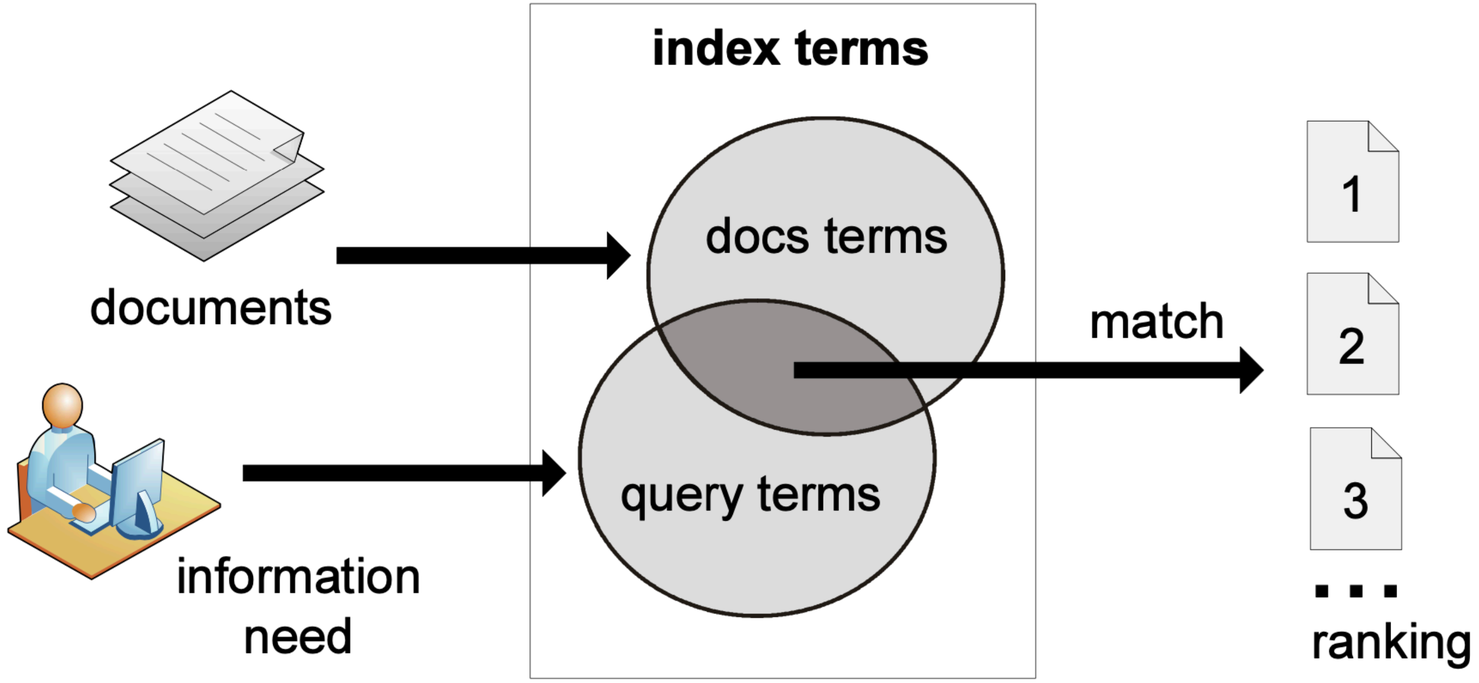


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

IR Models

- An IR model defines…
  - A representation (d_i) of the documents (D) in the collection;
  - A representation (q_j) of the queries (Q) submitted by users:
  - A ranking function $R(q_j, d_i)$.

Taxonomy of IR Models

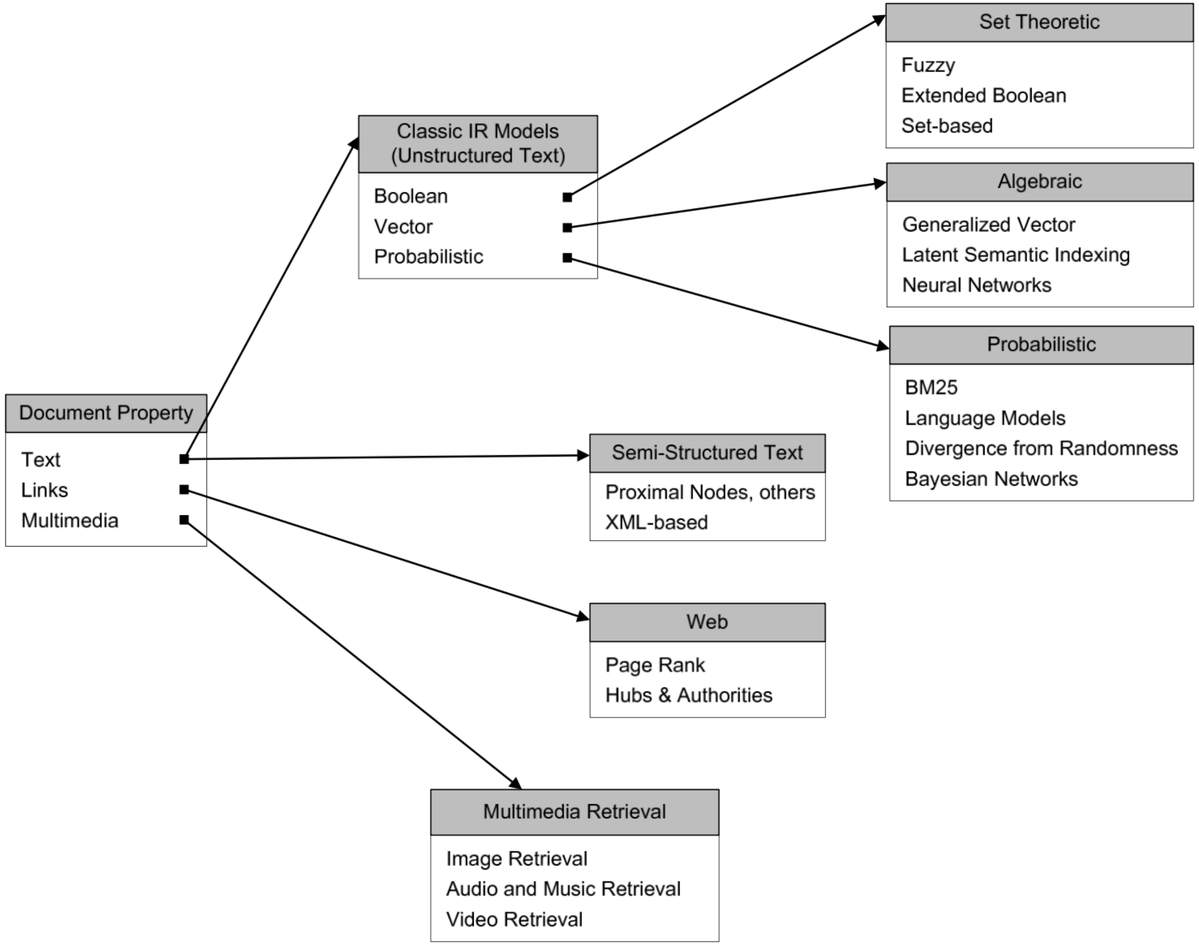


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

Boolean Model

➔ Documents are represented in a term-document matrix.

➔ Queries are specified as boolean expressions.

➔ The Boolean model predicts if documents are either relevant or non-relevant.

➔ No ranking is provided.

Vector Space Model

Vector Space Model

- ➔ The representation of a set of documents as vectors in a common vector space is known as the vector space model and is fundamental to number Information Retrieval operations.

- ➔ In a nutshell, each document is represented as a vector, with a component vector for each dictionary term. tf-idf weights are used as component weights.

- ➔ Thus, the set of documents in a collection may be viewed as a set of vectors in a vector space, in which there is one axis for each term.

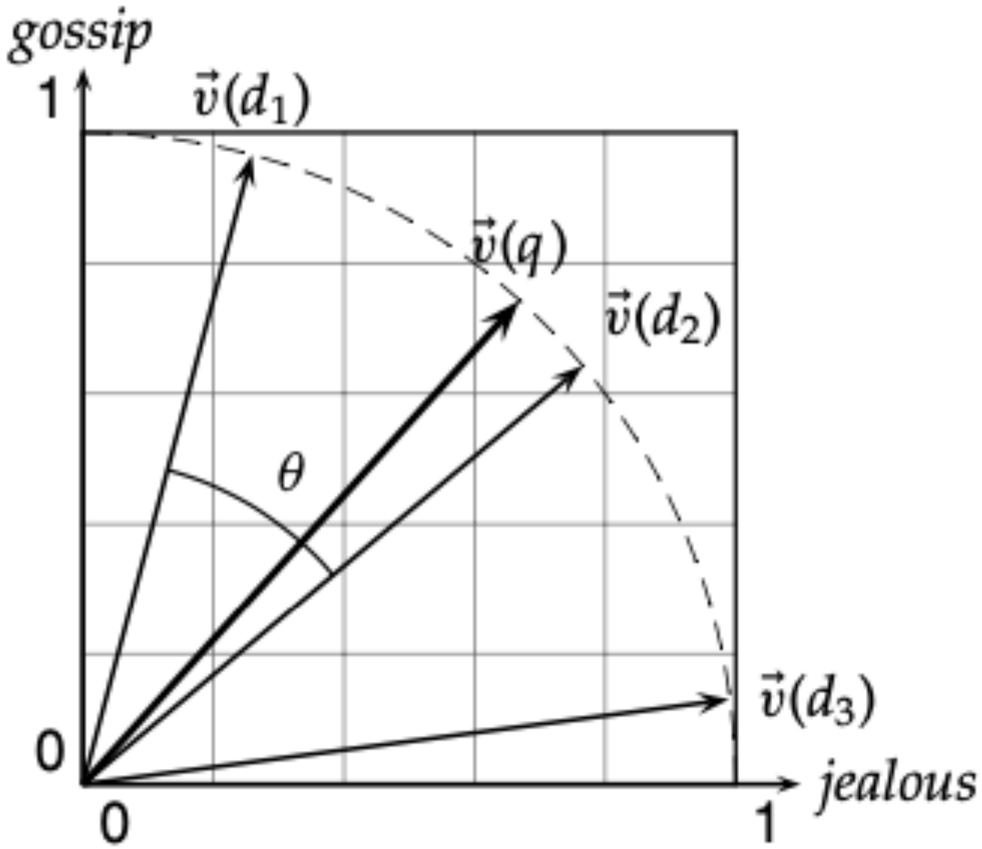


Figure 6.11 Cosine similarity illustrated: $\text{sim}(d_1, d_2) = \cos\theta$.

Image from Introduction to Information Retrieval, Manning et al. (2008)

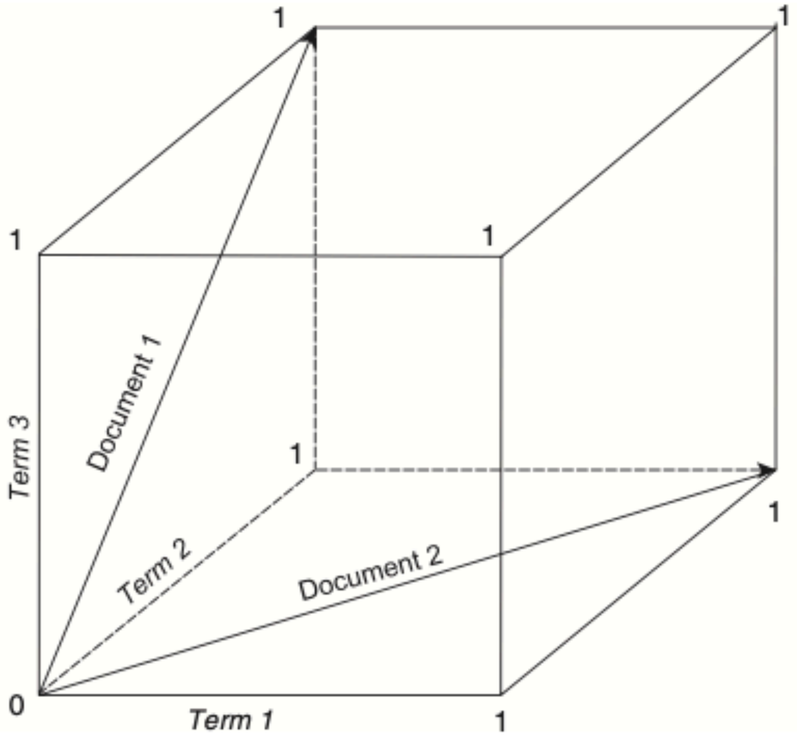


Figure 3.1
Document vectors on the vertices of *the unit hypercube*: each text is represented by a vector whose component, along each term axis, is either 0 or 1. Thus, all vectors must terminate only at one of the corners of the cube. In real life the number of dimensions is far greater than can be illustrated on a two-dimensional page.

Image from Text Information Retrieval Systems, Meadow et al. (2007)

Cosine Similarity

- ➔ To quantify the similarity between two documents in this vector space, the cosine similarity of the vector representations of the two documents.
- ➔ In other words, the similarity between two documents is given by the cosine of the angle between the two vector representations of the documents.
- ➔ This approach compensates the effect of document length.

$$\mathrm{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|},$$

Queries as Vectors

- Queries can also be represented as vectors in a n-dimensional space, being n the number of terms in the query. Basically, queries are viewed as very short documents.

- The top ranked results for a given query are thus the documents whose vectors have the highest cosine similarity in comparison with the query vector.

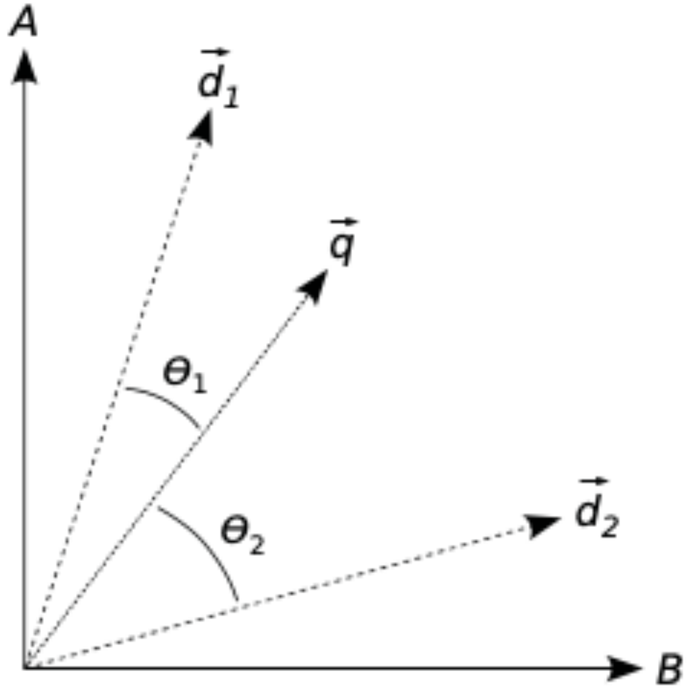


Figure 2.8 Document similarity under the vector space model. Angles are computed between a query vector $\vec{q}$ and two document vectors $\vec{d_1}$ and $\vec{d_2}$. Because $\theta_1 < \theta_2$, d_1 should be ranked higher than d_2.

Image from Information Retrieval: Implementing and Evaluating Search Engines, Buttcher et al. (2016)

Inverse Document Frequency

- ➔ Use Google to estimate the inverse document frequency (idf) value of the following keywords:
- ➔ portugal, technology, health.
- ➔ Again using Google, and the same set of keywords, consider the subset of documents in the .pt domain [site:pt].

$$idf_k = \log \frac{N}{n_k}$$

- ➔ Reflect on the differences.

Inverse Document Frequency Solutions

- **What is Google's index size?**
  - max([the], [and], [is]) -> ~26B
- **World Wide Web**
  - portugal – log (26,000,000,000 / 1,160,000,000) = log (22.41) ≈ 1.35
  - technology – log (26,000,000,000 / 2,640,000,000) ≈ 0.99
  - health – log (26,000,000,000 / 3,140,000,000) ≈ 0.92
- **.pt domain**
  - Update index size [pt site:pt] -> ~278M
  - portugal – log (278,000,000 / 99,600,000) = log (2.79) ≈ 0.44
  - technology – log (278,000,000 / 1,940,000) ≈ 2.16
  - health – log (278,000,000 / 8,260,000) ≈ 1.53

Document Representation

- Consider the following headlines
  - H1: Portugal eyes political balance in presidential election
  - H2: Campaigning begins for Portuguese presidential election
  - H3: After Portuguese elections, Spain braces for elections
- Obtain
  - An incidence matrix considering the following terms: portugal, election, presidential.
  - A matrix with term frequency weights (raw and log normalized).
  - Inverse document frequency values for each of the three terms.
  - A matrix with tf-idf weights.
- For each headline, obtain a unitary vector representation using raw tf in 1 dimension (portugal), 2 dimensions (portugal, election) and in 3 dimensions (portugal, election, presidential).
- Calculate the cosine similarity between each document and the following query: [portugal election].

Incidence Matrix

	H1	H2	H3
portugal	1	0	0
election	1	1	1
presidential	1	1	0

- H1: Portugal eyes political balance in presidential election
- H2: Campaigning begins for Portuguese presidential election
- H3: After Portuguese elections, Spain braces for elections

Term Frequency (raw and log normalized)

	H1	H2	H3
portugal	1 (1)	0 (0)	0 (0)
election	1 (1)	1 (1)	2 (1.3)
presidential	1 (1)	1 (1)	0 (0)

- H1: Portugal eyes political balance in presidential election
- H2: Campaigning begins for Portuguese presidential election
- H3: After Portuguese elections, Spain braces for elections

Inverse Document Frequency

	idf
portugal	$\log(3 / 1) \approx 0.48$
election	$\log(3 / 3) = 0$
presidential	$\log(3 / 2) \approx 0.18$

- H1: Portugal eyes political balance in presidential election
- H2: Campaigning begins for Portuguese presidential election
- H3: After Portuguese elections, Spain braces for elections

TF-IDF (log normalized tf)

	H1	H2	H3
portugal	1 x 0.48 = 0.48	0 x 0.48 = 0	0
election	0	0	0
presidential	0.18	0.18	0

- H1: Portugal eyes political balance in presidential election
- H2: Campaigning begins for Portuguese presidential election
- H3: After Portuguese elections, Spain braces for elections

Unitary Vector Documents

- 1-dimension: (portugal)
  - H1: (1)
  - H2: (0)
  - H3: (0)
- 2-dimension (portugal, election)
  - H1: (1,1) > (0.707, 0.707)
  - H2: (0,1) > (0, 1)
  - H3: (0,2) > (0, 1)

Document Vectors

- 3-dimension: (portugal, election, presidential)
  - H1: (1,1,1) > (0.577, 0.577, 0.577)
    - $||H1|| = \sqrt{(1^2+1^2+1^2)} = 1.732$
  - H2: (0,1,1) > (0, 0.707, 0.707)
  - H3: (0,2,0) > (0, 1, 0)

TF-IDF (log normalized tf)

- ➔ Q: (portugal, election, presidential) = (1,1,0)
  - ➔ Unitary vector: (1 / √(1^2 + 1^2)) = 0.707
  - ➔ = (0.707, 0.707, 0)
- ➔ sim(H1,Q) = (0.577 x 0.707) + (0.577 x 0.707) + + (0 x 0.707) = 0,815
- ➔ sim(H2,Q) = (0.707 x 0) + (0.707 * 0.707) + + (0 x 0.707) = 0.5 (60 degrees)
- ➔ sim(H3,Q) = 0.707
- ➔ Ranking for Q: H1, H3, H2

$$Cosine(D_i, Q) = \frac{\sum_{j=1}^{t} d_{ij} \cdot q_j}{\sqrt{\sum_{j=1}^{t} {d_{ij}}^2 \cdot \sum_{j=1}^{t} {q_j}^2}}$$

Extra: similarity between documents

- ➔ H1: (0.577, 0.577, 0.577)
- ➔ H2: (0, 0.707, 0.707)
- ➔ H3: (0, 1, 0)

	H1	H2	H3
H1	1	(0.577 x 0.707) x 2 = 0.815	0,577
H2	0,815	1	0,707
H3	0,577	0,707	0 x 0 + 1 x 1 + 0 x 0 = 1

Language Models for Information Retrieval

Language Models for Information Retrieval

- ➔ Central idea: a document is a good match for a query if the document's model is likely to generate the query.
- ➔ In the basic language model approach, a probabilistic language model is built for each document in the collection (M_d).
- ➔ For a given query, documents are ranked based on the probability of the model generating the query: $P(q|M_d)$.

Language Models

- A language model is a function that puts a probability measure over strings drawn from some vocabulary.
- The sum of all probabilities over a vocabulary for a language model is 1.
- The simplest language model, discards all context information (i.e., nearby words), and estimated the probability of each term independently — this is called an <u>unigram model</u>.
  - In this case, the probability of a sequence of terms (e.g., a query) is simply the product of independent term probabilities:
  - $P_{unigram}(t_1 t_2 t_3 t_4) = P(t_1) \times P(t_2) \times P(t_3) \times P(t_4)$

Language Models

- Bigram language models condition the probability of each term on the previous item:
  - $P_{bigram}(t_1 t_2 t_3 t_4) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_2) \times P(t_4|t_3)$
- More complex language models are important in tasks such as speech recognition, spelling correction, or machine translation.
- In Information Retrieval most language-modeling work uses unigram language models. In IR, language models are often estimated from a single model so there is little information to do more.

Example of Language Models

- D1: Portugal eyes political balance in presidential election
- D2: After Portuguese elections, Spain braces for elections

- Unigram Language Models:
  - M_{d1}: portugal: 0.143 (1/7); eye: 0.143; ...
  - M_{d2}: after: 0.143; portugal: 0.143; election: 0.286 (2/7); ...

Retrieval based on Language Models

- Approach for retrieving documents under a language model (LM):
  - 1. Infer a LM for each document.
  - 2. Estimate $P(q|M_{di})$, the probability of generating the query according to each of these document models.
  - 3. Rank the documents according to these probabilities.

Example of Retrieval with Language Models

- ➔ D1: Portugal eyes political balance in presidential election
- ➔ D2: After Portuguese elections, Spain braces for elections

- ➔ Q: [portugal election]
- ➔ $P(q|d_1) = P(portugal|M_{d1}) \times P(election|M_{d1}) = 1/7 \times 1/7 = 0.0204$
- ➔ $P(q|d_2) = P(portugal|M_{d2}) \times P(election|M_{d2}) = 1/7 \times 2/7 = 0.0408$

Taxonomy of IR Models

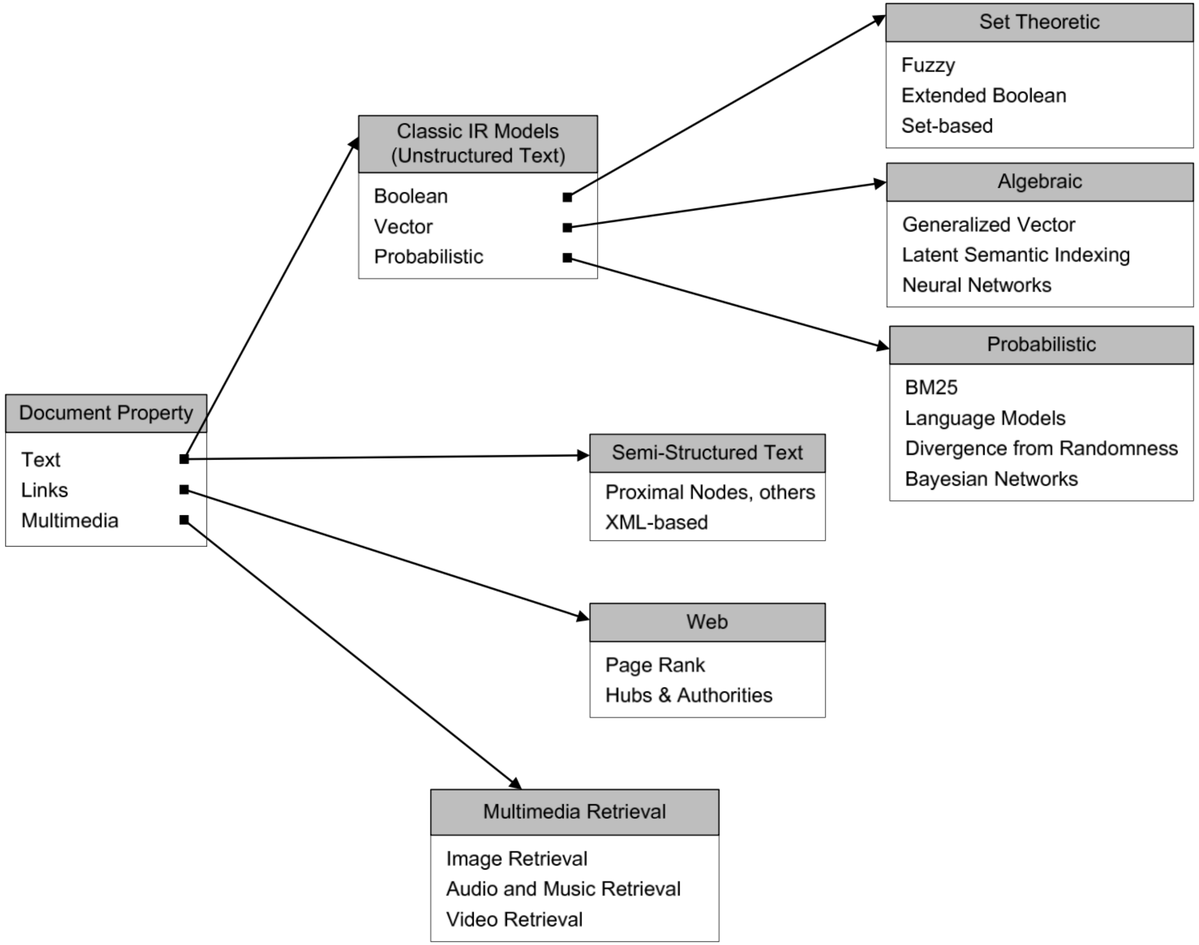


Image from Modern Information Retrieval, Baeza-Yates et al. (2011)

Information Retrieval on the Web

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Chapters 19, 20 and 21 from Introduction to Information Retrieval, Manning, C. et al. (2008)
Based on Chapters 11 and 12 from Modern Information Retrieval (2nd Edition), Baeza-Yates, R. et al. (2010)

Outline

- Overview of Information Retrieval on the Web
- Web Crawling
- Web Ranking
- Link Analysis
  - PageRank
  - HITS

Information Retrieval on the Web

The World Wide Web

- The web is unprecedented in many ways:
  - Unprecedented in scale (size and change);
  - Unprecedented in lack of central coordination;
  - Unprecedented in the diversity of users' backgrounds and needs.
- Two types of challenges:
  - Data: distribution, size, volatility, quality, unstructured, duplicates.
  - Interaction: user needs; relevance; diversity of users.

Information Access on the Web

- Early projects of making information discoverable on the web fell into two categories.
- **Full-text index search engines**, e.g. Altavista, Excite, Infoseek
  - Use keyword-based search supported by inverted indexes and ranking mechanisms;
- **Directories / Taxonomies**, e.g. Yahoo!, Open Directory Project
  - Navigate through a hierarchical tree of category labels.
  - Problems: mostly manual categorization (high cost and not scalable); mismatch between editors' and users' idea of how to organize a given node.

Web Search Challenges

- Decentralization of content publication (the web biggest innovation) means that there is no central point that keeps track of what exists, what was changed, or deleted.
- What is the size of the web?
- Content is created massively and in diverse forms
  - Diversity of languages and dialects;
  - Diversity of formats, i.e. structure, color, size, …;
  - Quality, i.e. truths, lies, contradictions, …;
  - Intent, i.e. legitimate, spam, search engine manipulation;

Web Characteristics

- ➔ The Web can be modeled as a graph.
- ➔ Web pages point to, and are pointed by, other pages.
- ➔ Links to other pages (out-links) usually include an "anchor text".
- ➔ The number of in-links to a page is called the in-degree.
- ➔ Studies of web characteristics and dynamics is an area of research.
- ➔ Early research suggests that the directed graph connecting web pages has a bowtie shape.

Web Graph

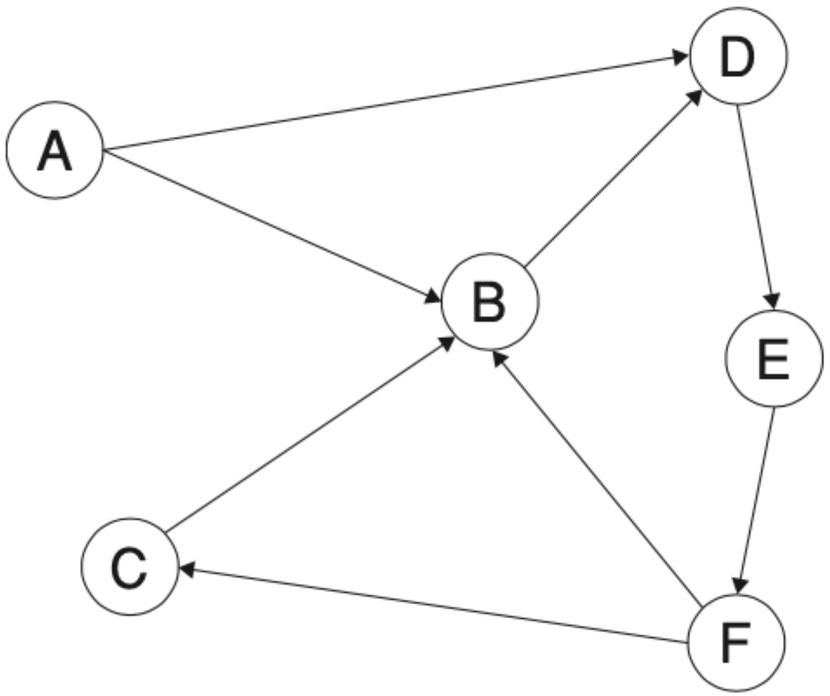
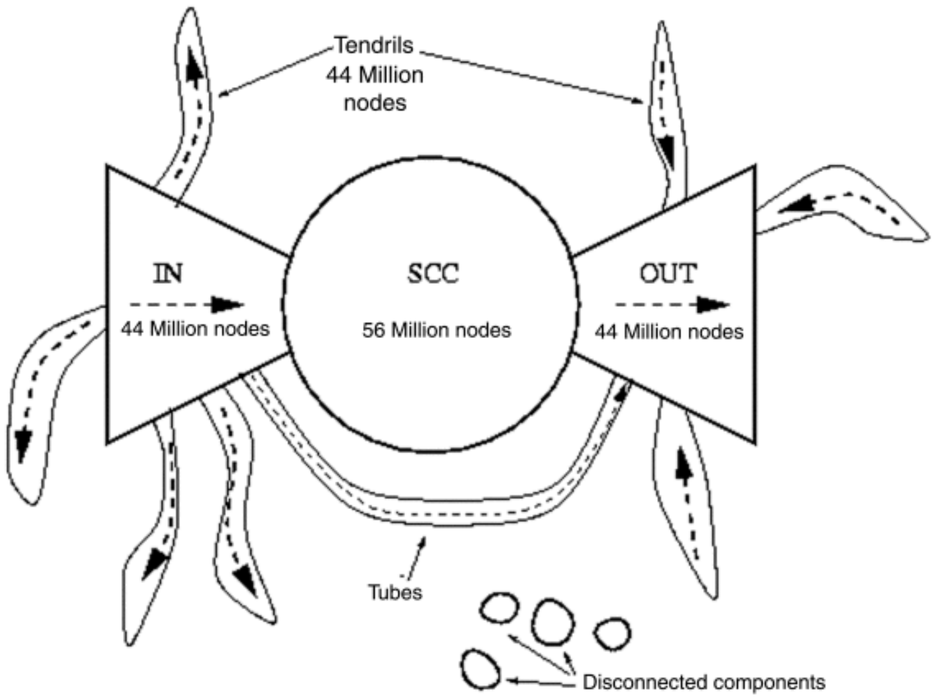


Figure 19.3 A sample small web graph. In this example we have six pages labeled A through F. Page B has in-degree 3 and out-degree 1. This example graph is not strongly connected: There is no path from any of pages B through F to page A.

Image from Introduction to Information Retrieval, Manning, C. et al. (2008)

Bowtie Shape of the Web



Download : Download high-res image (243KB) Download : Download full-size image

Fig. 9. Connectivity of the Web: one can pass from any node of IN through SCC to any node of OUT. Hanging off IN and OUT are TENDRILS containing nodes that are reachable from portions of IN, or that can reach portions of OUT, without passage through SCC. It is possible for a TENDRIL hanging off from IN to be hooked into a TENDRIL leading into OUT, forming a TUBE: i.e., a passage from a portion of IN to a portion of OUT without touching SCC.

Image from Graph Structure in the Web, Broder et al. Computer networks (2000)

Web Spam

- ➔ There is a (high) commercial value associated with appearing on the top ranked results for a given search.
- ➔ Search engines must be resistant to manipulation attempts (high frequency!).
- ➔ For instance, a search engine whose scoring depends on the frequency of keywords, would be easy to manipulate by including numerous repetitions of selected keywords.
- ➔ This is called web spam, i.e. the manipulation of content on the web with the purpose of manipulating search engine rankings. Examples include: cloaking, link farms, link spam, click spam, etc.
- ➔ Topics in the sub-area of adversarial information retrieval.

Cloaking

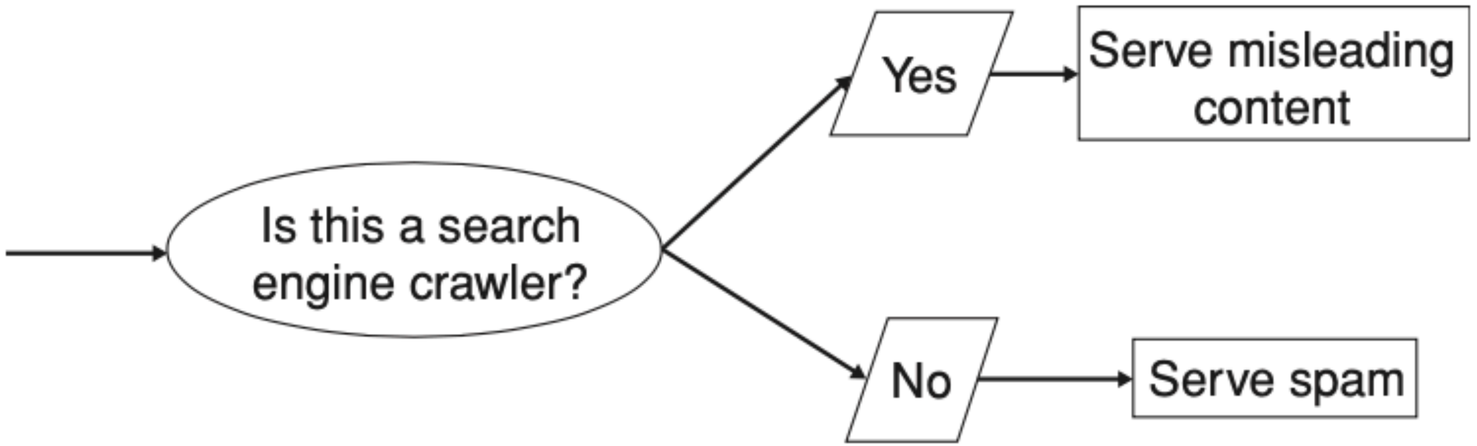


Figure 19.5 Cloaking as used by spammers.

Image from Introduction to Information Retrieval, Manning, C. et al. (2008)

"Mixed Motives in Search"

➔ Reflexion from Brin and Page in 1998.

8 Appendix A: Advertising and Mixed Motives

Currently, the predominant business model for commercial search engines is advertising. The goals of the advertising business model do not always correspond to providing quality search to users. For example, in our prototype search engine one of the top results for cellular phone is "The Effect of Cellular Phone Use Upon Driver Attention", a study which explains in great detail the distractions and risk associated with conversing on a cell phone while driving. This search result came up first because of its high importance as judged by the PageRank algorithm, an approximation of citation importance on the web [Page, 98]. It is clear that a search engine which was taking money for showing cellular phone ads would have difficulty justifying the page that our system returned to its paying advertisers. For this type of reason and historical experience with other media [Bagdikian 83], we expect that advertising funded search engines will be inherently biased towards the advertisers and away from the needs of the consumers.

Since it is very difficult even for experts to evaluate search engines, search engine bias is particularly insidious. A good example was OpenText, which was reported to be selling companies the right to be listed at the top of the search results for particular queries [Marchiori 97]. This type of bias is much more insidious than advertising, because it is not clear who "deserves" to be there, and who is willing to pay money to be listed. This business model resulted in an uproar, and OpenText has ceased to be a viable search engine. But less blatant bias are likely to be tolerated by the market. For example, a search engine could add a small factor to search results from "friendly" companies, and subtract a factor from results from competitors. This type of bias is very difficult to detect but could still have a significant effect on the market. Furthermore, advertising income often provides an incentive to provide poor quality search results. For example, we noticed a major search engine would not return a large airline's homepage when the airline's name was given as a query. It so happened that the airline had placed an expensive ad, linked to the query that was its name. A better search engine would not have required this ad, and possibly resulted in the loss of the revenue from the airline to the search engine. In general, it could be argued from the consumer point of view that the better the search engine is, the fewer advertisements will be needed for the consumer to find what they want. This of course erodes the advertising supported business model of the existing search engines. However, there will always be money from advertisers who want a customer to switch products, or have something that is genuinely new. But we believe the issue of advertising causes enough mixed incentives that it is crucial to have a competitive search engine that is transparent and in the academic realm.

http://infolab.stanford.edu/~backrub/google.html

Search Engines Economic Model

- ➔ In the early years, banner advertisements were used to support search engines operation, i.e. static banners. These banners are typically priced on a cost per mille (CPM) basis, i.e. cost per 1,000 impressions.
- ➔ Alternatively, commercial contracts can be set on a number of clicks basis — cost per click (CPC). The advertiser only pays when the searcher clicks on a search result.
- ➔ Current economic models are based on a (realtime) bidding paradigm, where advertisers bid on specific keywords and search engines show the ads based on the value of the bid.
- ➔ These results are known as sponsored search results, while the search engine's "pure" rankings are known as algorithmic or organic results.
- ➔ Targeting ads to keywords is a basic targeting strategy. More sophisticated (and problematic) approaches can explore contextual or behavioral user data, e.g. location, navigation patterns, previous searches, etc.

www.google.com/search?q=lego&biw=1111&bih=850

Google

lego

Entrar

Tudo | Imagens | Compras | Vídeos | Notícias | Mais | Ferramentas

Cerca de 1 090 000 000 resultados (0,77 segundos)

Anúncio · https://www.lego.com/

Conjuntos e exclusivos LEGO® - Compra conjuntos LEGO

Compra conjuntos e brinquedos **LEGO** e descobre o presente perfeito para as crianças. A maior seleção **LEGO** online. Explora agora em **LEGO**.com! Ganhe recompensas VIP. Qualidade garantida. Retorno sem complicações. Brands: Creator, Star Wars, Ideas.

Fender® Stratocaster™

Vive o sonho de estrela de rock com a guitarra Fender Strat! Compra já

Compra este Super Mario™

Revive memórias enquanto constrói o Bloco de interrogação Mario 64™

Compra o LEGO® Queer Eye

Constrói o Loft dos Fab Five da série galardoada Queer Eye

Anúncio · https://www.worten.pt/

Worten - Lego Construções - Aproveita Grandes Descontos

Entregas em Todo o País! Aproveita Já as Promoções em **Lego** Construções em Worten.pt.
-50% Jogos e Brinquedos · Lego Friends · Lego Disney · Lego Minecraft · Lego Star Wars

Anúncio · https://www.toysrus.pt/

Jogos d Sets. Ampla coleção da LEGO | ToysRUs - ToysRUs.pt

Envio grátis sem compra mínima. Em todo o site | Toys"R"Us. Por tempo limitado. Se...

https://www.lego.com › pt-pt

Home | Loja LEGO® Oficial PT

16/12/2020 — Explore the world of **LEGO**® through games, videos, products and more! Shop awesome **LEGO**® building toys and brick sets and find the perfect ...

Anúncios · Comprar

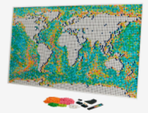
LEGO Still Life with…
24,99 €
LEGO.com
★★★★★ (3)
De Google

LEGO Hogwarts…
429,99 €
LEGO.com
★★★★★ (1k+)
De Google

LEGO Camp Nou – FC…
329,99 €
LEGO.com
De Google

LEGO Infinity…
74,99 €
LEGO.com
★★★★★ (337)
De Google

PROMOÇÃO



User Characteristics

- Web search users are not trained on how to write queries using the search operators offered by search engines.
- Studies in search user behavior show that users use between two or three keywords, search operators are rarely used, precision in the top results is highly valued, and lightweight result pages are preferred.
- Early research has shown that users' information needs can be grouped in three types:
  - Informational queries, i.e. seek general information about a topic. There is typically not a single source of relevant information. Users typically gather information from multiple web pages.
  - Navigational queries, i.e. seek the website or home page of a single entity. Users' expectations is to find a specific resource.
  - Transactional queries, i.e. preludes the user performing a transaction on the web, such as purchasing a product, downloading a file, or doing a reservation.

Signals for Web Ranking

- Hundreds of signals are used by search engines to estimate quality.
- Signals can be grouped by different dimensions:
  - Query-independent signals (static).
  - Query-dependent signals (dynamic).

  - Document-based signals (content or structural), e.g. HTML.
  - Collection-based signals, e.g. Links.
  - User-based signals, e.g. Clicks.

Web Search Engine Architecture

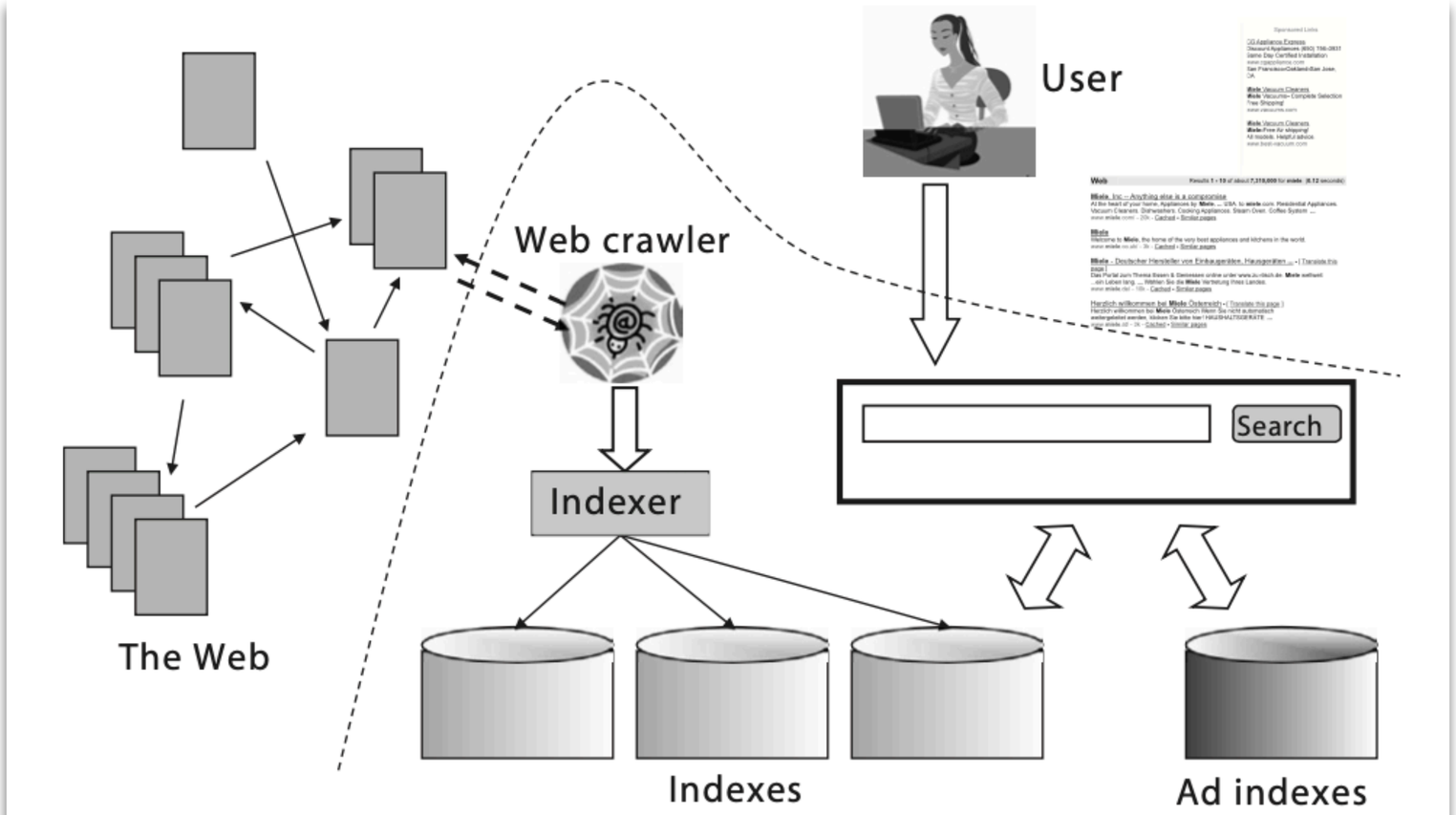
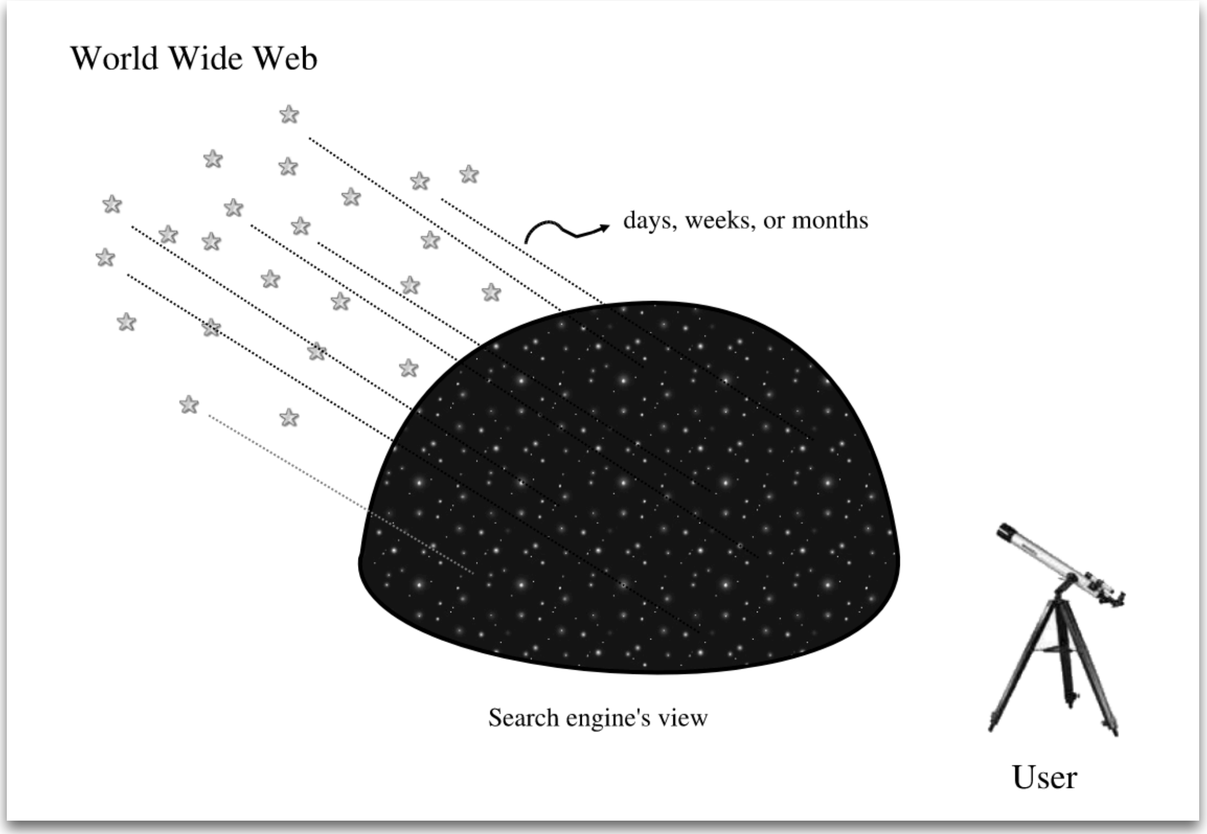


Figure 19.7 The various components of a web search engine.

Image from Introduction to Information Retrieval, Manning, C. et al. (2008)

Web Crawling



Crawling the Web, in a certain way, resembles watching the sky in a clear night: the star positions that we see reflects the state of the stars at different times

From Modern Information Retrieval, Baeza-Yates, R. et al. (2010)

Web Crawling

- Web crawling is the process by which pages are gathered from the web.
- The goal is to discover as quickly and as efficiently as possible quality web pages.
- Features a crawler **must** provide:
  - robustness in face of problems, unexpected content, or traps,
  - politeness to web hosts.
- Features a crawler **should** provide: execute in a distributed fashion, scalable, efficient use of resources, bias towards good quality pages, freshness depending on page change rate, and extensible to cope with innovations on the web.
- Industry standards: robots.txt (robotstxt.org); sitemap protocol (sitemaps.org).

Politeness

- A crawler would ideally use all the available resources to obtain the collection
- However, crawlers should fulfill **politeness**
  - That is, a crawler cannot overload a web site with HTTP requests;
  - That implies that a crawler should wait a small delay between two requests to the same web site.
- A crawler that is impolite may be banned by the hosting provider.
- Three basic rules for web crawler operation:
  - A crawler must identify itself as such, and must not pretend to be a regular user.
  - A crawler must obey the robots exclusion protocol (robots.txt).
  - A crawler must keep a low bandwidth usage in a given web site.
- Example crawler behavior:
  - Google Crawler https://developers.google.com/search/docs/crawling-indexing/googlebot

Robot Exclusion Protocol (1)

- The robot exclusion protocol involves three types of exclusion:
  - server-wide, page-wise exclusions, and cache exclusions.
- **Server-wide exclusion** instructs about directories that should not be crawled.
  - These exclusions are defined via the robots.txt file located at the root.
  - All crawlers should not download the directory /data/private.
    - `User-agent: *`
    - `Disallow: /data/private`
- Examples:
  - https://sigarra.up.pt/robots.txt
  - https://facebook.com/robots.txt

Robot Exclusion Protocol (2)

- **Page-wise exclusion** is done by the inclusion of meta tags directly on in the pages.
- Meta tags are part of the HTML standard.
- This example indicates that crawlers should neither index this page, not follow links contained on it.
  - `<meta name="robots" content="noindex,nofollow" />`
- **Cache exclusion** is used to instruct search engines not to show the user a local cached copy of the page.
  - `<meta name="robots" content="nocache" />`

Applications of Web Crawling

- A web crawler can be used to
  - create an index covering broad topics (general web search);
  - create an index covering specific topics (vertical web search);
  - archive content (archiving, backup);
  - analyze web sites for extracting aggregate statistics (characterization, analytics);
  - keep copies or replicate web content (mirroring);
  - web site analysis.

Taxonomy of Crawling

- ➔ Crawlers assign different importance to issues such as freshness, quality, and volume.
- ➔ Crawlers can be classified according to these three axes.

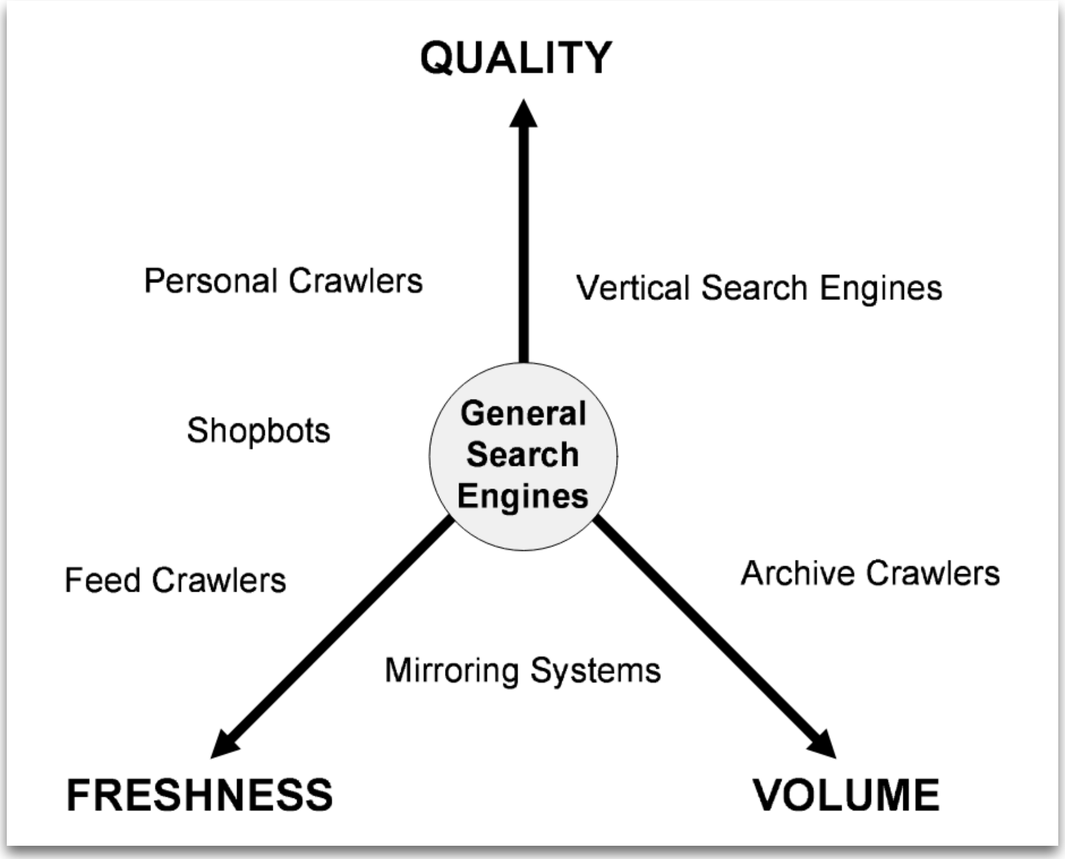
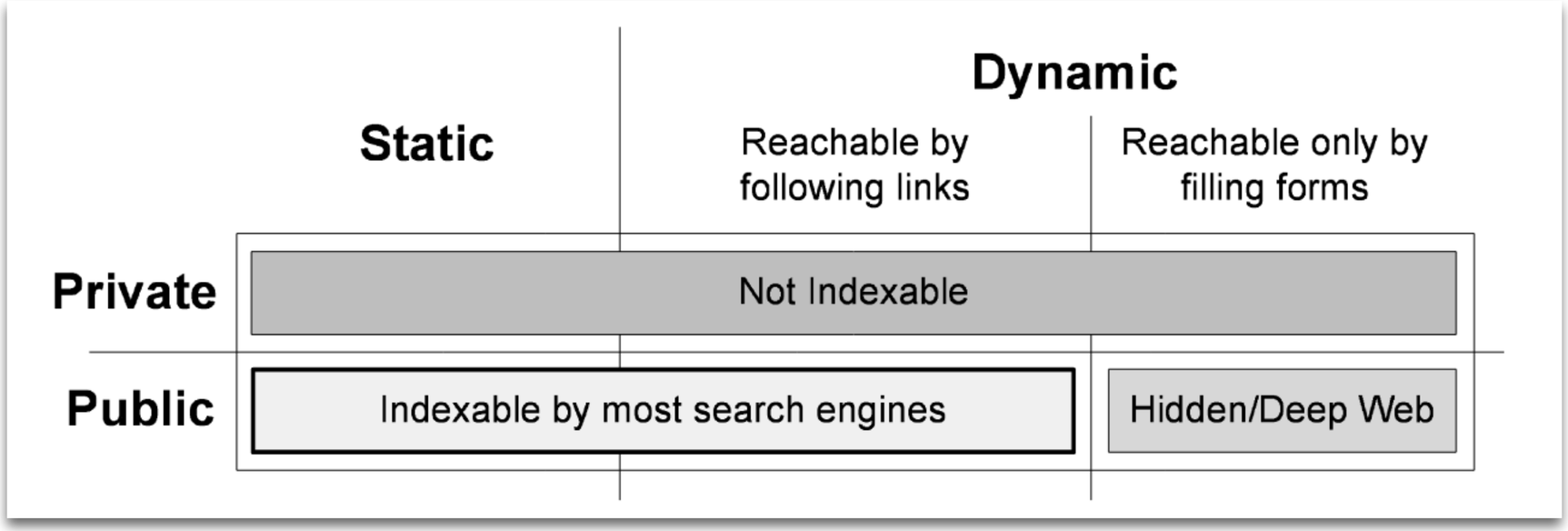


Image from Modern Information Retrieval, Baeza-Yates, R. et al. (2010)

Taxonomy of Web Pages



Web Crawling Process

- ➔ The crawler starts with a set of **seed pages**,
  - ➔ these are downloaded, parsed, and scanned for new links.
- ➔ The links to pages that have not yet been downloaded are added to a central queue for download later (URL frontier).
- ➔ The crawler selects a new page for download and the processes is repeated:
  - ➔ until a stop criterion is met.
  - ➔ considering an established update / refresh policy.

Web Crawler Architecture

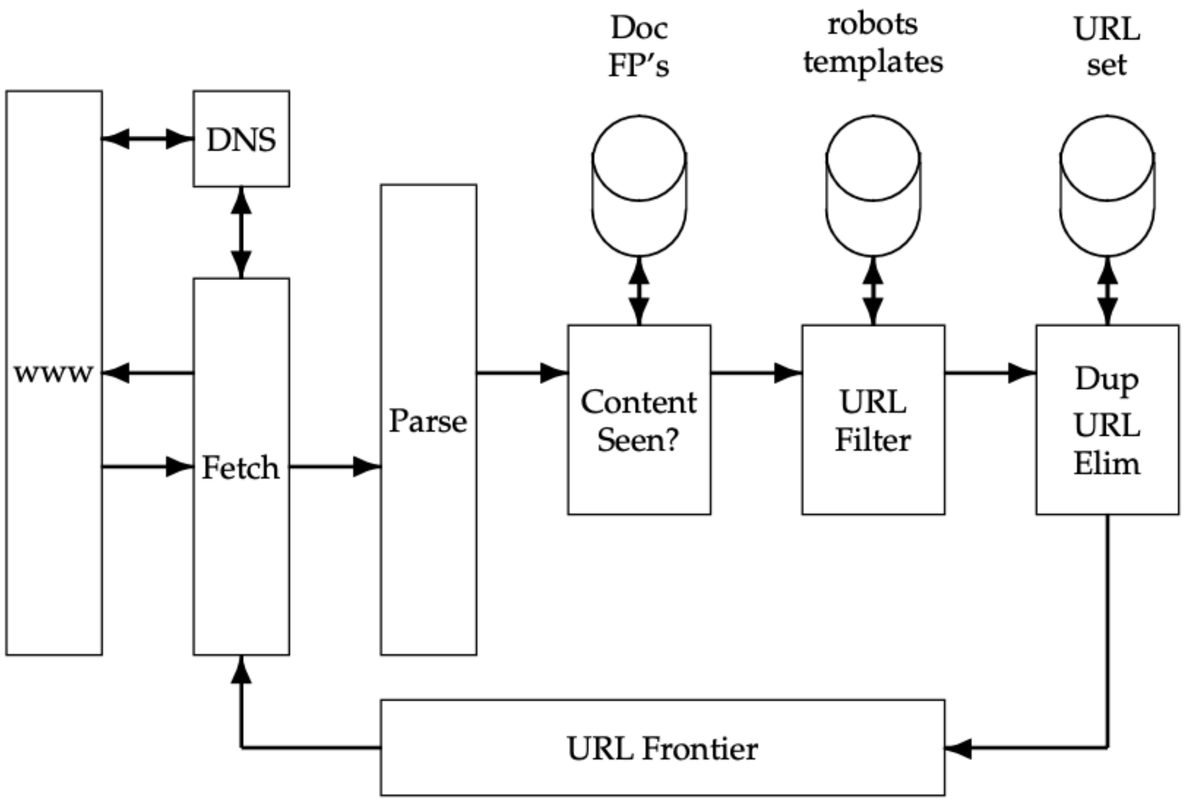


Figure 20.1 Basic crawler architecture.

Image from Introduction to Information Retrieval, Manning, C. et al. (2008)

Near-Duplicate Detection

- A large percentage of content on the web are near-duplicates, i.e. differing only in small parts. Standard duplicate detection methods do not work, e.g. fingerprinting.
- Crawlers need to decide if new pages are duplicates of existing content and if pages being revisited have changed since last visit, e.g. estimate change rate.
- Common solution: obtain shingles of k size from web pages; compare shingles from two pages to determine if they are near-duplicates. The more shingles in common, the more similar are the pages.
- Example of k=4 shingles for the text: [these are red and blue roses]
  - [these are red and], [are red and blue], [red and blue roses]

Open Source Crawlers

- ➔ Heritrix, Internet Archive (Java), https://github.com/internetarchive/heritrix3
- ➔ Nutch, Apache (Java), https://nutch.apache.org
- ➔ Scrapy, (Python), https://github.com/scrapy/scrapy

Web Ranking

Web Ranking

- ➔ Ranking is the most complex and most important function of a search engine.
- ➔ First challenge, implement an evaluation process:
  - ➔ Devise an adequate process of evaluating the ranking in terms of relevance of results.
  - ➔ Such evaluation is necessary to fine tune the ranking function.
- ➔ Second challenge, identify quality content:
  - ➔ Collect and combine evidence of quality from different signals.
- ➔ Third challenge, avoid, prevent and manage web spam.
- ➔ Four challenge, define a ranking function and compute it.

Ranking Signals

- Signals can be organized in different types.
- Content signals
  - related to the text itself.
  - consider HTML semantics (headings, sections, links).
- Structural signals
  - related to the link structure of the web.
  - can be textual (e.g. anchor text), or related to the links (e.g. number).
- Usage signals
  - related to the feedback provided by the users (e.g. clicks).
  - other usage signals include geographical location, technological context, temporal context.

Ranking Signals

- Signals can also be distinguished according to other dimensions.
  - User-dependent, depend on the user's characteristics.
  - Query-dependent, depend on the user's query.
  - Document-dependent, depend on a single document.
  - Collection-dependent, depend on information from the complete collection.

- The characteristics of each signal will impact its computation.

Learning to Rank

- Given dozens or hundreds of signals, how to combine them in a ranking function?
- Traditional approaches rely on manual selection of signals and respective weights, depending on domain knowledge.
- Modern approaches are based on machine learning techniques to
  - learn the ranking of the results.
  - Given a query Q, different training data can be used:
    - pointwise, a set or relevant pages for Q.
    - pairwise, a set of pairs of relevant pages indicating the ranking relation between the two pages.
    - listwise, a set of ordered relevant pages.

Link Analysis

Link-based Signals

- Links are one of the distinctive features of a collection of web documents.
- Base assumption: the number of hyperlinks pointing to a page provides a measure of its popularity and quality.
- Link-based ranking algorithms build on the assumption that an hyperlink from page A to page B represents an endorsement of page B, by the creator of page A.
- Two classic algorithms: PageRank and HITS.
  - PageRank, Larry Page and Sergey Brin (1996)
  - HITS, Jon Kleinberg (1997)

PageRank

- The PageRank of a node is a value between 0 and 1.
- It is a query-independent value computed offline that only depends on the structure of the web graph.
- The algorithms models a random surfer who begins at a web page and, at each step, randomly chooses between the out-links to move to the next page. If the random surfer executes this "forever", he will visit some pages more frequently than others. The PageRank value of a page represents this probability.

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

PageRank Illustration

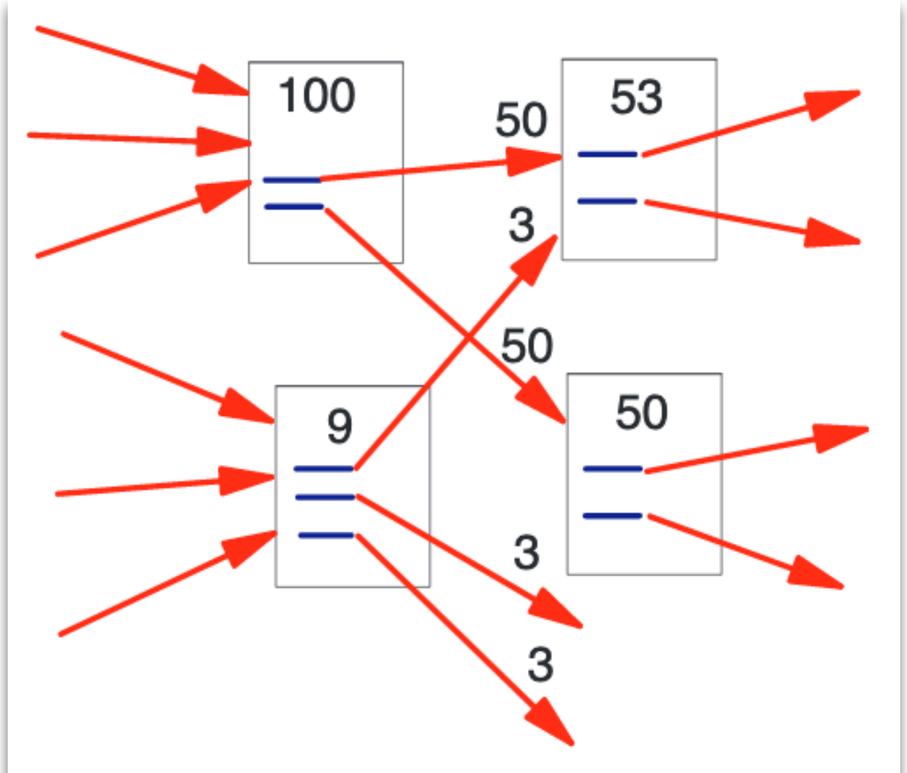


Figure 2: Simplified PageRank Calculation

From: "The PageRank Citation Ranking: Bringing Order to the Web" (1998) http://ilpubs.stanford.edu:8090/422/

Hyperlink Induced Topic Search (HITS)

- Query-dependent algorithm.
- Starts with the answer set (e.g. pages containing the keywords).
- Computes two scores for each page: authority score and hub score.
  - Pages with many links pointing to them are called <u>authorities</u>.
  - Pages with many outgoing links are called <u>hubs</u>.

$$H(p) = \sum_{u \in S \mid p \to u} A(u) \,, \qquad A(p) = \sum_{v \in S \mid v \to p} H(v)$$

HITS Illustration

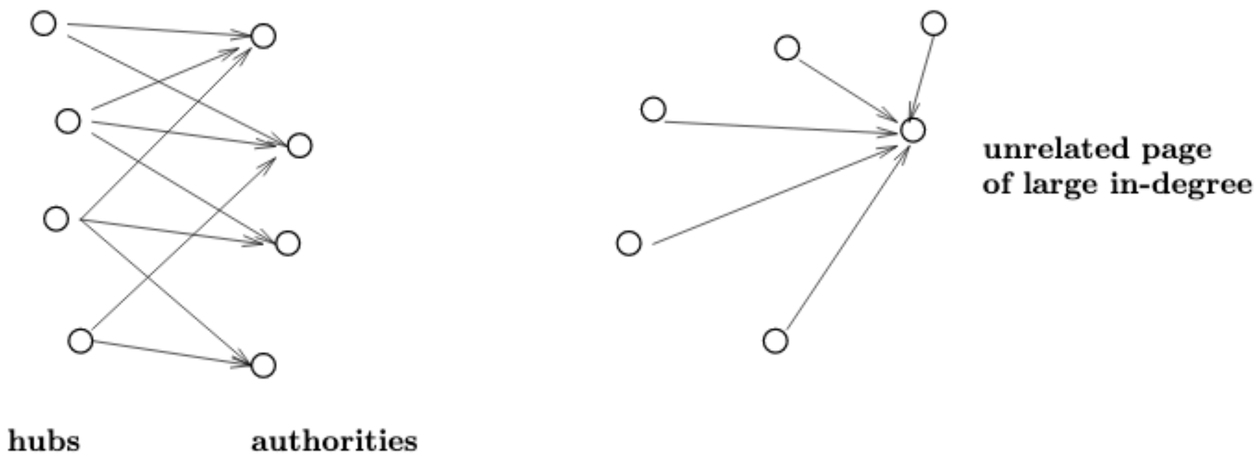


Figure 2: A densely linked set of hubs and authorities.

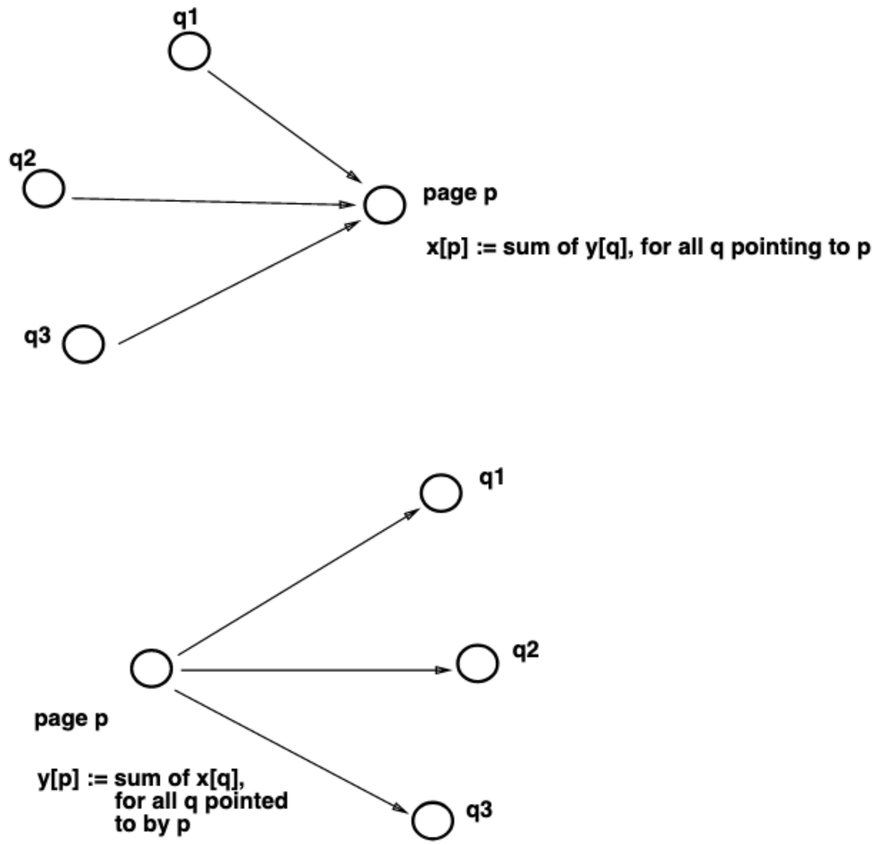


Figure 3: The basic operations.

From: "Authoritative Sources in a Hyperlinked Environment" (1999) https://www.cs.cornell.edu/home/kleinber/auth.pdf

Anchor Text as a Signal

- The text used in HTML anchors, i.e. links, is called anchor text.
- <a href="http://www.fe.up.pt">Faculty of Engineering of the University of Porto</a>
- Represents a description from "others" about a given web page.
- The collection of all anchor texts can be explored with standard IR techniques, and incorporated as an additional features in an inverted index.
- Important feature for image search.

- See "Google bombing", https://en.wikipedia.org/wiki/Google_bombing

References

- Sergey Brin and Lawrence Page, The anatomy of a large-scale hypertextual Web search engine. Comput. Netw. ISDN Syst. 30, 1-7 (April 1998), http://infolab.stanford.edu/~backrub/google.html

- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search
  - Chapter 11: Web Retrieval + Slides for Chapter 11: Web Retrieval
  - Slides for Chapter 12: Web Crawling

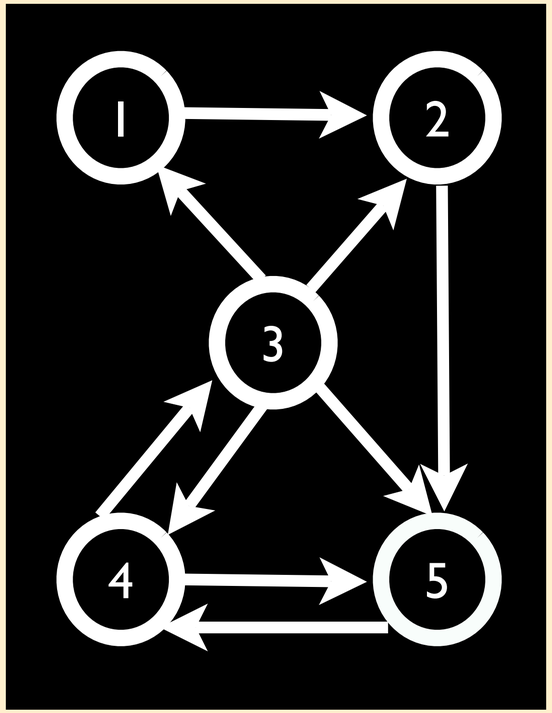
- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval
  - Chapter 19: Web search basics
  - Chapter 20: Web crawling and indexes
  - Chapter 21: Link analysis

Exercises

PageRank and HITS

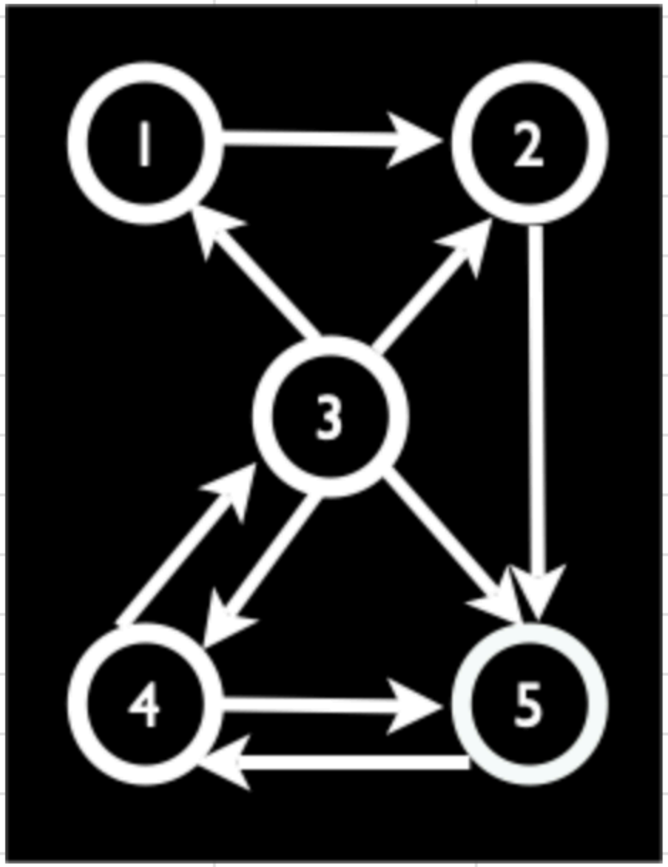
➔ Rank all nodes according to their PageRanks score.

➔ Rank all nodes according to their HITS Authority and Hub scores.



PageRank

- ➔ $PR_1 = PR_3 / 4$
- ➔ $PR_2 = PR_1 / 1 + PR_3 / 4$
- ➔ $PR_3 = PR_4 / 2$
- ➔ ....

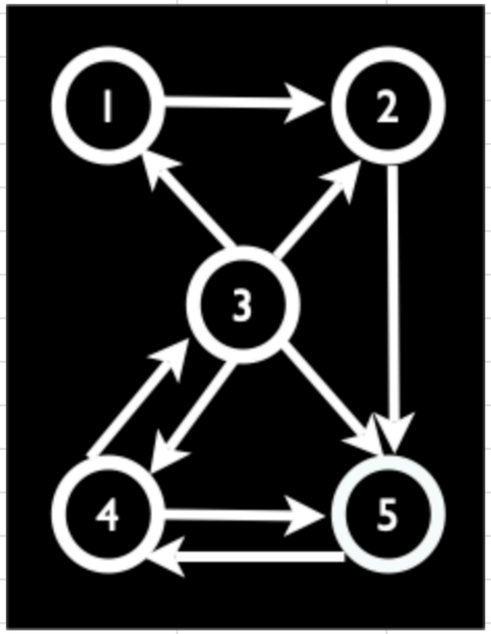


PageRank Score

	Iteration 1	2
P1	0.2	0.050
P2	0.2	0.250
P3	0.2	0.100
P4	0.2	0.250
P5	0.2	0.350
sum		1.000
error		0.500

HITS

- ➔ $Auth_1$ = (Hub_3) / SUM(All Hub Votes)
- ➔ $Auth_2$ = (Hub_1 + Hub_3) / SUM(All Hub Votes)
- ➔ $Auth_3$ = (Hub_4) / SUM(All Hub Votes)
- ➔ ...
- ➔ Hub_1 = ($Auth_2$) / SUM(All Auth Votes)
- ➔ Hub_2 = ($Auth_5$) / SUM(All Auth Votes)
- ➔ Hub_3 = ($Auth_1$ + $Auth_2$ + $Auth_4$ + $Auth_5$) / SUM(All Auth Votes)
- ➔ ....



Authority scores		
	Iteration 1	**2**
P1	1	0.111
P2	1	0.222
P3	1	0.111
P4	1	0.222
P5	1	0.333
sum		9
Hub scores		
	Iteration 1	**2**
P1	1	0.105
P2	1	0.158
P3	1	0.421
P4	1	0.211
P5	1	0.105
sum		2.111
convergence		8.000000

Query Processing

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Chapters 9 from Introduction to Information Retrieval, Manning et al. (2008)
Based on Chapter 5 from Information Retrieval - Implementing and Evaluating Search Engines, Büttcher et al. (2010)
Based on Chapters 5, 6 and 7 from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Outline

- ➔ Recap
- ➔ Query Processing
  - ➔ Processing strategies, document and term evaluation
  - ➔ Query transformation
  - ➔ Query expansion
  - ➔ Relevance feedback
- ➔ Search User Interface

Ranking Components

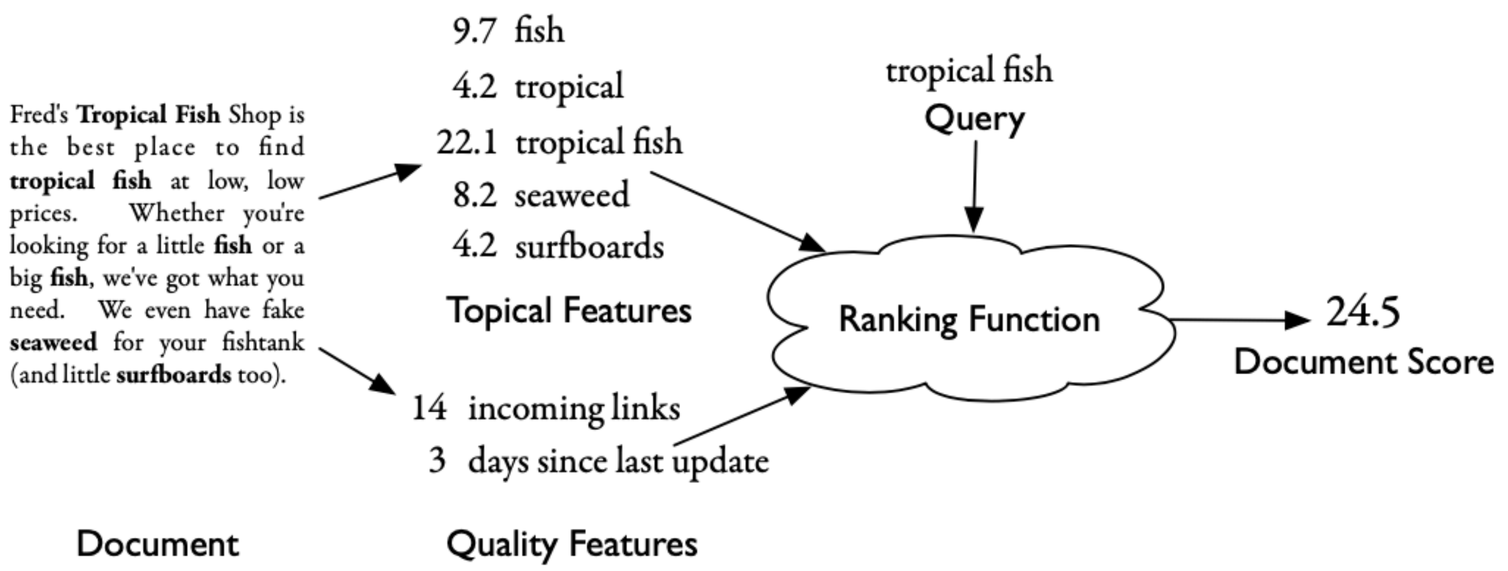


Fig. 5.1. The components of the abstract model of ranking: documents, features, queries, the retrieval function, and document scores

Image from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Inverted Index (with counts)

S_1 Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

S_2 Fishkeepers often use the term tropical fish to refer only those requiring fresh water, with saltwater tropical fish referred to as marine fish.

S_3 Tropical fish are popular aquarium fish, due to their often bright coloration.

S_4 In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Table 5.1. Four sentences from the Wikipedia entry for *tropical fish*

and	1:1			
aquarium	3:1			
are	3:1	4:1		
around	1:1			
as	2:1			
both	1:1			
bright	3:1			
coloration	3:1	4:1		
derives	4:1			
due	3:1			
environments	1:1			
fish	1:2	2:3	3:2	4:2
fishkeepers	2:1			
found	1:1			
fresh	2:1			
freshwater	1:1	4:1		
from	4:1			
generally	4:1			
in	1:1	4:1		
include	1:1			
including	1:1			
iridescence	4:1			
marine	2:1			
often	2:1	3:1		
only	2:1			
pigmented	4:1			
popular	3:1			
refer	2:1			
referred	2:1			
requiring	2:1			
salt	1:1	4:1		
saltwater	2:1			
species	1:1			
term	2:1			
the	1:1	2:1		
their	3:1			
this	4:1			
those	2:1			
to	2:2	3:1		
tropical	1:2	2:2	3:1	
typically	4:1			
use	2:1			
water	1:1	2:1	4:1	
while	4:1			
with	2:1			
world	1:1			

Fig. 5.4. An inverted index, with word counts, for the documents in Table 5.1

Image from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Inverted Index (with positions)

S_1 Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

S_2 Fishkeepers often use the term tropical fish to refer only those requiring fresh water, with saltwater tropical fish referred to as marine fish.

S_3 Tropical fish are popular aquarium fish, due to their often bright coloration.

S_4 In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Table 5.1. Four sentences from the Wikipedia entry for *tropical fish*

and	1,15					marine	2,22				
aquarium	3,5					often	2,2	3,10			
are	3,3	4,14				only	2,10				
around	1,9					pigmented	4,16				
as	2,21					popular	3,4				
both	1,13					refer	2,9				
bright	3,11					referred	2,19				
coloration	3,12	4,5				requiring	2,12				
derives	4,7					salt	1,16	4,11			
due	3,7					saltwater	2,16				
environments	1,8					species	1,18				
fish	1,2	1,4	2,7	2,18	2,23	term	2,5				
			3,2	3,6	4,3	the	1,10	2,4			
			4,13			their	3,9				
fishkeepers	2,1					this	4,4				
found	1,5					those	2,11				
fresh	2,13					to	2,8	2,20	3,8		
freshwater	1,14	4,2				tropical	1,1	1,7	2,6	2,17	3,1
from	4,8					typically	4,6				
generally	4,15					use	2,3				
in	1,6	4,1				water	1,17	2,14	4,12		
include	1,3					while	4,10				
including	1,12					with	2,15				
iridescence	4,9					world	1,11				

Fig. 5.5. An inverted index, with word positions, for the documents in Table 5.1

Image from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Query Processing

Query Processing

- Once the necessary data structures are in place, we need to efficiently use them to obtain the search results in response to a user query.

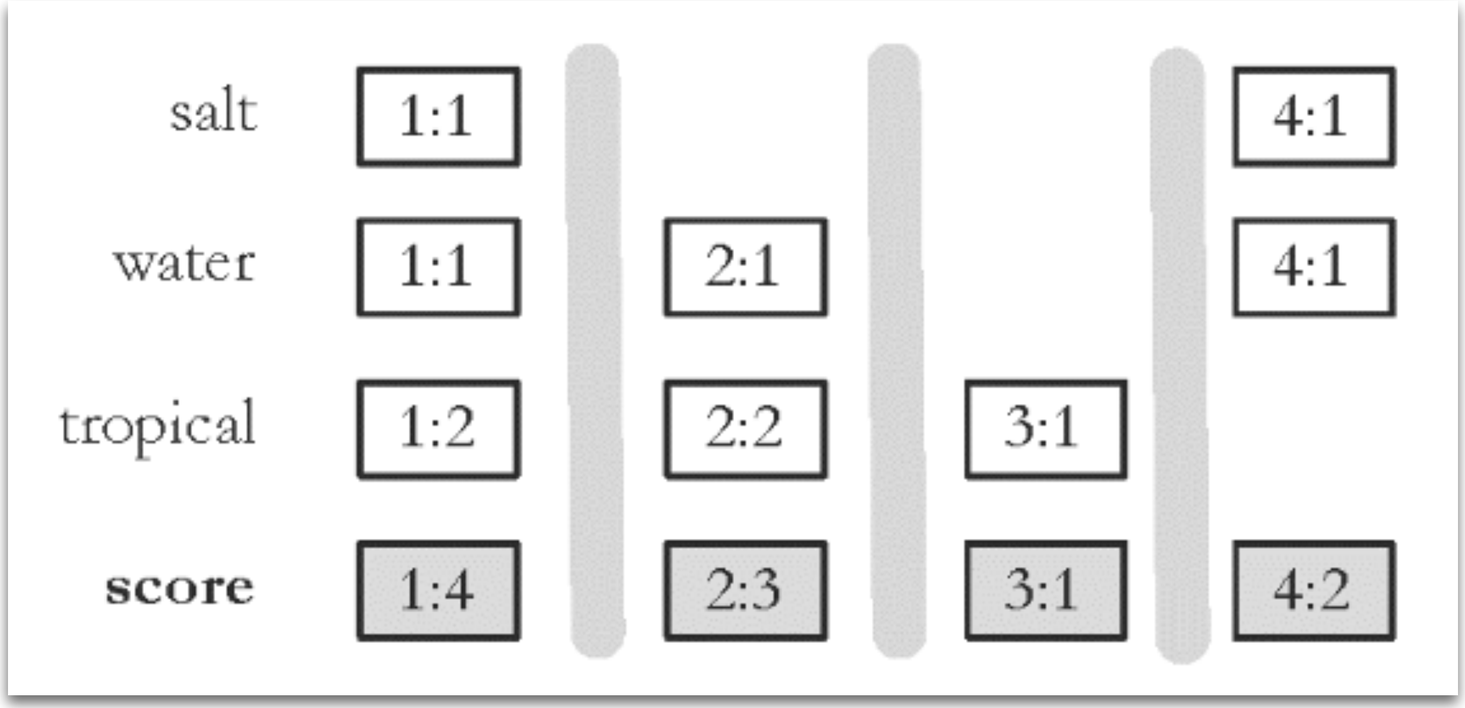
- Two main query processing techniques:
  - Document-at-a-time, calculates complete scores for documents by processing all term lists, one document at a time. At the end all documents are sorted according to their score.
  - Term-at-a-time, accumulates scores for documents by processing term lists one at a time. When all terms are processed, the accumulators contain the final scores of all matching documents.

- In both approaches, optimization techniques can significantly reduce the time required.

Document-at-a-Time

➔ query = [salt water tropical]

inverted lists



score is the sum of words in each document

each column represents a different document

1st step: compute all counts for the 1st document

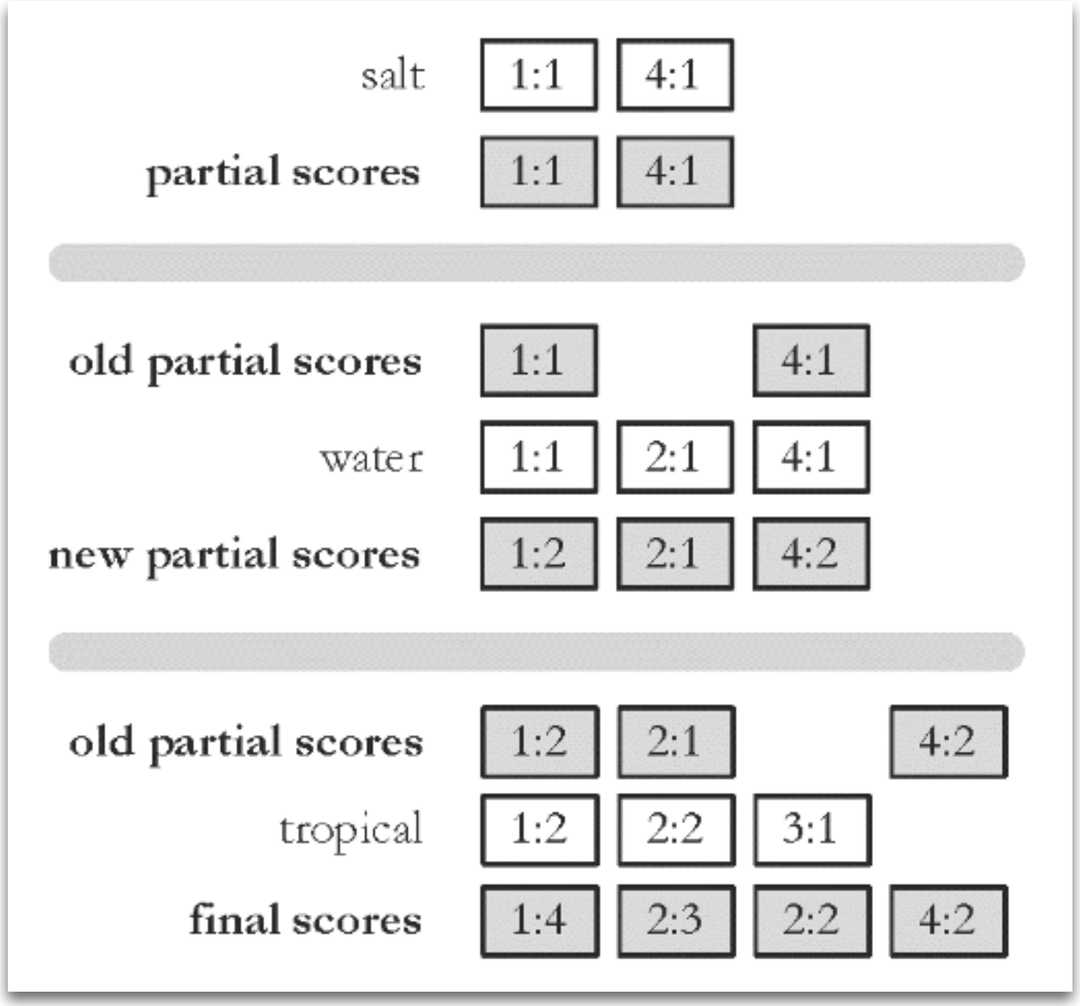
one document in each step

Image from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Term-at-a-Time

➔ query = [salt water tropical]

accumulators keep track of partial scores



1st step: compute all counts for the 1st term [salt]

one term is processed in each step

Image from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Optimization Techniques

- Two classes of optimization techniques for query processing:
  - Read less data from the index.
  - Process fewer documents.
- When using complex feature functions, focusing on scoring fewer documents is the main concert.
- With simple feature functions, performance gains come from ignoring as much of the inverted list data as possible.

Skip Pointers

➔ Skip pointers are use to speed-up inverted index scans.

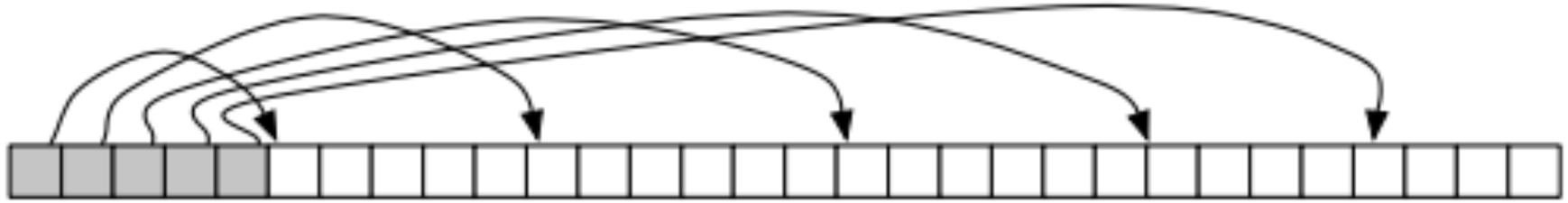


Fig. 5.19. Skip pointers in an inverted list. The gray boxes show skip pointers, which point into the white boxes, which are inverted list postings.

Image from Search Engines - Information Retrieval in Practice, Croft et al. (2015)

Conjunctive Processing

- Base assumption: only return documents that contain all query terms.
- This is the default in web search engines and the default users' expectations.
- Conjunctive processing works best when one of the terms is rare.
  - [wine baga], since 'baga' is rarer, we can skip most of the inverted list for 'wine'.
- Can be employed for both document-at-a-time or query-at-a-time.
- In short queries, benefits both efficiency and effectiveness.
- Long queries, e.g. phrase searches, are not good candidates for conjunctive processing.

Other Optimization Techniques

- Early termination of query processing:
  - Ignore high-frequency word lists in term-at-a-time,
    common terms have long postings lists, thus high processing costs.
  - Ignore documents at end of lists in document-at-a-time,
    when documents are ordered by some quality metric.
- Order postings in inverted indexes
  - Order inverted lists by quality metric (e.g. number of IN links in web IR).
- Caching
  - Cache popular query results

Relevance Feedback and Query Expansion

Relevance Feedback

- Exact matches aren't the only way to obtain relevant results in search systems.
- The vocabulary mismatch between the user and the collection contribute to this problem. Also, the fact that synonyms exist.
- For example, a search for [aircraft] should also include results for [airplane].
- This can be addressed by manually refining the query.
- On the system side, this can be tackled with different techniques, broadly grouped in:
  - Global methods, expand or reformulate the query terms independently of the query or the results returned from it, e.g. using thesaurus, and using spelling correction.
  - Local methods, adjust a query relative to the documents that initially appear to match the query, e.g. relevance feedback, and pseudo-relevance feedback.

Local Relevance Feedback Methods

Relevance Feedback

- The idea of relevance feedback is to involve the user in the process to improve the final result set by considering user feedback about the initial set of results.
- Basic procedure (one or more iterations are possible):
  - The user issues a (short, simple) query.
  - The system returns an initial set of retrieval results.
  - The user marks some returned documents as relevant or non relevant.
  - The system computes a better representation of the information need based on the user feedback.
  - The system displays a revised set of retrieved results.

(a) Query: New space satellite applications

(b)
+ 1. 0.539, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
+ 2. 0.533, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
3. 0.528, 04/04/90, Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
4. 0.526, 09/09/91, A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
5. 0.525, 07/24/90, Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
6. 0.524, 08/22/90, Report Provides Support for the Critics Of Using Big Satellites to Study Climate
7. 0.516, 04/13/87, Arianespace Receives Satellite Launch Pact From Telesat Canada
+ 8. 0.509, 12/02/87, Telecommunications Tale of Two Companies

(c)
2.074 new 15.106 space
30.816 satellite 5.660 application
5.991 nasa 5.196 eos
4.196 launch 3.972 aster
3.516 instrument 3.446 arianespace
3.004 bundespost 2.806 ss
2.790 rocket 2.053 scientist
2.003 broadcast 1.172 earth
0.836 oil 0.646 measure

(d)
* 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
* 2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
4. 0.493, 07/31/89, NASA Uses 'Warm' Superconductors For Fast Circuit
* 5. 0.492, 12/02/87, Telecommunications Tale of Two Companies
6. 0.491, 07/09/91, Soviets May Adapt Parts of SS-20 Missile For Commercial Use
7. 0.490, 07/12/88, Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
8. 0.490, 06/14/90, Rescue of Satellite By Space Agency To Cost $90 Million

Figure 9.2 Example of relevance feedback on a text collection. (a) The initial query. (b) The user marks some relevant documents (shown with a plus sign). (c) The query is then expanded by 18 terms with weights as shown. (d) The revised top results are then shown. A * marks the documents which were judged relevant in the relevance feedback phase.

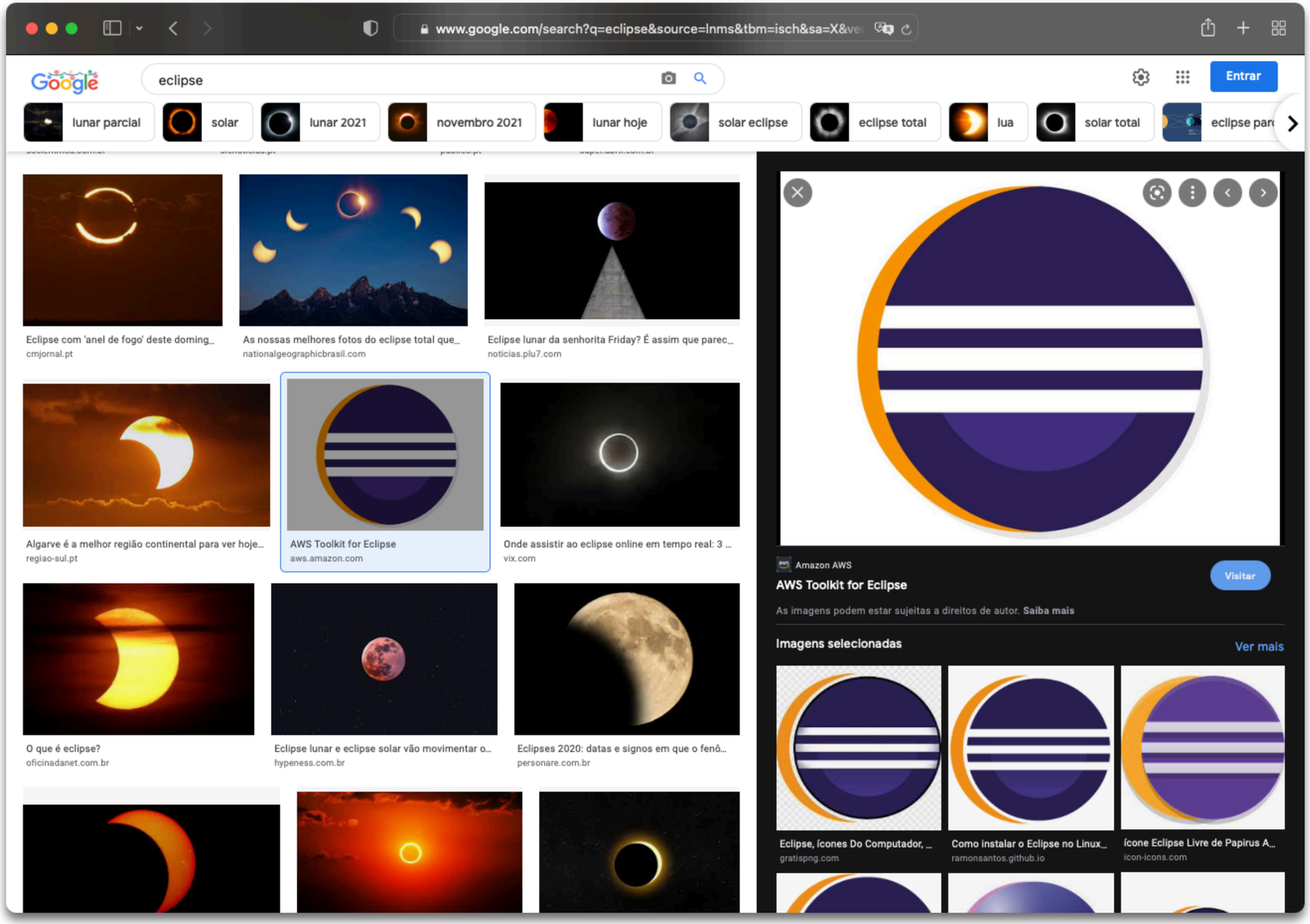
Image from Introduction to Information Retrieval, Manning et al. (2008)

Relevance Feedback

- ➔ Relevance feedback explores the idea that it may be difficult to formulate a good query when you don't known the collection, but it is easy to judge particular documents.

- ➔ Relevance feedback can be effective in tracking a user's evolving information need, i.e. seeing some documents may lead users to refine their understanding of the original information need.

- ➔ Image search is a good example of relevance feedback, a task where it may be difficult to express a query in words, but is easy to determine if an image is relevant.


www.google.com/search?q=eclipse&source=lnms&tbm=isch&sa=X&ve
Google
eclipse
Entrar
lunar parcial
solar
lunar 2021
novembro 2021
lunar hoje
solar eclipse
eclipse total
lua
solar total
Eclipse com 'anel de fogo' deste doming...
cmjornal.pt
As nossas melhores fotos do eclipse total que...
nationalgeographicbrasil.com
Eclipse lunar da senhorita Friday? É assim que parec...
noticias.plu7.com
Algarve é a melhor região continental para ver hoje...
regiao-sul.pt
AWS Toolkit for Eclipse
aws.amazon.com
Onde assistir ao eclipse online em tempo real: 3 ...
vix.com
O que é eclipse?
oficinadanet.com.br
Eclipse lunar e eclipse solar vão movimentar o...
hypeness.com.br
Eclipses 2020: datas e signos em que o fenô...
personare.com.br
Amazon AWS
AWS Toolkit for Eclipse
Visitar
As imagens podem estar sujeitas a direitos de autor. Saiba mais
Imagens selecionadas
Ver mais
Eclipse, ícones Do Computador, ...
gratispng.com
Como instalar o Eclipse no Linux...
ramonsantos.github.io
Ícone Eclipse Livre de Papirus A...
icon-icons.com

Rocchio Algorithm

- ➔ The Rocchio algorithm is the classic algorithm for implementing relevance feedback.
- ➔ It models a way of incorporating relevance feedback information into the vector space model.
- ➔ It is based on the concept of an optimal query vector, which maximizes the difference between the average vector representing the relevant documents, and the average vector representing the non-relevant documents.
- ➔ Considering that only partial relevance information is available about the collection, the Rocchio algorithm defines the "modified query" as a weighted combination of the initial query and the difference vector between the centroid of the documents marked as relevant and the centroid of the documents marked as non relevant.

Rocchio Optimal Query

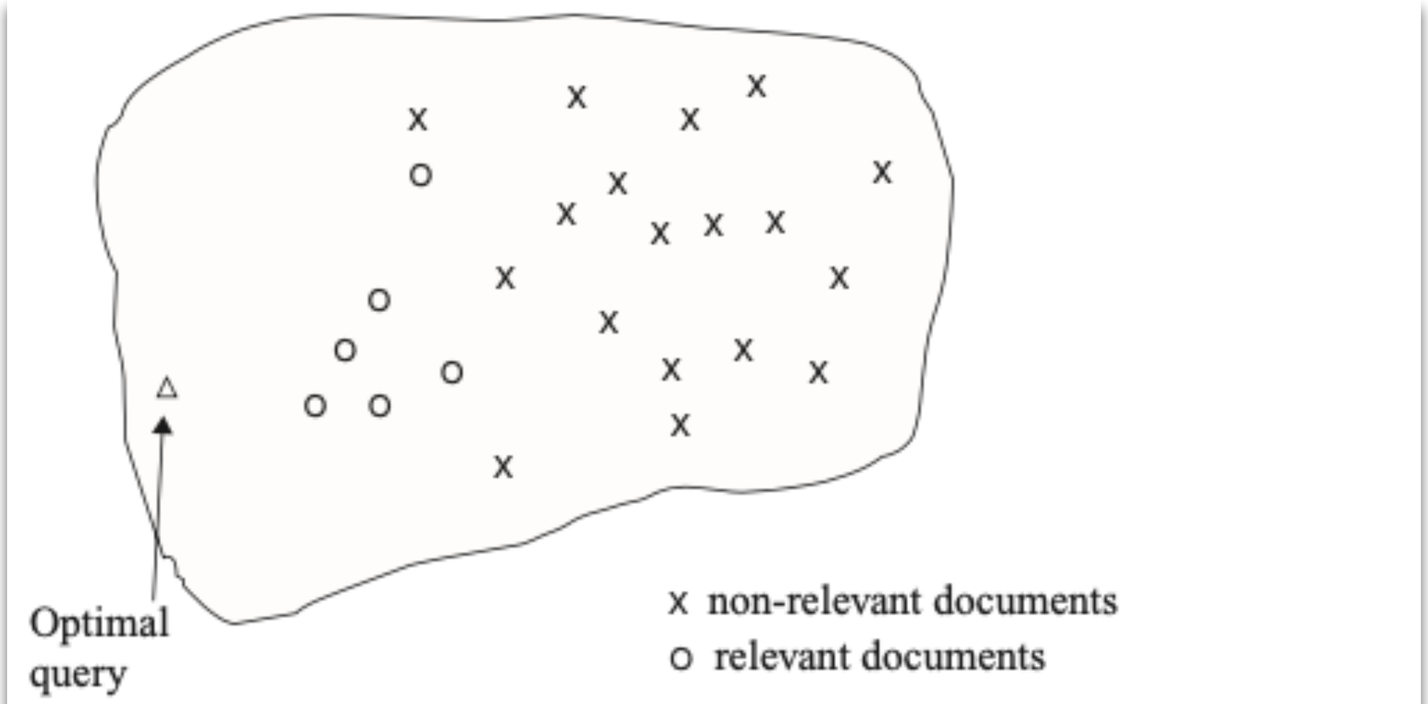


Figure 9.3 The Rocchio optimal query for separating relevant and nonrelevant documents.

Image from Introduction to Information Retrieval, Manning et al. (2008)

Rocchio Adjusted Query

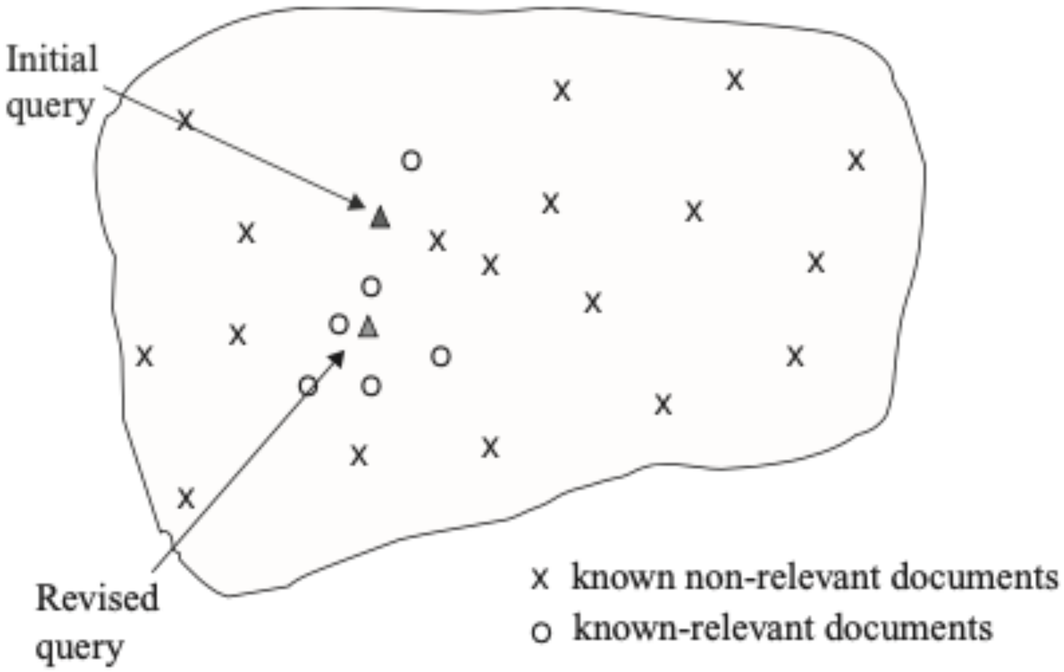


Figure 9.4 An application of Rocchio's algorithm. Some documents have been labeled as relevant and nonrelevant and the initial query vector is moved in response to this feedback.

Image from Introduction to Information Retrieval, Manning et al. (2008)

Rocchio Algorithm Performance

- ➔ Relevance feedback can improve both recall and precision.
- ➔ In practice it has been shown to be most useful for increasing recall — partially an effect of the use case: when a user wants high recall (i.e. see all relevant documents), they are expected to take the time to review results and iterate over the search.
- ➔ Positive feedback is more valuable than negative feedback. Many systems only allow for positive feedback.

Relevance Feedback Limitations

- Cases where relevance feedback might be insufficient:
  - Misspellings, if the user spells a term in a different way to the way it is spelled in the document collection, relevance feedback is unlikely to be effective.
  - Cross-language retrieval, documents in another language are not near (in a vector space) of other topic related documents, rather documents in the same language are closer.
  - Vocabulary mismatch, in this cases the initial query mostly likely will fail and thus relevant feedback won't be effective.
- Users are often reluctant to provide explicit feedback. Users expect single interactions in search and the concept of relevance feedback is hard to explain to the average user.
- Additionally, it is often harder to understand why a particular document was retrieved after relevance feedback was applied.

Pseudo Relevance Feedback

- ➔ Pseudo relevance feedback provides a method for automatic local analysis.
- ➔ It automates the manual part so that a relevance feedback algorithm is applied without extended user interaction.
- ➔ The method is applied to normal retrieval — assume that the top k ranked documents are relevant, and apply a relevance feedback algorithm under this assumption.

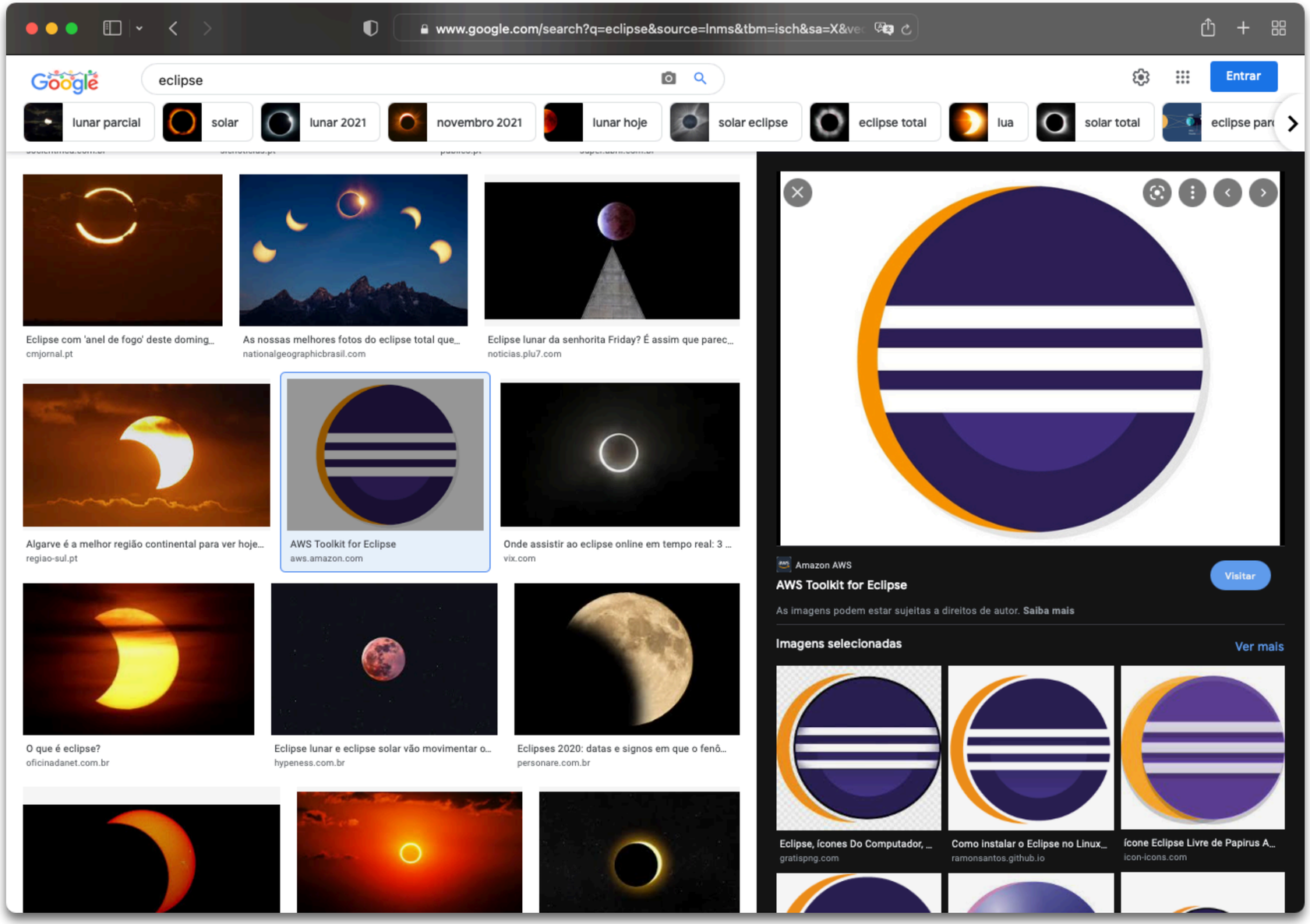
Implicit Relevance Feedback

- ➔ We can use indirect sources of evidence rather than explicit feedback on relevance.
- ➔ This is called implicit (relevance) feedback.
- ➔ Implicit feedback is less reliable than explicit feedback, but is more useful than pseudo relevance feedback, which contains no evidence of user judgements.
- ➔ Clickstreams are one of the main examples of indirect relevance information — clicks on links are assumed to indicate that the page was likely relevant for the query.

Global Query Reformulation Methods

Query Expansion

- With query expansion, user give additional input on query words or phrases to suggest additional terms. Users opt to use one of alternative query suggestions.
- How to generate alternative question expansions for the user:
  - Use synonyms and related words from a global thesaurus.
    - Use a controlled vocabulary to build a thesaurus;
    - Use a manual thesaurus built by editors;
    - Automatically derive thesaurus, e.g. use text statistics;
    - Use query log mining to find related expansions (global, contextual, user-based);


www.google.com/search?q=eclipse&source=lnms&tbm=isch&sa=X&ve
Google
eclipse
Entrar
lunar parcial
solar
lunar 2021
novembro 2021
lunar hoje
solar eclipse
eclipse total
lua
solar total
Eclipse com 'anel de fogo' deste doming...
cmjornal.pt
As nossas melhores fotos do eclipse total que...
nationalgeographicbrasil.com
Eclipse lunar da senhorita Friday? É assim que parec...
noticias.plu7.com
Algarve é a melhor região continental para ver hoje...
regiao-sul.pt
AWS Toolkit for Eclipse
aws.amazon.com
Onde assistir ao eclipse online em tempo real: 3 ...
vix.com
O que é eclipse?
oficinadanet.com.br
Eclipse lunar e eclipse solar vão movimentar o...
hypeness.com.br
Eclipses 2020: datas e signos em que o fenô...
personare.com.br
Amazon AWS
AWS Toolkit for Eclipse
Visitar
As imagens podem estar sujeitas a direitos de autor. Saiba mais
Imagens selecionadas
Ver mais
Eclipse, ícones Do Computador, ...
gratispng.com
Como instalar o Eclipse no Linux...
ramonsantos.github.io
Ícone Eclipse Livre de Papirus A...
icon-icons.com

www.google.com/search?q=igreja&bih=1017&biw=1538&hl=pt-PT&ei=2

Google

igreja

Entrar

Tudo | Maps | Imagens | Notícias | Vídeos | Mais | Ferramentas

Cerca de 466 000 000 resultados (0,93 segundos)

https://pt.wikipedia.org › wiki › Igreja

Igreja – Wikipédia, a enciclopédia livre

A **Igreja** é "o povo que Deus convoca e reúne de todos os confins da Terra, para constituir a assembleia daqueles que, pela fé e pelo Batismo, se tornam filhos de ...

Igreja (edifício) · Igreja Apostólica · Igreja Militante · Igreja orgânica

https://pt.wikipedia.org › wiki › Igreja_Católica

Igreja Católica – Wikipédia, a enciclopédia livre

A **Igreja** Católica ensina que é a **Igreja** única, santa, católica e apostólica fundada por Jesus Cristo em sua Grande Comissão, que seus bispos são os sucessores ...

Número de igrejas: + 221 mil paróquias
Fundador: Jesus, segundo a tradição cat...
Origem: Século I, Terra Santa, Império Rom...
Sede: Vaticano

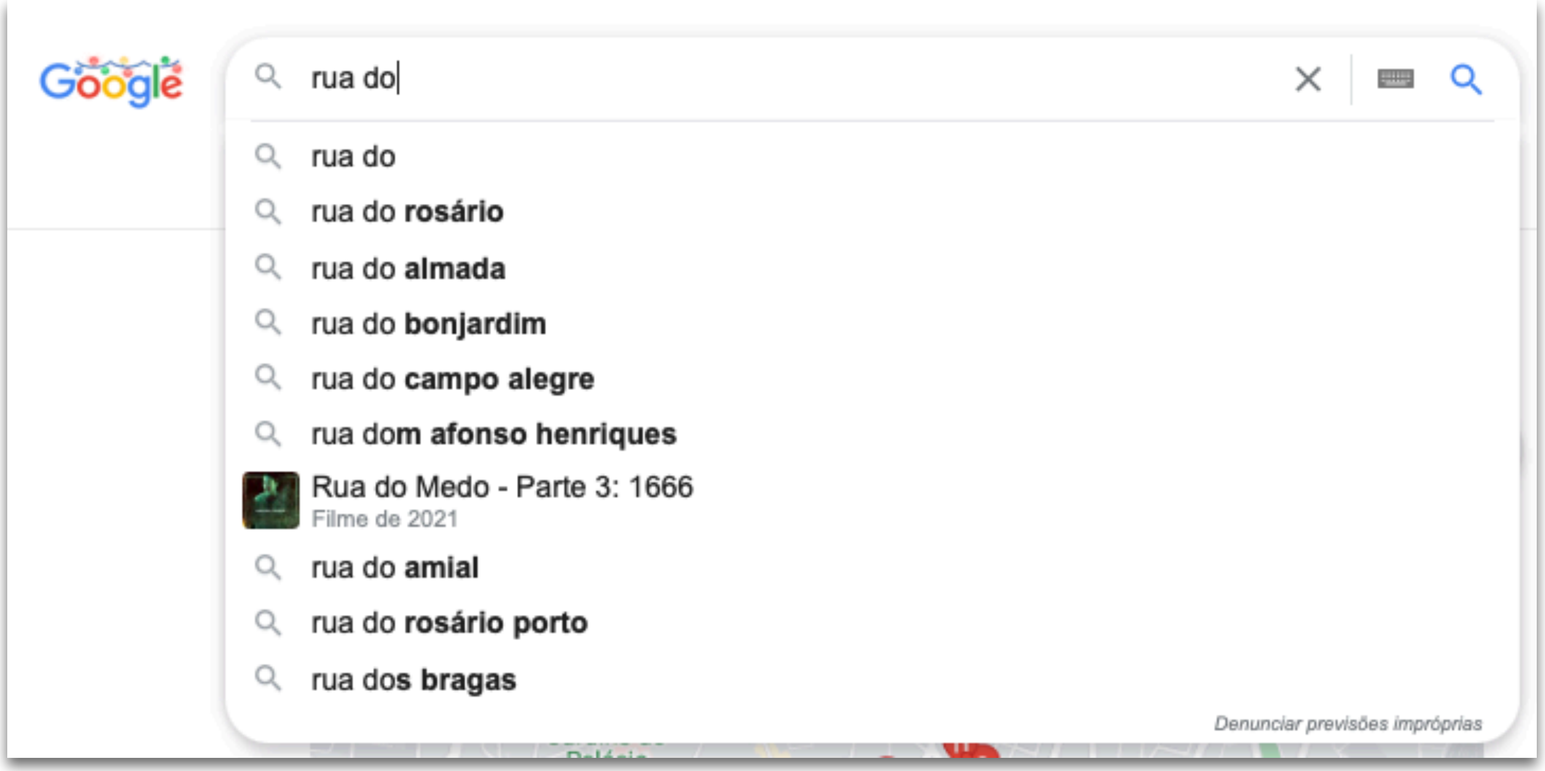
Igreja do Marquês
4000-391 Porto

https://www.tripadvisor.pt › Attractions-g189180-Activi...

Confira 10 Igrejas e catedrais em Porto - Tripadvisor

Queria dizer:

- **Igreja** — Igreja, no sentido genérico religioso, tem várias acepções ...
- **Igreja de Santo Ildefonso** — Igreja católica no Porto
- **Igreja Católica** — Igreja Católica, às vezes chamada de Igreja Católica ...
- **Igreja Cristã** — Função do edifício

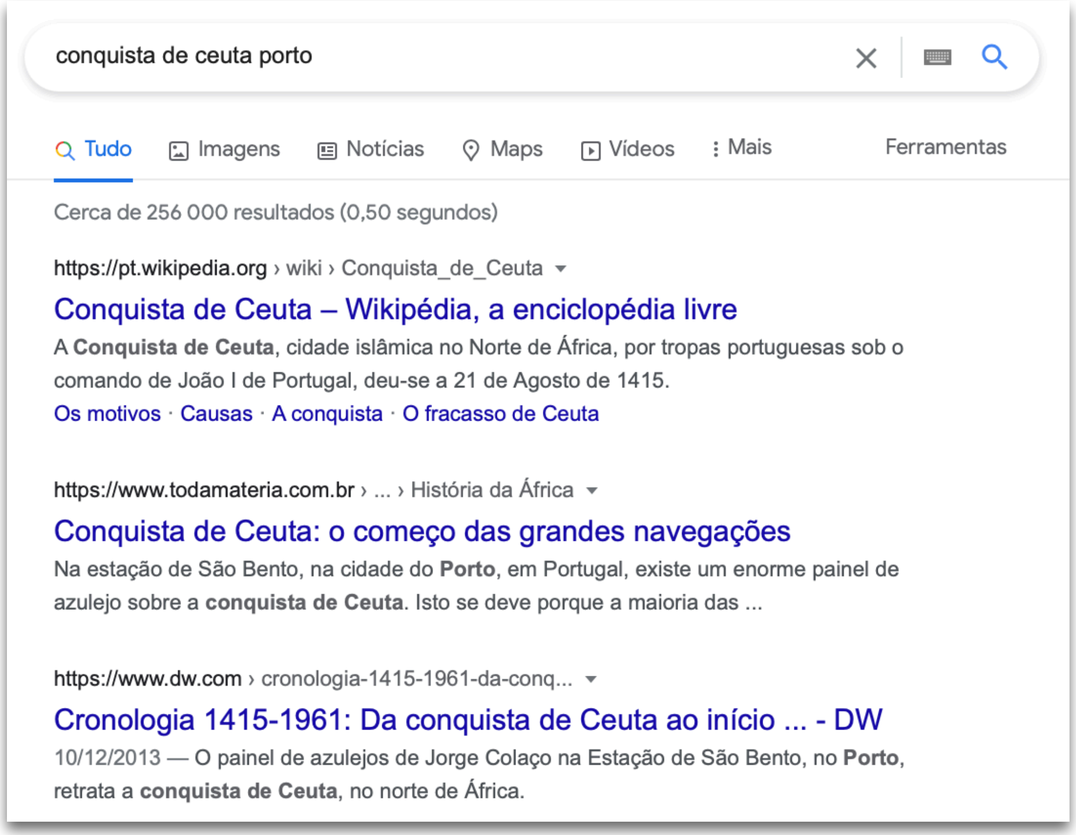

Google
rua do
rua do
rua do rosário
rua do almada
rua do bonjardim
rua do campo alegre
rua dom afonso henriques
Rua do Medo - Parte 3: 1666
Filme de 2021
rua do amial
rua do rosário porto
rua dos bragas
Denunciar previsões impróprias

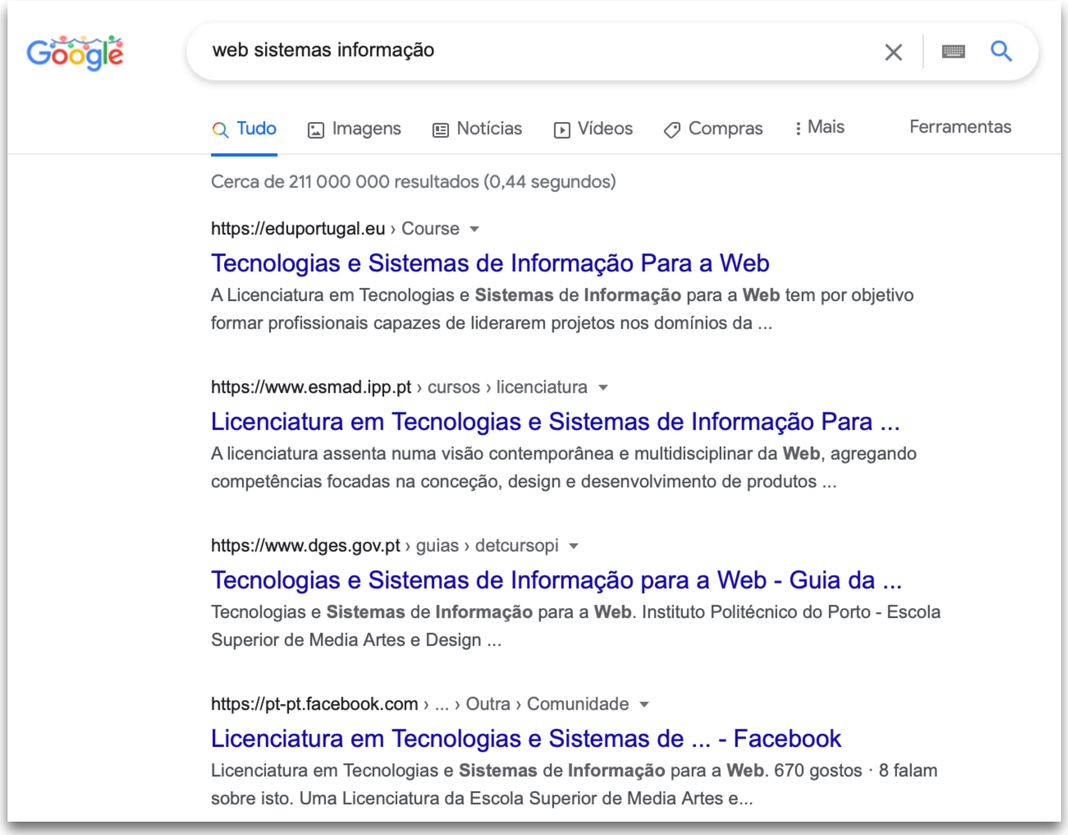
Search User Experience

Search User Experience

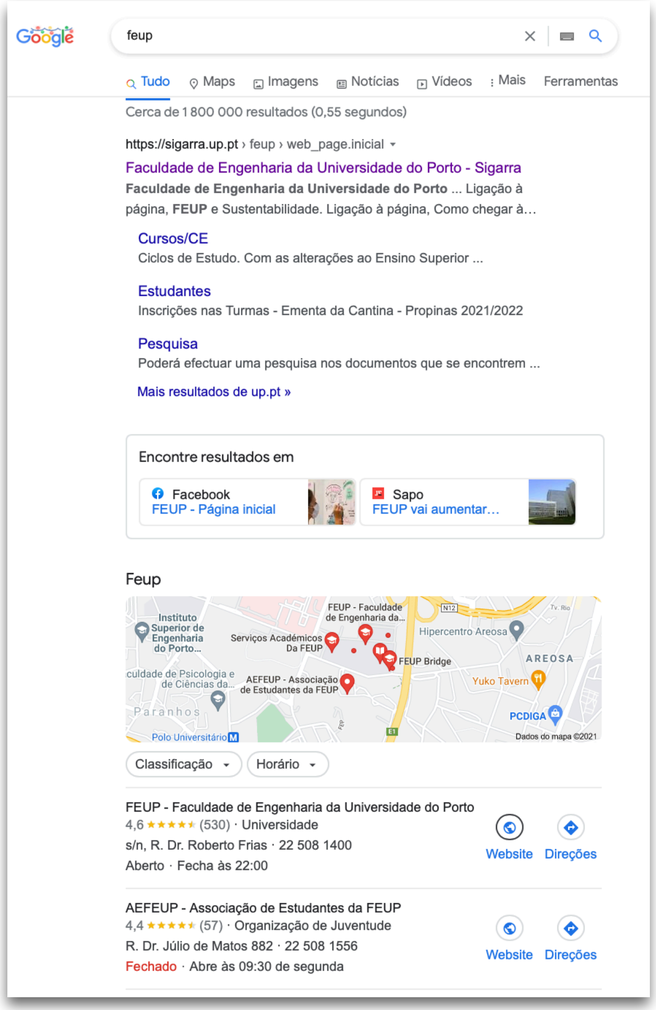
- Ranking is only a part of the information retrieval process, user interaction is central to the overall user experience and success of the user task.
- Designing the search user experience involves designing how the user interacts with the system during query formulation and reformulation, and while browsing the results.
- Example of search user experience techniques:
  - Support natural language queries.
  - Query auto-complete and suggestions;
  - Results snippets, e.g. query-dependent result snippets.
  - Clustering results.
  - Support site search.

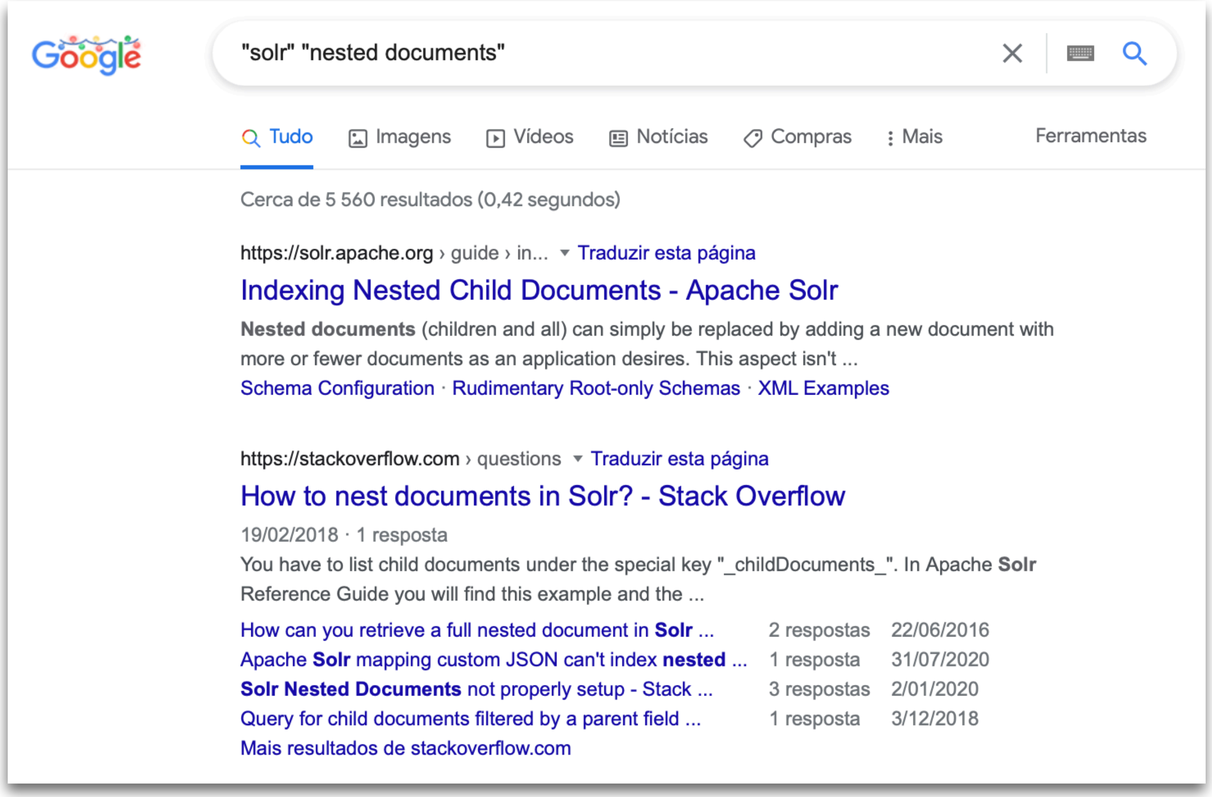
Contextual Snippets



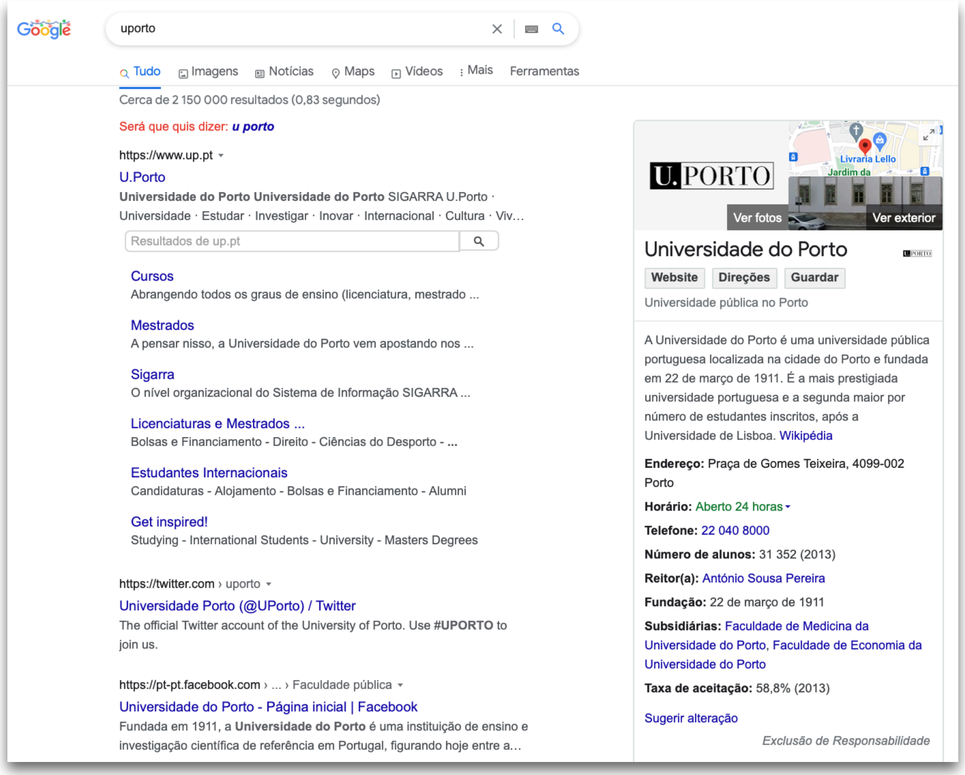


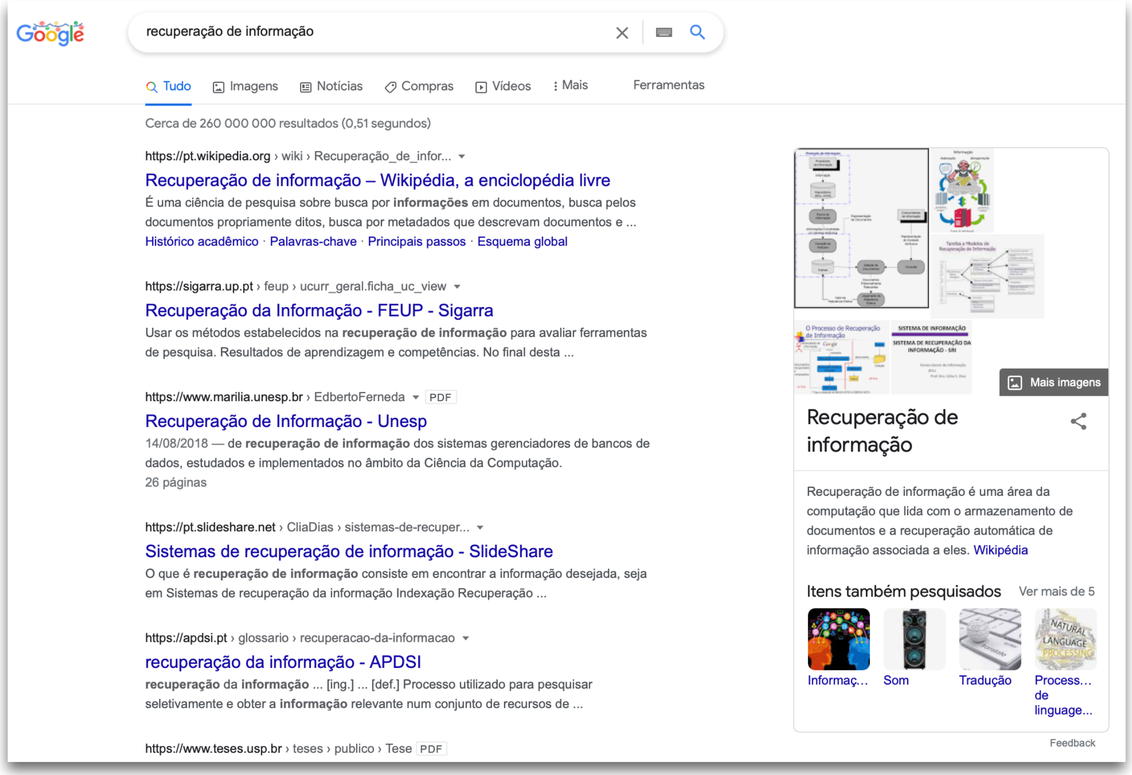
Clustering Results



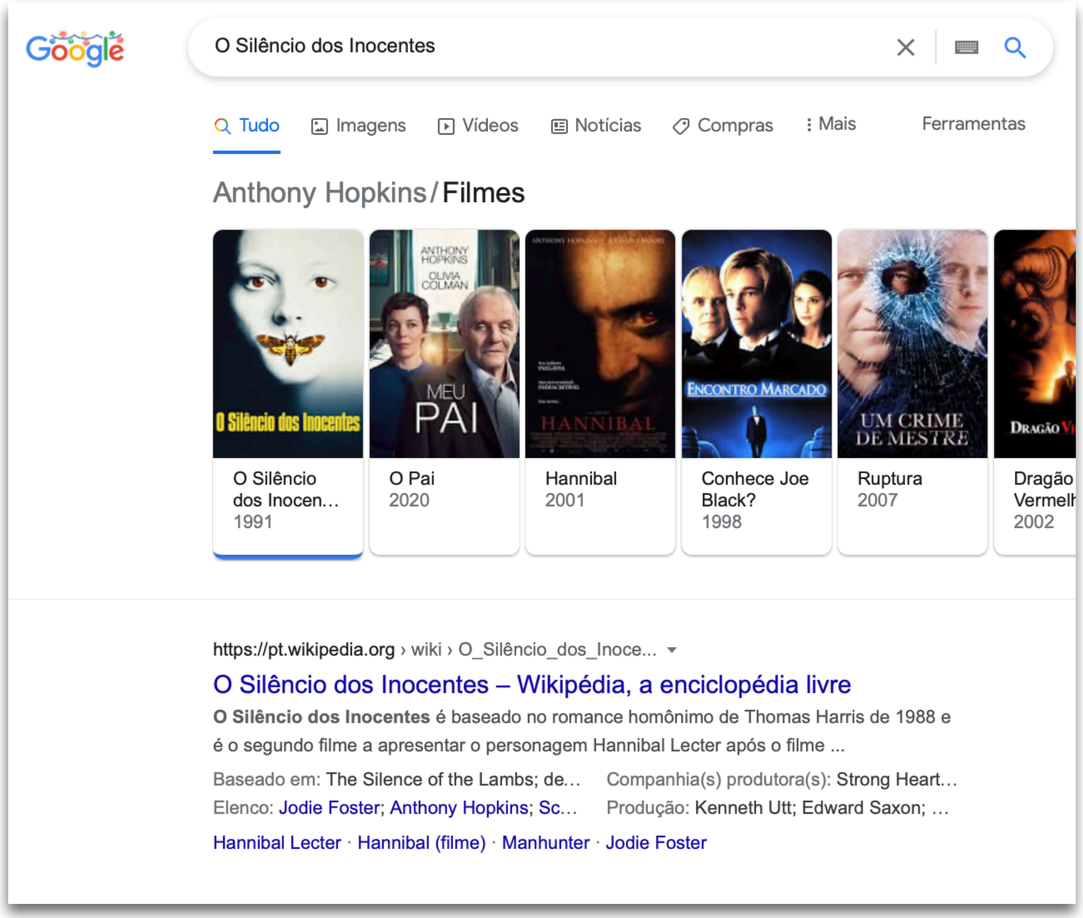


Infoboxes

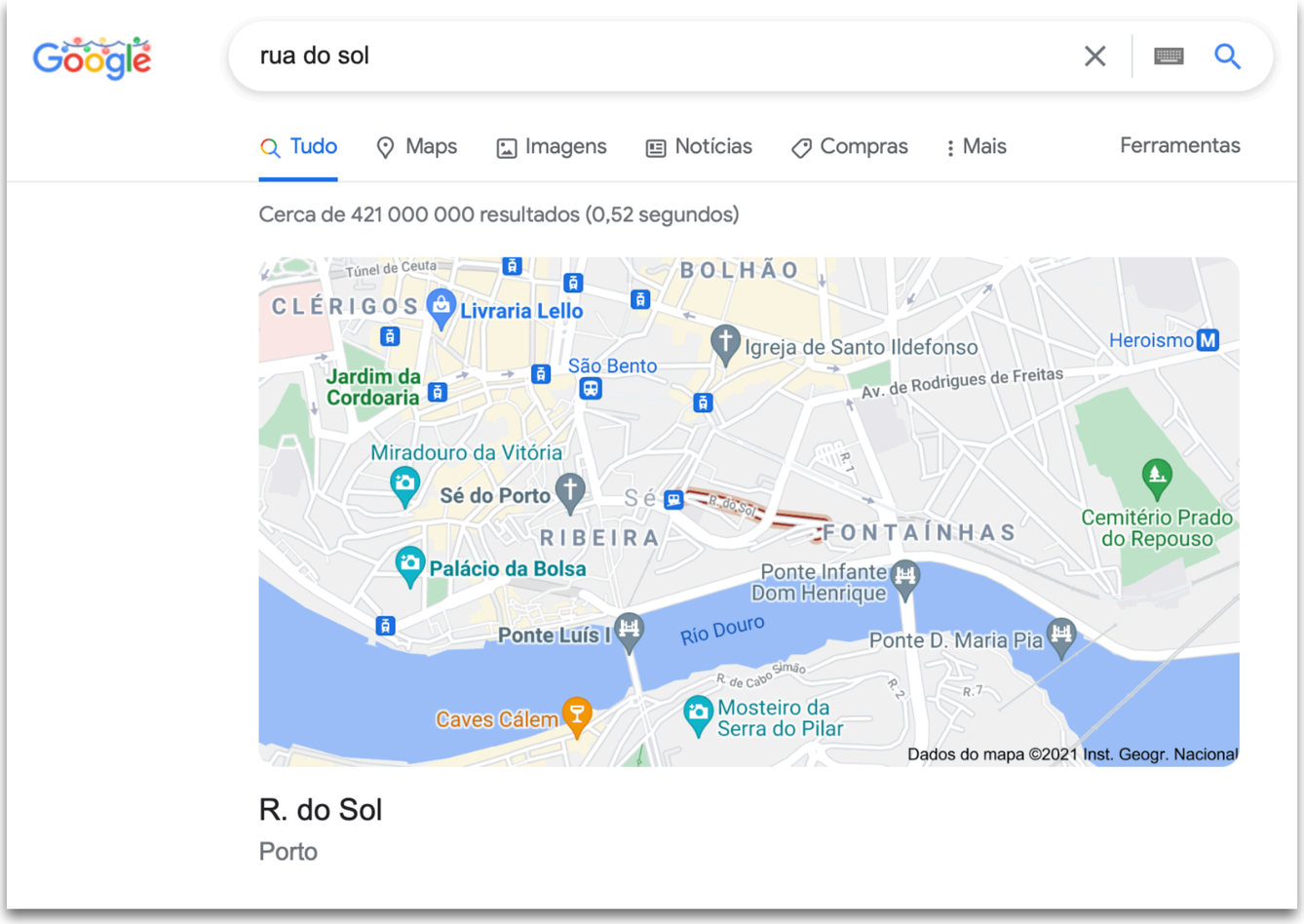

Google
uporto
Tudo
Imagens
Notícias
Maps
Vídeos
Mais
Ferramentas
Cerca de 2 150 000 resultados (0,83 segundos)
Será que quis dizer: u porto
https://www.up.pt
U.Porto
Universidade do Porto Universidade do Porto SIGARRA U.Porto ·
Universidade · Estudar · Investigar · Inovar · Internacional · Cultura · Viv...
Resultados de up.pt
Cursos
Abrangendo todos os graus de ensino (licenciatura, mestrado ...
Mestrados
A pensar nisso, a Universidade do Porto vem apostando nos ...
Sigarra
O nível organizacional do Sistema de Informação SIGARRA ...
Licenciaturas e Mestrados ...
Bolsas e Financiamento - Direito - Ciências do Desporto - ...
Estudantes Internacionais
Candidaturas - Alojamento - Bolsas e Financiamento - Alumni
Get inspired!
Studying - International Students - University - Masters Degrees
https://twitter.com › uporto
Universidade Porto (@UPorto) / Twitter
The official Twitter account of the University of Porto. Use #UPORTO to join us.
https://pt-pt.facebook.com › ... › Faculdade pública
Universidade do Porto - Página inicial | Facebook
Fundada em 1911, a Universidade do Porto é uma instituição de ensino e investigação científica de referência em Portugal, figurando hoje entre a...
U.PORTO
Livraria Lello
Jardim da
Ver fotos
Ver exterior
Universidade do Porto
Website
Direções
Guardar
Universidade pública no Porto
A Universidade do Porto é uma universidade pública portuguesa localizada na cidade do Porto e fundada em 22 de março de 1911. É a mais prestigiada universidade portuguesa e a segunda maior por número de estudantes inscritos, após a Universidade de Lisboa. Wikipédia
Endereço: Praça de Gomes Teixeira, 4099-002 Porto
Horário: Aberto 24 horas
Telefone: 22 040 8000
Número de alunos: 31 352 (2013)
Reitor(a): António Sousa Pereira
Fundação: 22 de março de 1911
Subsidiárias: Faculdade de Medicina da Universidade do Porto, Faculdade de Economia da Universidade do Porto
Taxa de aceitação: 58,8% (2013)
Sugerir alteração
Exclusão de Responsabilidade


Google
recuperação de informação
Tudo
Imagens
Notícias
Compras
Vídeos
Mais
Ferramentas
Cerca de 260 000 000 resultados (0,51 segundos)
https://pt.wikipedia.org › wiki › Recuperação_de_infor...
Recuperação de informação – Wikipédia, a enciclopédia livre
É uma ciência de pesquisa sobre busca por informações em documentos, busca pelos documentos propriamente ditos, busca por metadados que descrevam documentos e ...
Histórico acadêmico · Palavras-chave · Principais passos · Esquema global
https://sigarra.up.pt › feup › ucurr_geral.ficha_uc_view
Recuperação da Informação - FEUP - Sigarra
Usar os métodos estabelecidos na recuperação de informação para avaliar ferramentas de pesquisa. Resultados de aprendizagem e competências. No final desta ...
https://www.marilia.unesp.br › EdbertoFerneda PDF
Recuperação de Informação - Unesp
14/08/2018 — de recuperação de informação dos sistemas gerenciadores de bancos de dados, estudados e implementados no âmbito da Ciência da Computação.
26 páginas
https://pt.slideshare.net › CliaDias › sistemas-de-recuper...
Sistemas de recuperação de informação - SlideShare
O que é recuperação de informação consiste em encontrar a informação desejada, seja em Sistemas de recuperação da informação Indexação Recuperação ...
https://apdsi.pt › glossario › recuperacao-da-informacao
recuperação da informação - APDSI
recuperação da informação ... [ing.] ... [def.] Processo utilizado para pesquisar seletivamente e obter a informação relevante num conjunto de recursos de ...
https://www.teses.usp.br › teses › publico › Tese PDF
Mais imagens
Recuperação de informação
Recuperação de informação é uma área da computação que lida com o armazenamento de documentos e a recuperação automática de informação associada a eles. Wikipédia
Itens também pesquisados
Ver mais de 5
Informaç...
Som
Tradução
Process... de linguage...
Feedback

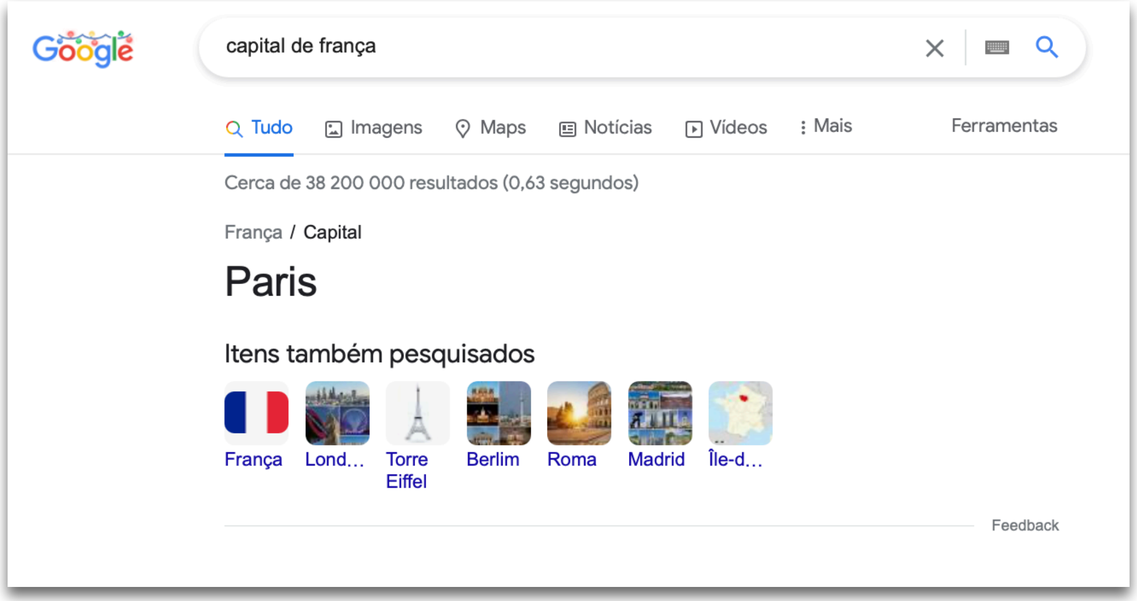
Results Browsing

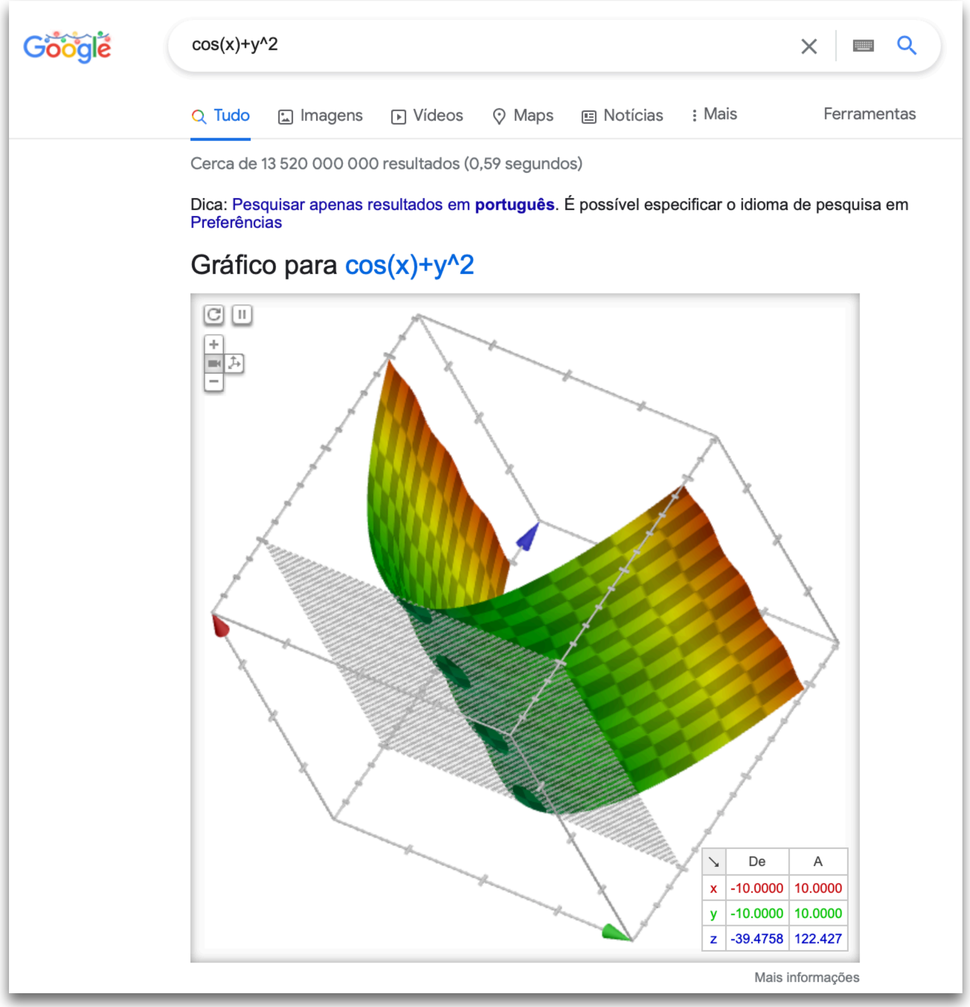


Geographical Results

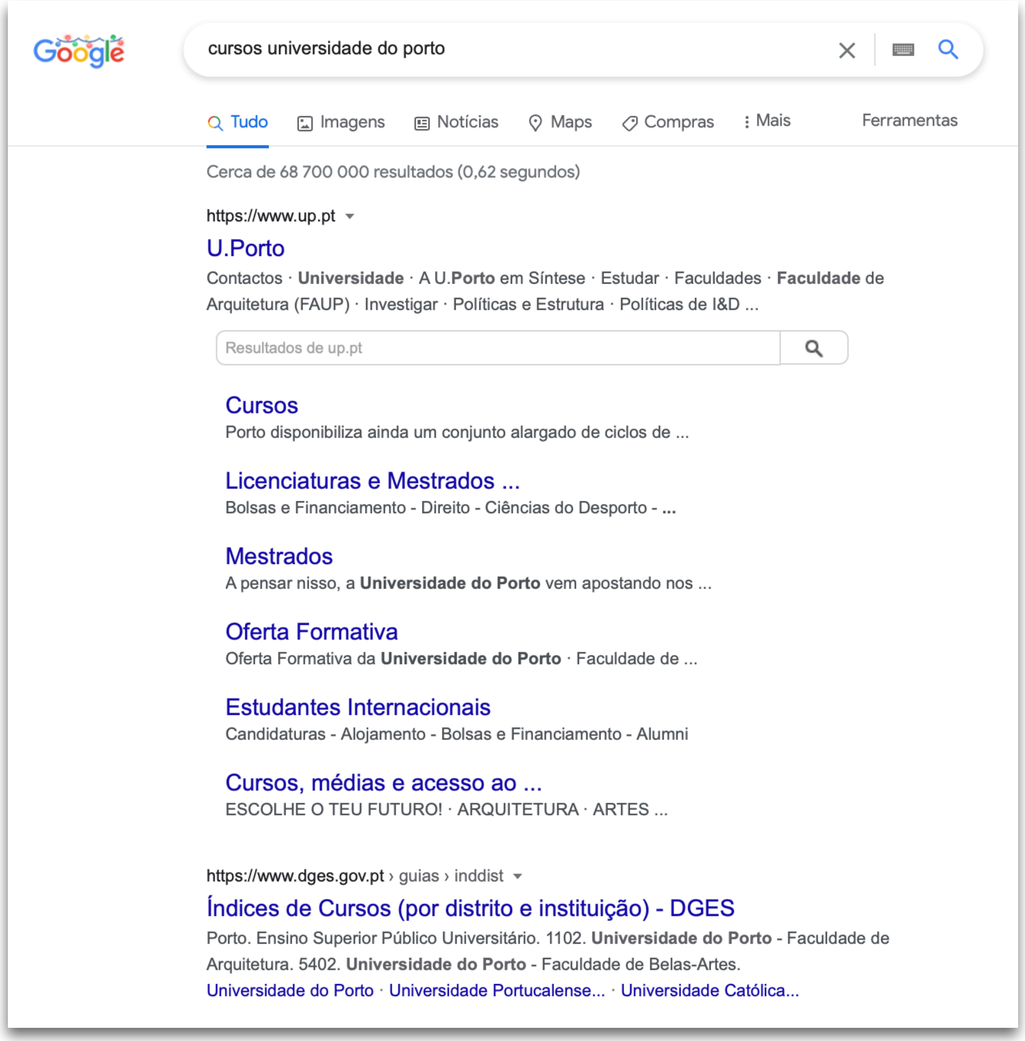


Immediate Answers





Domain Specific Search



References

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval
  - Chapter 9: Relevance feedback and query expansion
- W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines - Information Retrieval in Practice
  - Chapter 5: Ranking with Indexes + Slides
  - Chapter 6: Queries and Interfaces + Slides
  - Chapter 7: Retrieval Models + Slides
- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search
  - Chapter 2: User Interfaces for Search + Slides

Entity-Oriented Search

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Entity-Oriented Search, Springer, K. Balog (2018)

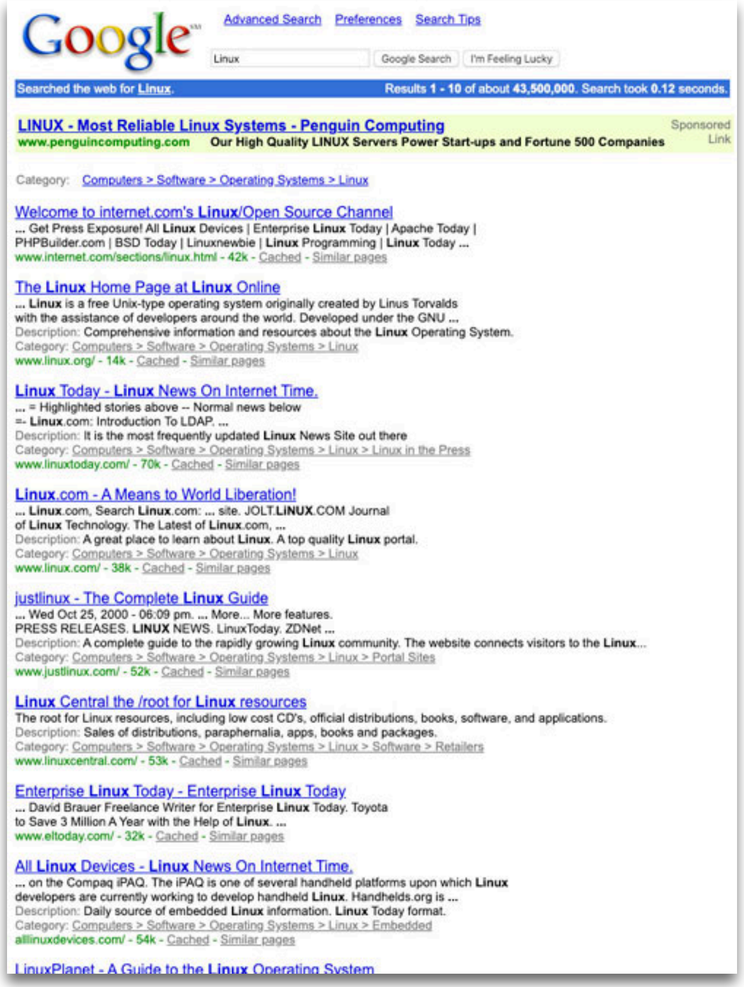
Outline

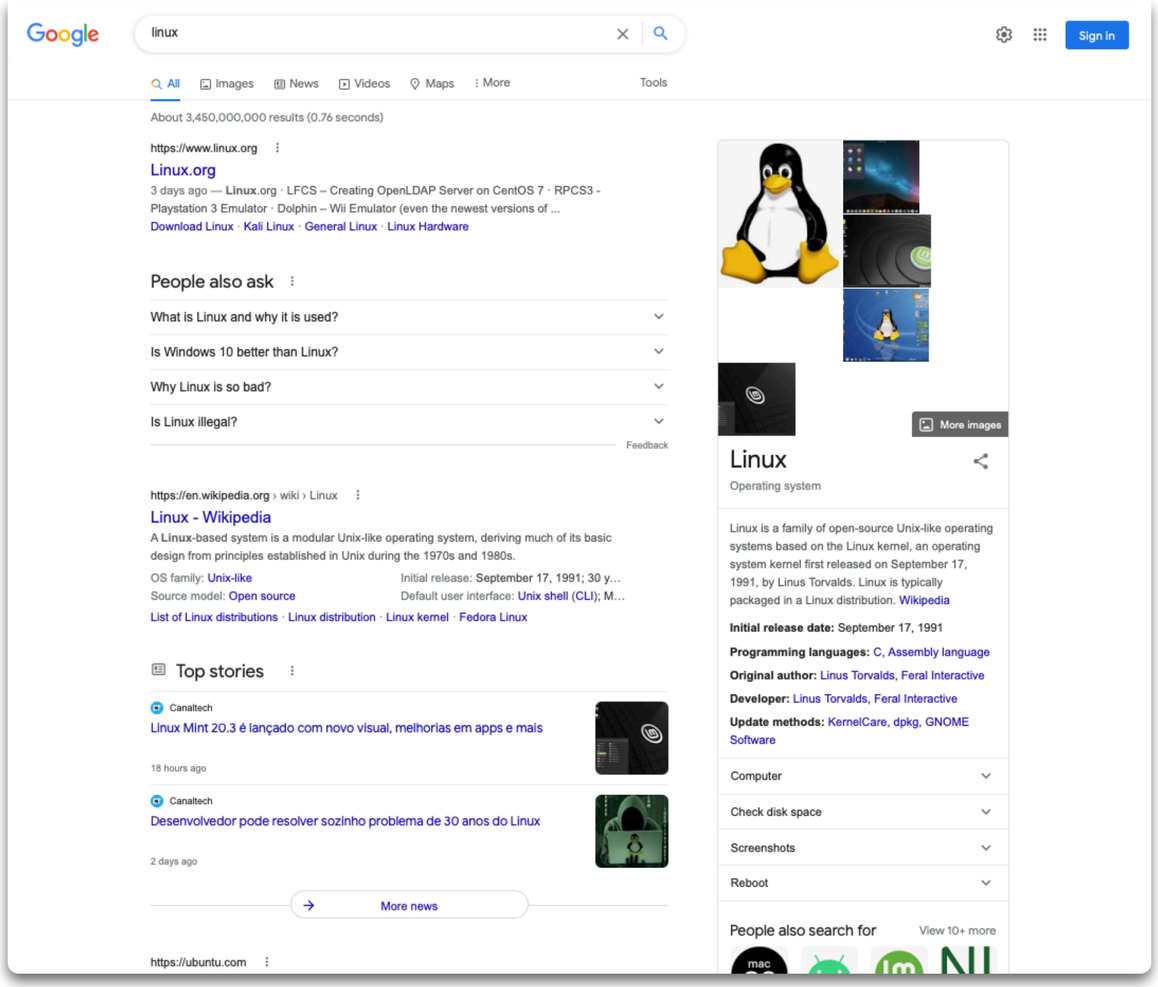
- ➔ Overview of entity-oriented search
- ➔ Types of data sources
- ➔ Entity ranking
- ➔ Entity linking (briefly)

Entity-Oriented Search

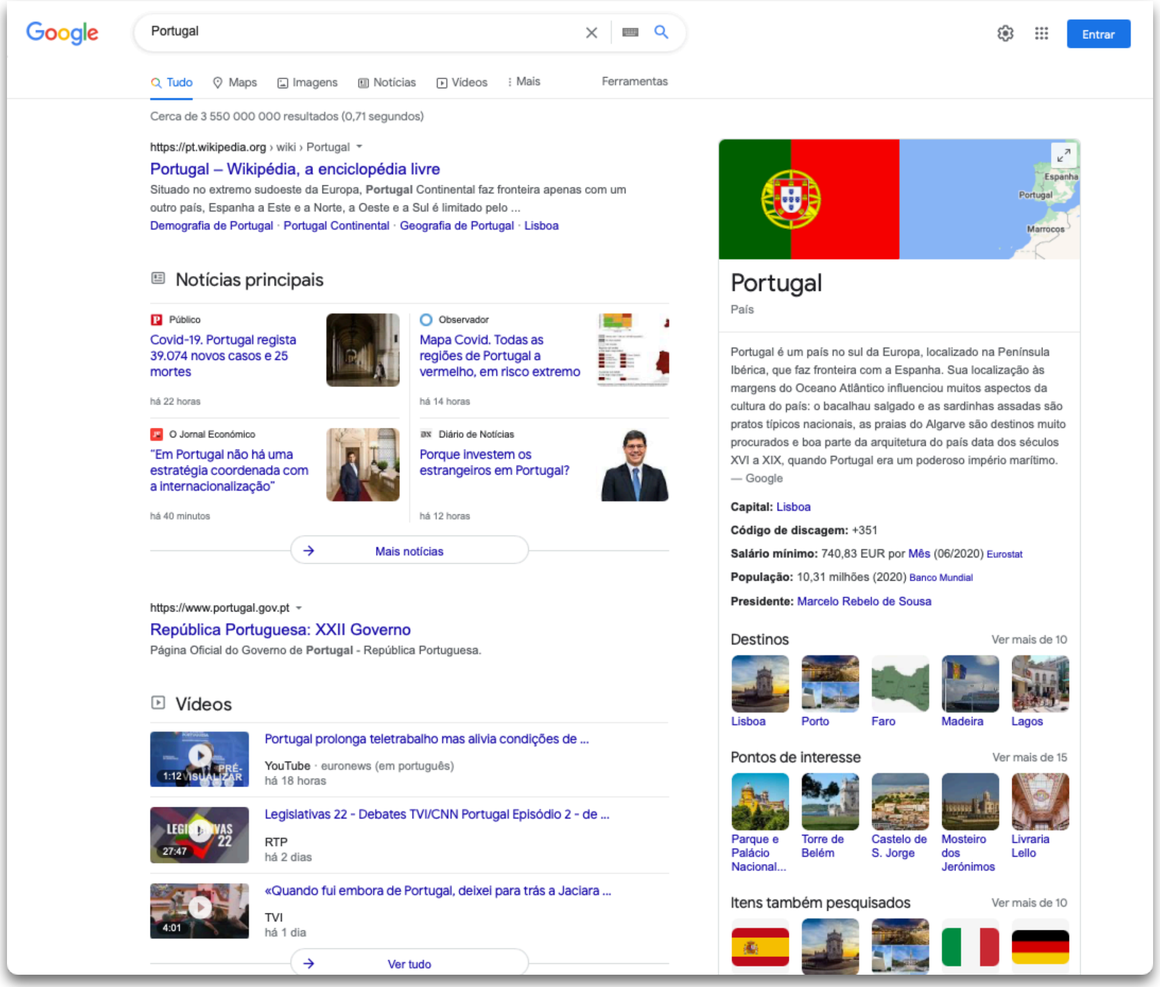
Context

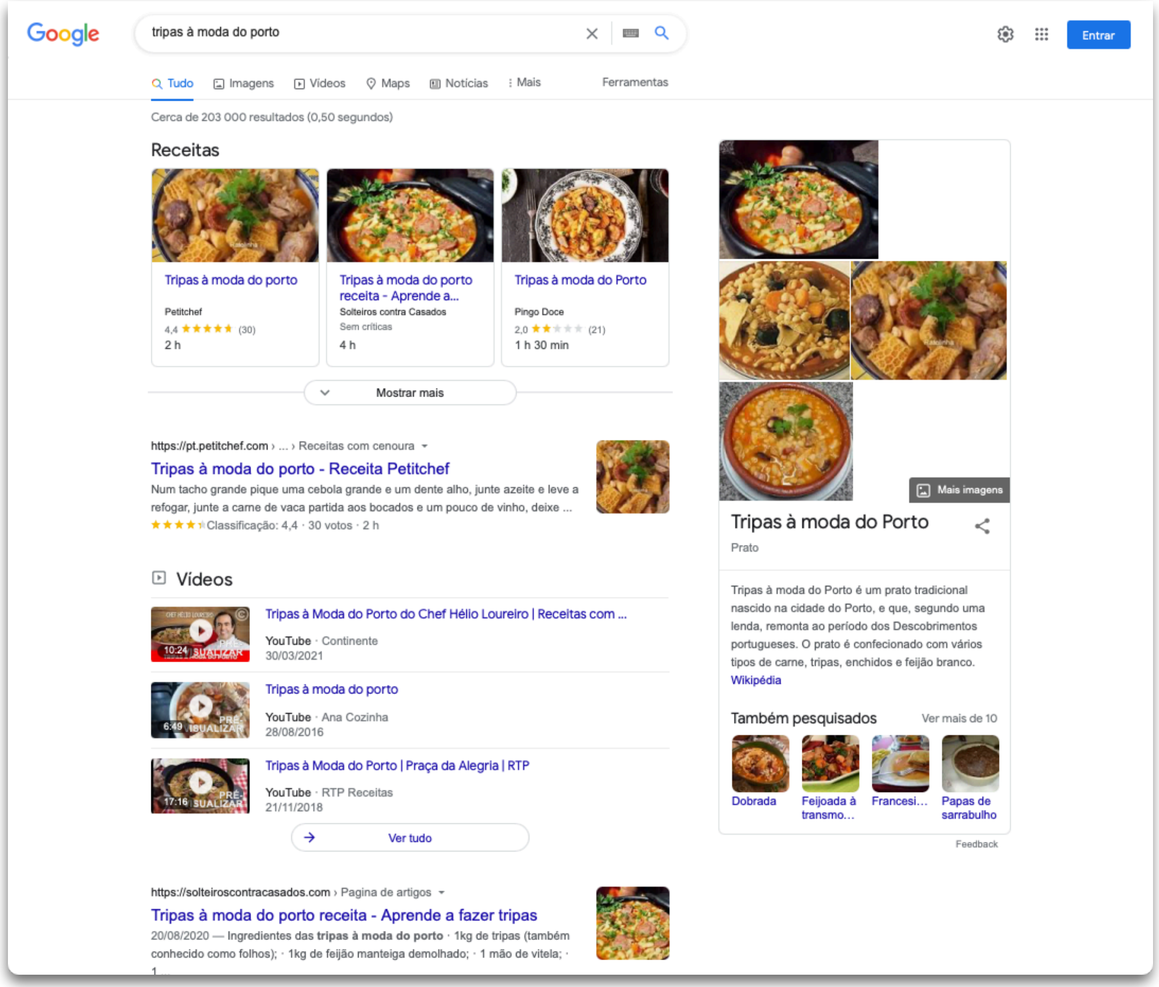
- ➔ Search is pervasive — on the web, on websites, on applications, on devices.
- ➔ Users expect to find a search box to help them find the information they need.
- ➔ Information retrieval deals with matching <u>information needs</u> with <u>information objects</u>.
- ➔ Queries are the expression of information needs.
- ➔ Queries may range from simple keywords [ps5 games] to full sentences ["Where can I buy a Playstation 5?"].
- ➔ The search engine responds with a ranked list of information objects.
- ➔ Traditionally these information objects were documents, but today's search engines return rich results pages beyond the classic "ten blue links" — *semantic search*.



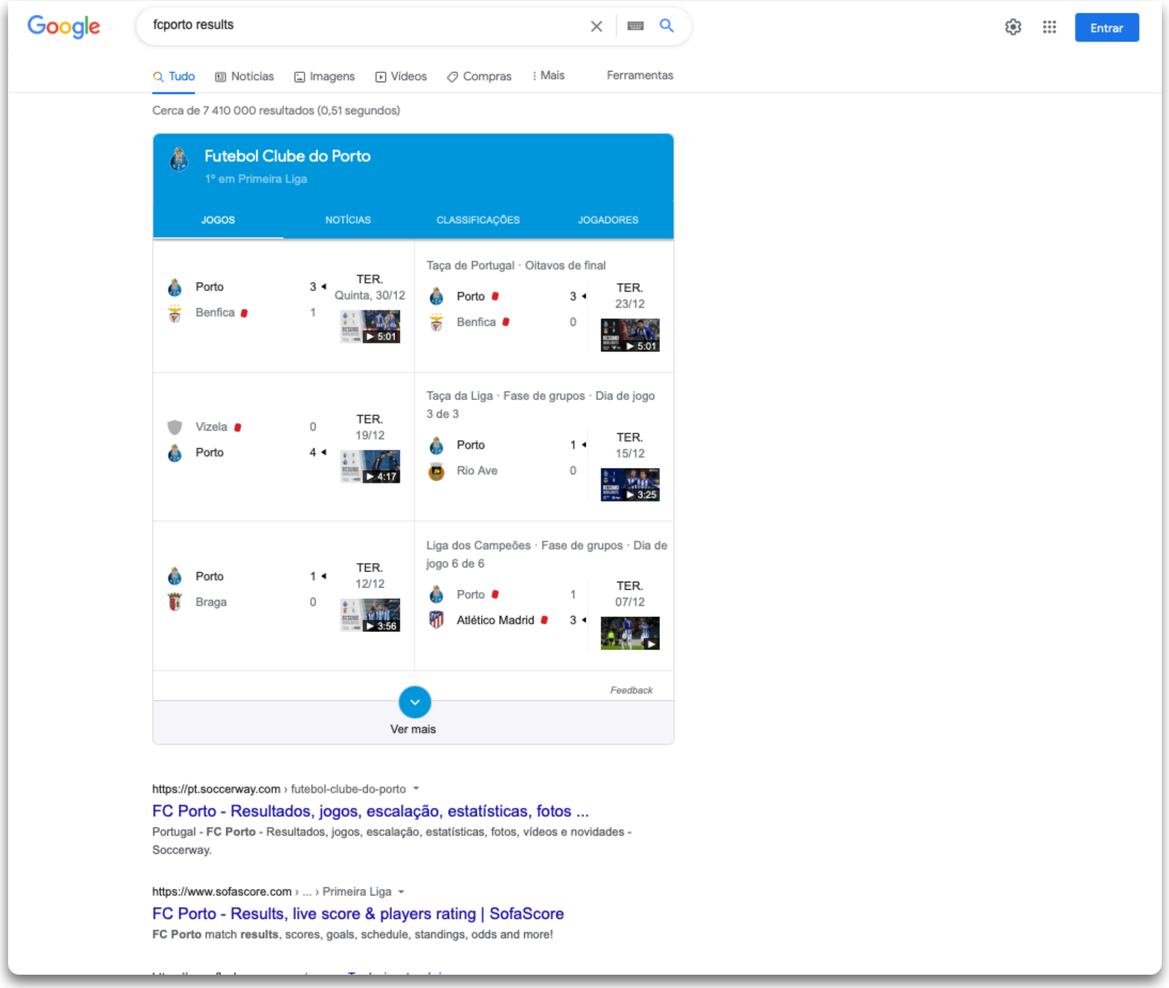


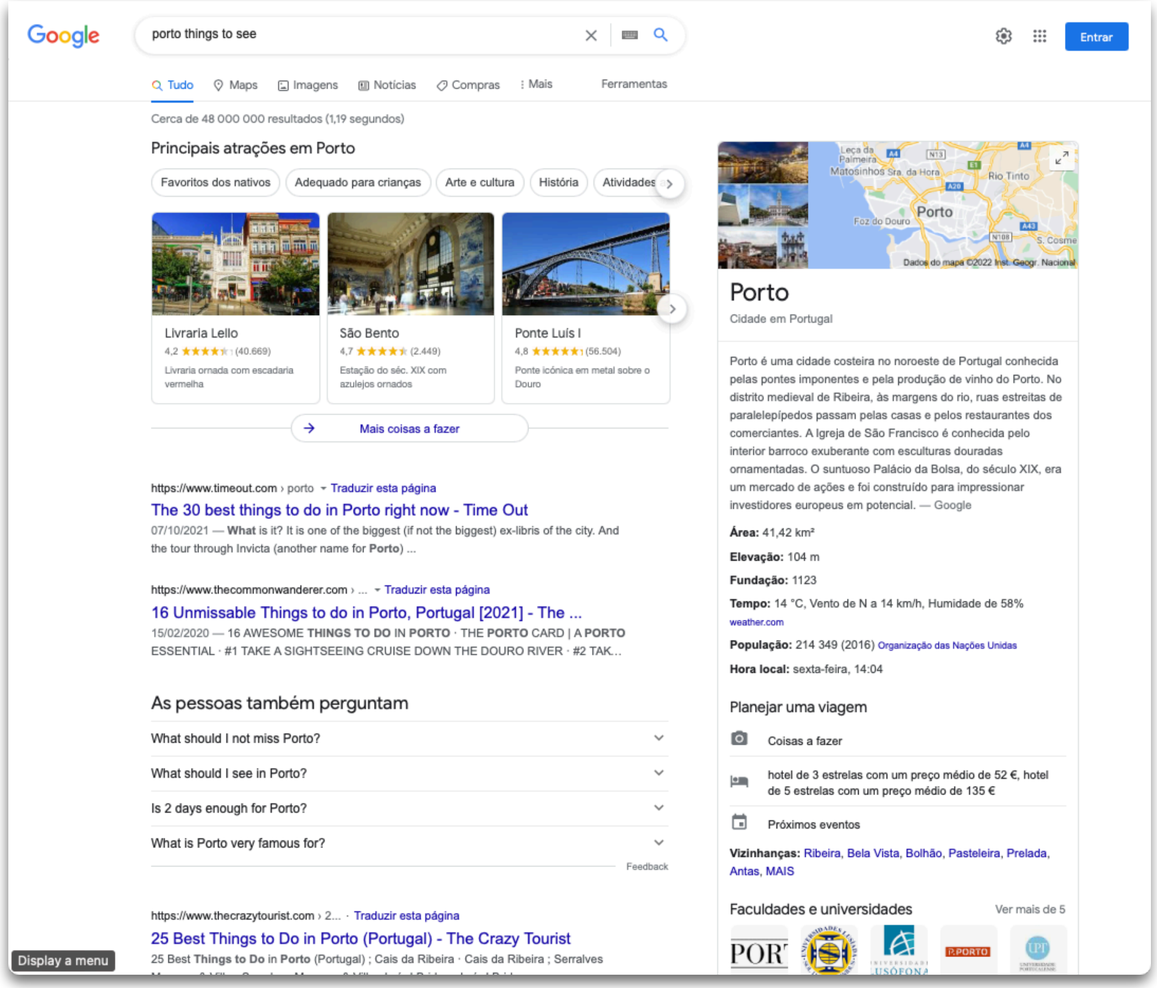
Google search for ["linux"] circa 2000 and today (2021).





Rich search engine results pages (SERPs), containing links to documents, information boxes, direct display of entities, facts, and other structured results.





More examples of "semantic search".

Knowledge Bases and Entities

- ➔ A primary component enabling these advanced search services is the availability of large-scale structured knowledge repositories, called knowledge bases.
- ➔ Knowledge bases organize information around specify things or objects, called "entities".
- ➔ An entity is a uniquely identifiable object or thing, characterized by its name(s), type(s), attributes, and relationships to other entities.
- ➔ Common types of entities include: people, organizations, products, locations, and events.

Properties of Entities

- **Unique identifier**: entities need to be uniquely identifiable. There must be a one-to-one correspondence between each entity identifier and the object it represents. Examples include: username, email address, URI, Wikipedia page ID, etc.

- **Name(s)**: entities are known and referred to by their name. Unlike IDs, names do not uniquely identify entities; multiple entities may share the same name; and the same entity may be known by more than a single name.

- **Type(s)**: entities may be categorized into multiple entities types. Types work as semantic categories that group together entities with similar properties. The set of possible entity type is often organized in a hierarchical structure, e.g. "scientist" is a subtype of "person".

- **Attributes**: the characteristics or features of an entity are described by a set of attributes. Different entities have different sets of attributes. Some of the characteristics may be entities themselves, in this we cases they are not treated as attributes but as relationships, e.g. "place of birth" links to a location entity. Attributes always have literal values and optionally may also include data type information.

- **Relationships**: describe how two entities are associated to each other. Relationships may also be seen as "typed links" between entities.

Representing Properties of Entities

- Information about entities can be represented and stored in semi-structured or in structured form.
  - Wikipedia is an example of a knowledge repository that organizes information about entities and their attributes and relationships in a semi-structured form.
  - To adopt a structured form, a knowledge representation model is necessary. The Resource Description Framework (RDF) is a standard way to describe entities, where an entity is represented as a set of RDF statements.
- A knowledge base (KB) is a structured knowledge repository that contains a set of facts (assertions) about entities.
- Entities in a knowledge base may be seen as nodes in a graph, with relationships between them as edges. Thus, knowledge bases are also referred to as knowledge graphs.

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: University of Porto

An Entity of Type: Public university, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

The University of Porto (Universidade do Porto) is a Portuguese public university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of Lisbon, and has one of the most noted research outputs in Portugal.

Property	Value
dbo:abstract	• The University of Porto (Universidade do Porto) is a Portuguese public university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of Lisbon, and has one of the most noted research outputs in Portugal. (en)
dbo:affiliation	• dbr:CESAER
dbo:city	• dbr:Porto
dbo:country	• dbr:Portugal
dbo:motto	• Virtus Unita Fortius Agit
dbo:numberOfPostgraduateStudents	• 8235 (xsd:nonNegativeInteger)
dbo:numberOfStudents	• 30640 (xsd:nonNegativeInteger)
dbo:numberOfUndergraduateStudents	• 22405 (xsd:nonNegativeInteger)
dbo:officialSchoolColour	• Gold
dbo:other	• 742 (xsd:integer)
dbo:thumbnail	• wiki-commons:Special:FilePath/Fonte_dos_leões_e_reitoria.jpg?width=300

Example of the "University of Porto" entity in DBpedia
https://dbpedia.org/page/University_of_Porto

```
<http://dbpedia.org/resource/University_of_Porto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/University> .
<http://dbpedia.org/resource/University_of_Porto> <http://xmlns.com/foaf/0.1/homepage>  <https://sigarra.up.pt/up/en/WEB_PAGE.INICIAL%7C2=www.up.pt> .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/Portugal> .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/city>  <http://dbpedia.org/resource/Porto> .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/wikiPageUsesTemplate>  <http://dbpedia.org/resource/Template:Color_box> .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/wikiPageWikiLink>  <http://dbpedia.org/resource/Higher_education_in_Portugal> .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/rector>  "António Sousa Pereira"@en .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/motto> "Virtus Unita Fortius Agit" .
<http://dbpedia.org/resource/University_of_Porto> <http://xmlns.com/foaf/0.1/name>  "Universidade do Porto"@en .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/abstract>  "The University of Porto (Universidade do Porto) is a Portuguese public
university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of
Lisbon, and has one of the most noted research outputs in Portugal."@en .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/country> <http://dbpedia.org/resource/Portugal> .
<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/nativeName>  "Universidade do Porto"@en .
```

Excerpt from the RDF definition of the "University of Porto" entity in DBpedia in N-Triples
https://dbpedia.org/page/University_of_Porto

Semantic Web

- The Semantic Web is a term coined be Tim Berners-Lee in 2001 referring to an envisioned extension to the original web.
- While the original web is a medium of documents for people — a Web of Documents.
- The semantic web is meant to be a medium for machines (intelligent agents) that have access to data and rules for reasoning about the data — a Web of Data.
- Semantic Web technologies include: URI, XML, RDF, various serializations of RDF (Turtle, N-Triples, RDFa, etc), SPARQL, OWL, etc — commonly referred to as the Semantic Web Stack.
- This vision is yet (?) to come true, however it has contributed to the development and publication of structured data in an unprecedented scale.

Entity-Oriented Search

- ➔ Entity-Oriented Search is the search paradigm of organizing and accessing information centered around entities, and their attributed and relationships.

- ➔ From a user perspective, entities are natural units for organizing information. Allowing users to interact with specific entities offers a richer and more effective user experience than what is provided by conventional document-based retrieval systems.

- ➔ From a machine perspective, entities allow for a better understanding of search queries, of document content, and even of users. Entities enable search engines to be more effective.

Entity-Oriented Search

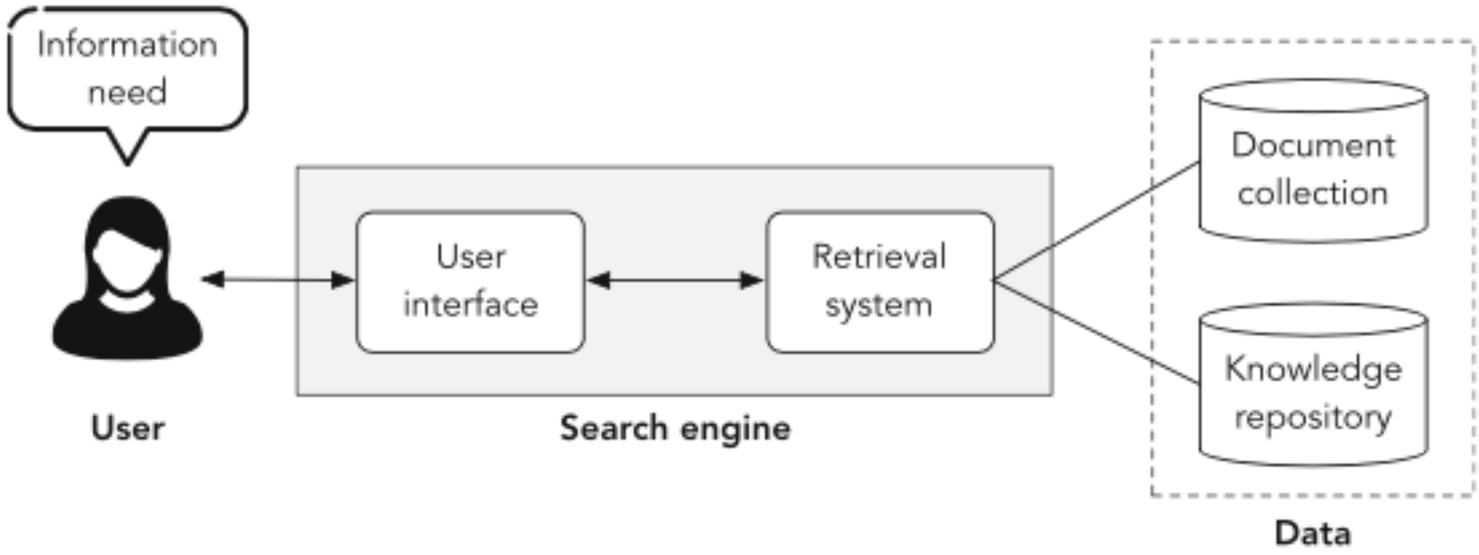


Fig. 1.3 Architecture of an entity-oriented search system

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Users and Information Needs

- ➔ Users articulate their information needs in many different ways.
- ➔ **Keyword queries**, also known as "free text queries", have become the most common approach with the advent of the web. Keyword queries are easy to formulate but imprecise.
- ➔ **Structured queries**, structured data sources are traditionally queried using formal query languages (e.g., SQL, SPARQL). These queries are very precise and usually intended for expert users with well-defined and precise information needs.
- ➔ **Keyword++ queries**, correspond to keyword queries that are complemented with additional structural elements, e.g. [retrieval site:fe.up.pt].
- ➔ **Natural language queries**, are formulated using natural language, the same way one would express it in a conversation, e.g. ["What is the capital of Nepal?"].
- ➔ **Zero-queries**, the traditional information access is reactive — the search system responds to a user-initiated query. Proactive systems, anticipate the user's information need, without requiring the user to issue a query. In practice, the context of the user is used to infer information needs.

Data

- Data can be grouped in three types.

- **Unstructured data**, which can be found in vast quantities in a variety of forms: web pages, spreadsheets, emails, tweets, etc. All these may be treated as textual documents, i.e. a sequence of words.

- **Semi-structured data**, is characterized by the lack of rigid, formal structure. Typically, it contains tags or other markup to separate textual content from semantic elements, e.g. HTML data.

- **Structured data**, adheres to a predefined (fixed) schema and is typically organized in a tabular format (e.g., relational databases). The schema defines how the data is organized and imposes constrains to ensure consistency.

Tasks in Entity-Oriented Search

- **Entity Retrieval — entities as the unit of retrieval**, between 40% to 70% of the queries in web search mention or target specific entities. Such queries are better answered by returning a ranked list of entities, as opposed to a list of documents. Challenges involved:
  - 1) how to represent information needs;
  - 2) how to represent entities;
  - 3) how to match those representations.
- **Entity Linking — entities for knowledge representation**, recognizing mentions of entities in text and associating these mentions with the corresponding entities in knowledge bases.
- **Entities for an enhanced user experience**, besides providing meaningful retrieval as information organization units, entities can improve the user experience throughout the entire search process.

Data Sources

Data Types and Search

➔ Recall the basic data types of data sources.

Fig. 2.1 The data spectrum

Table 2.1 Comparison of unstructured, semi-structured, and structured data search

	Unstructured	Semi-structured	Structured
Unit of retrieval	Documents	Objects	Tuples
Schema	No	Self-describing	Fixed
Queries	Keyword	Keyword++	Formal languages

Images from Entity-Oriented Search, Springer, K. Balog (2018)

Knowledge Bases

- A knowledge base is comprised of a large set of assertions about the world.
- When the emphasis is on the relationships between the entities, the term knowledge graph is commonly used.
- Resource Description Framework (RDF) is a language designed to describe "things", which are referred to as resources.
- Each resource is assigned a Uniform Resource Identifier (URI), making it uniquely and globally identifiable.
- Each RDF statement is a triple, consisting of subject, predicate, and object components
  - Subject, always a URI, denoting a resource;
  - Predicate, always a URI, corresponding to a relationship or property of the subject resource;
  - Object, either a URI (referring to another resource) or a literal.

Knowledge Bases

Michael Schumacher (born 3 January 1969) is a retired German racing driver, who raced in Formula One for Ferrari.

<dbr:Michael_Schumacher>	<foaf:name>	"Schumacher, Michael"
<dbr:Michael_Schumacher>	<dbo:birthPlace>	<dbr:West_Germany>
<dbr:Michael_Schumacher>	<dbo:birthDate>	"1969-01-03"
<dbr:Michael_Schumacher>	<rdf:type>	<dbo:RacingDriver>
<dbr:Michael_Schumacher>	<dct:subject>	<dbc:Ferrari_Formula_One_drivers>

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Knowledge Bases

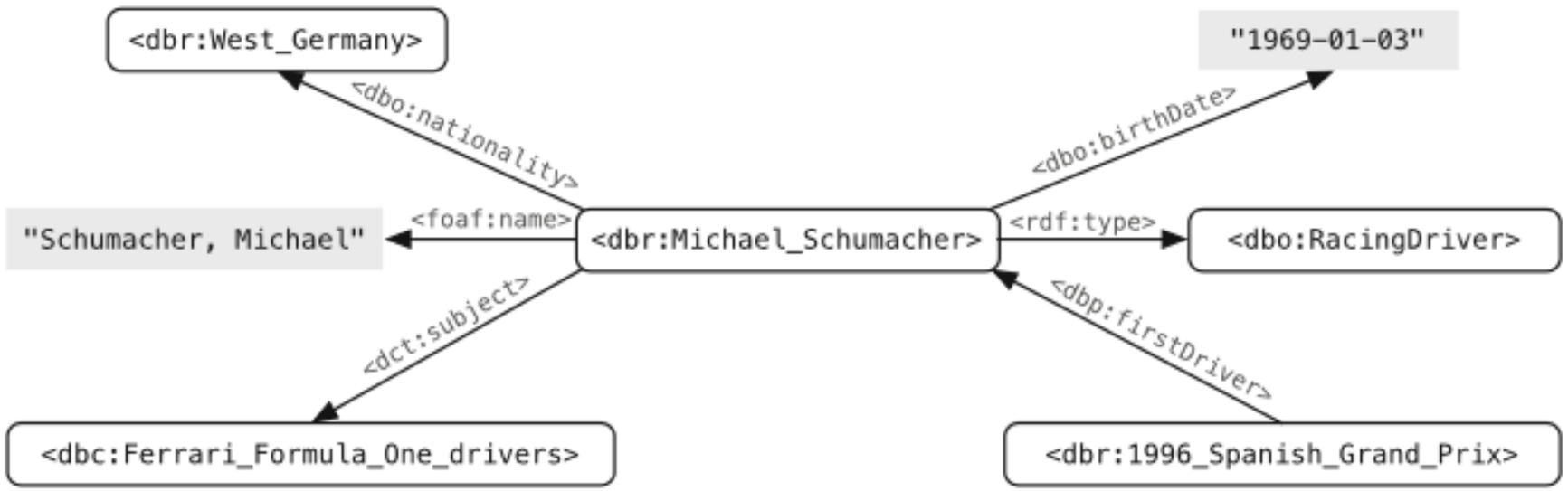


Fig. 2.3 Excerpt of an RDF graph (taken from DBpedia). URIs (i.e., entities) are represented by rounded rectangles, literals (i.e., attribute values) are denoted by shaded rectangles

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Related Technologies

- RDF describes the instance level in the knowledge base.
- RDFS and OWL are vocabularies for ontological modeling
  - An ontology is a means to formalize knowledge. Building blocks of an ontology include classes, instances, relations, attributes, restrictions, and rules and axioms.
- Serializations for RDF data: Notation-3, Turtle, N-Triples, RDFa, and RDF/JSON.
- SPARQL is structured query language for retrieving and manipulating RDF data.
- Triplestores are special-purpose databases designed for storing and querying RDF data.

Public Knowledge Bases

- DBpedia
  - Extracted from Wikipedia (mostly from inboxes) using a set of manually constructed mapping rules.
  - Community effort, users collaboratively create and edit the mapping rules.
  - Available in multiple languages.
  - Contains over 5 million entities (English).

DBpedia

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: Porto

An Entity of Type: Municipality, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

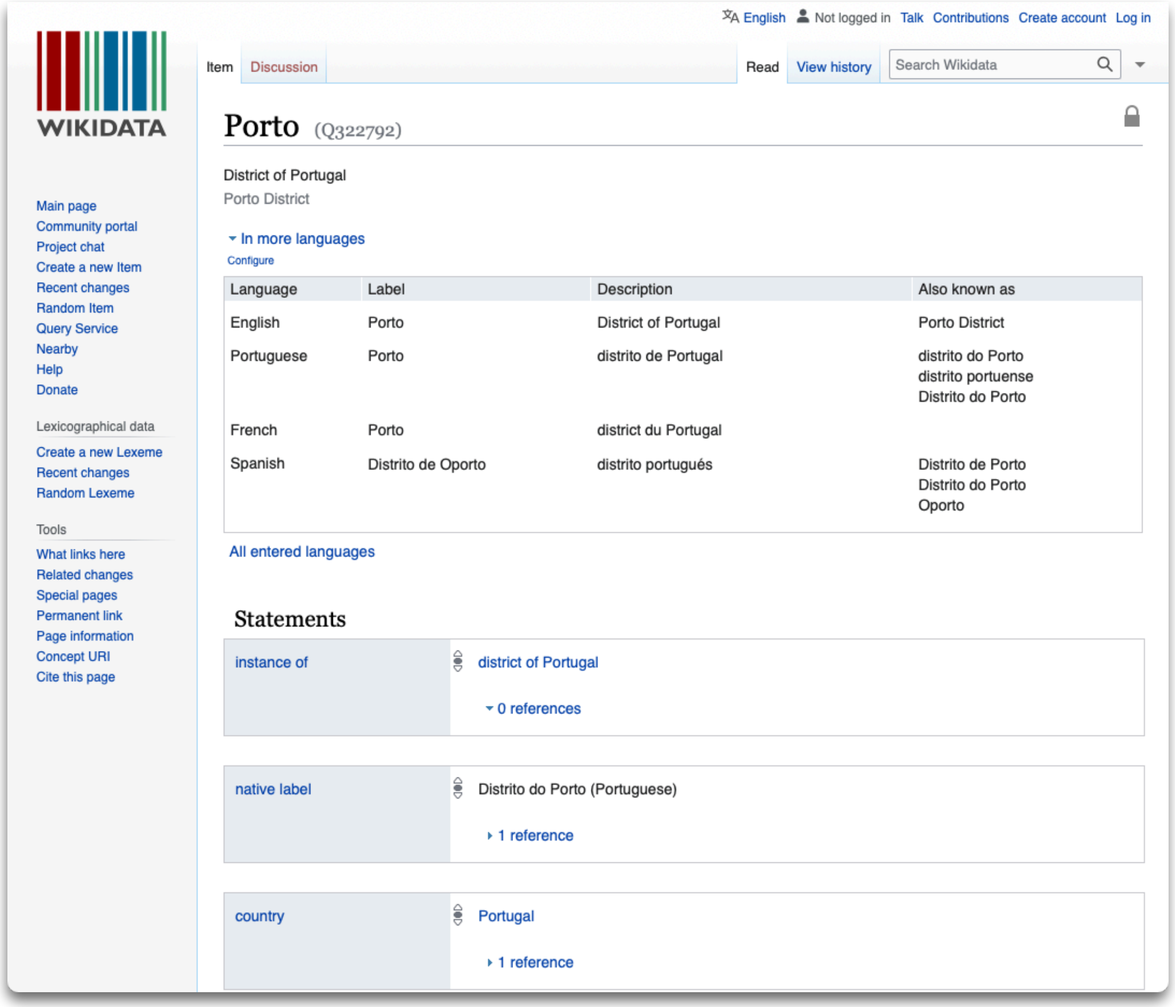
Porto or Oporto (Portuguese pronunciation: ['portu]) is the second-largest city in Portugal and one of the Iberian Peninsula's major urban areas. Porto city proper, which is the entire municipality of Porto, is small compared to its metropolitan area, with an estimated population of just 215,945 people in a municipality with only 41.42 km2. Porto's metropolitan area has around 1.7 million people (2019) in an area of 2,395 km2 (925 sq mi), making it the second-largest urban area in Portugal. It is recognized as a global city with a Gamma + rating from the Globalization and World Cities Research Network.

Property	Value
dbo:PopulatedPlace/areaTotal	• 41.42
dbo:abstract	• Porto or Oporto (Portuguese pronunciation: ['portu]) is the second-largest city in Portugal and one of the Iberian Peninsula's major urban areas. Porto city proper, which is the entire municipality of Porto, is small compared to its metropolitan area, with an estimated population of just 215,945 people in a municipality with only 41.42 km2. Porto's metropolitan area has around 1.7 million people (2019) in an area of 2,395 km2 (925 sq mi), making it the second-largest urban area in Portugal. It is recognized as a global city with a Gamma + rating from the Globalization and World Cities Research Network. Located along the Douro River estuary in northern Portugal, Porto is one of the oldest European centres, and its core was proclaimed a World Heritage Site by UNESCO in 1996, as "Historic Centre of Porto, Luiz I Bridge and Monastery of Serra do Pilar". The historic area is also a National Monument of Portugal. The western part of its urban area extends to the coastline of the Atlantic Ocean. Its settlement dates back many centuries, when it was an outpost of the Roman Empire. Its combined Celtic-Latin name, Portus Cale, has been referred to as the origin of the name Portugal, based on transliteration and oral evolution from Latin. In Portuguese, the name of the city includes a definite article: o Porto ("the port" or "the harbor"), which is where its English name "Oporto" comes from. Port wine, one of Portugal's most famous exports, is named after Porto, since the metropolitan area, and in particular the cellars of Vila Nova de Gaia, were responsible for the packaging, transport, and export of fortified wine. In 2014 and 2017, Porto was elected The Best European Destination by the Best European Destinations Agency. Porto is on the Portuguese Way path of the Camino de Santiago. (en)

Public Knowledge Bases

- Wikidata
  - Operated by the Wikimedia Foundation.
  - Its goal is to provide the same information as Wikipedia, but in a structured format.
  - Wikidata considers "claims" not "facts"
    - Each claim must be supported by a reference;
    - Claims can contradict each other and coexist, thereby allowing opposing views to be expressed (e.g., different political positions).

Wikidata


English
Not logged in
Talk
Contributions
Create account
Log in
WIKIDATA
Item
Discussion
Read
View history
Search Wikidata
Porto (Q322792)
District of Portugal
Porto District
In more languages
Configure
Language | Label | Description | Also known as
English | Porto | District of Portugal | Porto District
Portuguese | Porto | distrito de Portugal | distrito do Porto, distrito portuense, Distrito do Porto
French | Porto | district du Portugal
Spanish | Distrito de Oporto | distrito portugués | Distrito de Porto, Distrito do Porto, Oporto
All entered languages
Statements
instance of
district of Portugal
0 references
native label
Distrito do Porto (Portuguese)
1 reference
country
Portugal
1 reference
Main page
Community portal
Project chat
Create a new Item
Recent changes
Random Item
Query Service
Nearby
Help
Donate
Lexicographical data
Create a new Lexeme
Recent changes
Random Lexeme
Tools
What links here
Related changes
Special pages
Permanent link
Page information
Concept URI
Cite this page

Proprietary Knowledge Bases

- ➔ Google Knowledge Graph
- ➔ Facebook Entity Graph
- ➔ Microsoft Satori

Entity Ranking

Ad Hoc Entity Retrieval Task

- ➔ Ad hoc entity retrieval is the task of answering queries with a ranked list of entities.
- ➔ "Ad hoc" refers to the standard form of retrieval in which the user, motivated by an ad hoc (extemporary) information need, initiated the search process by formulating and issuing a query.

Table 3.1 Example entity search queries taken from various benchmarking campaigns [4]

Query
martin luther king
disney orlando
Apollo astronauts who walked on the Moon
Winners of the ACM Athena award
EU countries
Hybrid cars sold in Europe
birds cannot fly
Who developed Skype?
Which films starring Clint Eastwood did he direct himself?

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Main Strategy

- Model the "entity retrieval" problem as a "document retrieval" problem and take advantage of all the existing body of work.
- Create an entity description, or "profile document", for each entity in the catalog, containing all the existing "knowledge" about that entity, based on the available data.
- Once these entity descriptions are created, they can be indexed and ranked using existing document retrieval algorithms.
- Involves two steps, discussed in the next slides:
  - Constructing the profile documents;
  - Ranking the profile documents.

Constructing Entity Descriptions

- ➔ First step is to create a "profile document" with all the information we have about that entity.
- ➔ This will serve as the textual representation of the given entity, the "entity description".
- ➔ Entity descriptions can be assembled by considering the textual content, from a document collection, in which the entities occurs.
- ➔ In some cases, such descriptions may already be easily available — e.g., Wikipedia page of an entity. Also, there is a lot of information about entities organized and stored in knowledge bases.
- ➔ Three possible data sources: unstructured, semi-structured and structured.
- ➔ The goal is to estimate a term count associated with each entity.

Entity Components

- The estimative of the number of times a term is associated with an entity can be used to compute standard components, in an analogous way to the document model.
- **Entity length**, is the total number of terms in the entity description.
- **Term frequency (TF)**, is the normalized term count (by length) in the entity description.
- **Entity frequency (EF)**, is the number of entities in which the term occurs.
- **Inverse entity frequency (IEF)**, is the log normalized ratio between the total number of entities in the catalog, and the entity frequency.

Entity Representations from Unstructured Data

- ➔ This is the scenario where we want to find entities in arbitrary document collections.
- ➔ An approach is to have documents annotated with the entities that are references in them, and use these documents as a proxy to connect terms and entities.
- ➔ These annotations may be provided by human editors or be automated (e.g., entity linking).
- ➔ In this scenario, the estimated entity term count is obtained by considering each document, and adding the number of co-occurrences between a term and an entity, weighted by the strength of the association between the entity and the document.

Entity Representations from Semi-Structured Data

- A significant portion of the web content is already organized around entities, e.g., Wikipedia page about a person, IMDb page about a movie or an actor.
- The standard way of incorporating the internal document structure into the retrieval model is through the use of document fields, where document segments correspond to fields — e.g., title, summary, introduction, etc.
- The standard statistics are now computed at field-level, e.g. the number of times a term appears in a field of the description an entity.
- A special "catch-all" field is introduced to
  - (1) quickly filter entities in a first-pass retrieval;
  - (2) since entity description fields are often sparse, improve ranking by combining field-level scores with entity-level (the catch-all field) scores.

Entity Representations from Semi-Structured Data

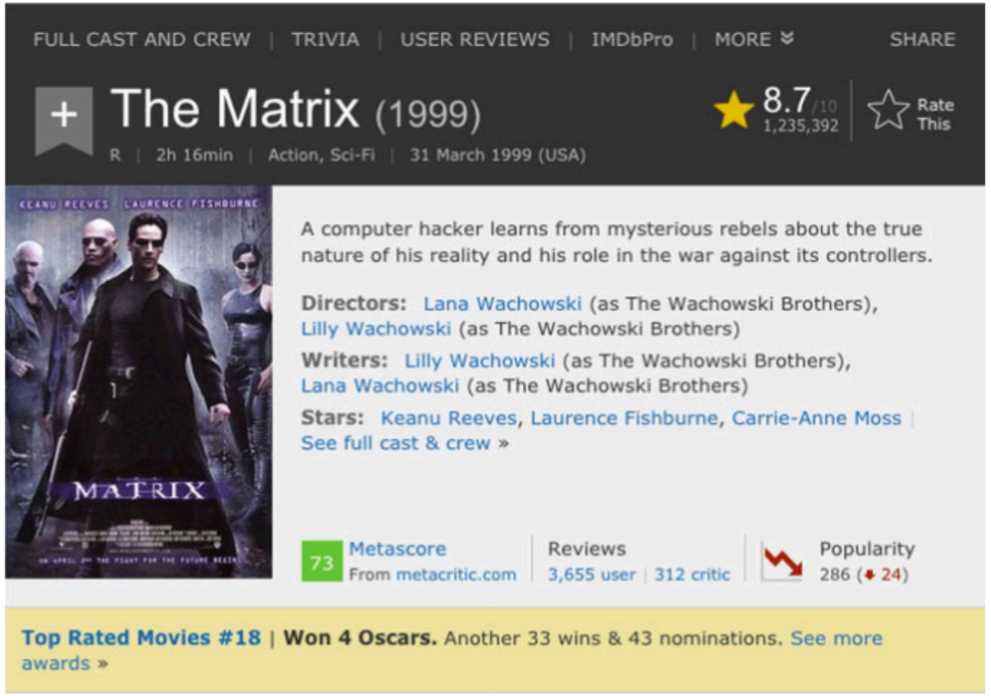


Fig. 3.4 Web page of the movie THE MATRIX from IMDb (http://www.imdb.com/title/tt0133093/)

Table 3.3 Fielded entity description created for THE MATRIX, corresponding to Fig. 3.4

Field	Content
Name	The Matrix
Genre	Action, Sci-Fi
Synopsis	A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers
Directors	Lana Wachowski (as The Wachowski Brothers), Lilly Wachowski (as The Wachowski Brothers)
Writers	Lilly Wachowski (as The Wachowski Brothers), Lana Wachowski (as The Wachowski Brothers)
Stars	Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss
Catch-all	The Matrix Action, Sci-Fi A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers. Lana Wachowski (as The Wachowski Brothers), Lilly Wachowski (as The Wachowski Brothers) Lilly Wachowski (as The Wachowski Brothers), Lana Wachowski (as The Wachowski Brothers) Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss

Images from Entity-Oriented Search, Springer, K. Balog (2018)

Entity Representations from Structured Data

- There is a growing number of large general-purpose publicly available knowledge bases.
- In knowledge bases information is organized using the RDF data model, specifically
  - each entity is uniquely identified by its URI (Uniform Resource Identifier);
  - its properties are described in the form of subject-predicate-object (SPO) triples.
- To represent an entity from a knowledge base, we consider the immediate vicinity of the entity node — i.e. the SPO tripes where the entity appears either as subject or object.
  - `<http://dbpedia.org/resource/Porto>  <http://dbpedia.org/property/name>  "Porto"@en .`
  - `<http://dbpedia.org/resource/Porto>  <http://xmlns.com/foaf/0.1/nick>  "(\"The Undefeated City\"),"@en`
  - `<http://dbpedia.org/resource/Porto>  <http://www.w3.org/2002/07/owl#sameAs>  <http://www.wikidata.org/entity/Q36433>`
- From the perspective of constructing entity representations, this is conceptually no different from having data stored in relational databases, where the same information is available through fields and foreign-key relationships.

Entity Representations from Triples

- ➔ The number of potential fields is huge (in the 1000s), thus we need a different strategy.
- ➔ A common solution is predicate folding — grouping predicates together into a small set of predefined categories, so that we have a handful go fields.
- ➔ Predicates may be grouped based on their type or importance.
- ➔ Common entity fields: name, name variants, attributed, types, outgoing relations, incoming relations, top predicates, catch-all.

RDF Graph

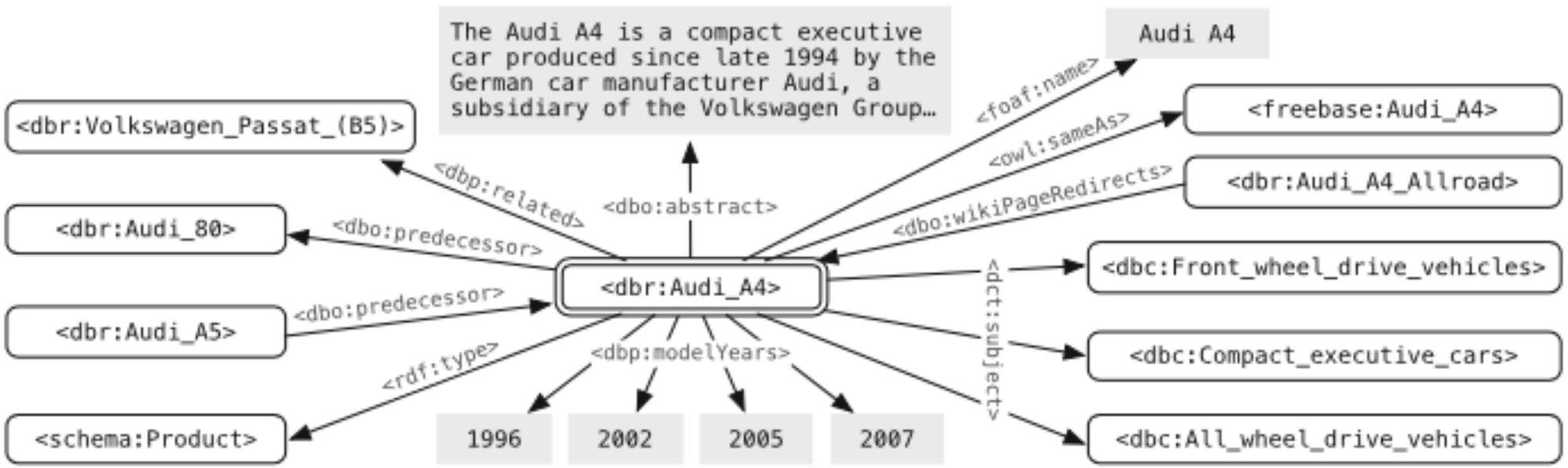


Fig. 3.5 Excerpt from an RDF graph. Rounded rectangles represent entities (URIs) and shaded rectangles denote attributes (literal values)

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Fielded Entity Description

Table 3.4 Excerpt from the fielded entity description of AUDI A4, corresponding to Fig. 3.5

Field	Content
Name	Audi A4
Name variants	Audi A4 … Audi A4 Allroad
Attributes	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [...] … 1996 … 2002 … 2005 … 2007
Types	Product … Front wheel drive vehicles … Compact executive cars … All wheel drive vehicles
Outgoing relations	Volkswagen Passat (B5) … Audi 80
Incoming relations	Audi A5
`<foaf:name>`	Audi A4
`<dbo:abstract>`	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [...]
Catch-all	Audi A4 … Audi A4 … Audi A4 Allroad … The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [...] … 1996 … 2002 … 2005 … 2007 … Product … Front wheel drive vehicles … Compact executive cars … All wheel drive vehicles … Volkswagen Passat (B5) … Audi 80 … Audi A5

Table from Entity-Oriented Search, Springer, K. Balog (2018)

Commonly Used Fields (1)

- **Name** contains the name(s) of the entity.
  - The two main predicates mapped to this field are <foaf:name> and <rdfs:label>.
  - One might follow a simple heuristic and additionally consider all predicates ending with "name", "label", or "title".
- **Name variants** (aliases) may be aggregated in a separate field.
  - In DBpedia, such variants may be collected via Wikipedia redirects (via <dbo:wikiPageRedirects>) and disambiguations (using <dbo:wikiPageDisambiguates>).
- **Attributes** includes all objects with literal values, except the ones already included in the name field.
  - In some cases, the name of the predicate may also be included along with the value, e.g., "founding date 1964" (vs. just the value part, "1964").

Commonly Used Fields (2)

- **Types** holds all types (categories, classes, etc.) to which the entity is assigned.
  - Commonly, <rdf:type> is used for types;
  - In DBpedia, <dct:subject> is used for assigning Wikipedia categories, which may also be considered as entity types.
- **Outgoing relations** contains all URI objects, i.e., names of entities (or resources in general) that the subject entity links to.
  - If the types or name variants fields are used, then those predicates are excluded;
  - Values might be prefixed with the predicate name, e.g., "spouse Michelle Obama".
- **Incoming relations** is made up of subject URIs from all SPO triples where the entity appears as object.
- **Top predicates** may be considered as individual fields
  - e.g., include the top-1000 most frequent DBpedia predicates as fields.
- **Catch-all** (or content) is a field that amasses all textual content related to the entity.

Triples to Text

- How to represent triples as text?
- Triple objects values are either URIs (links) or literals.
- Literals can be treated as regular text.
- URIs need to be converted to human-readable
  - Some are user-friendly: https://dbpedia.org/page/Porto
  - Others are not: http://www.wikidata.org/entity/Q36433
- URI resolution is the process of finding the a human-readable name/label for a URI.

URI Resolution

- The specific predicate that holds the name of a resource depends on the RDF vocabulary used.
- Commonly, <foaf:name> or <rdfs:label> are used
- Given a SPO triple, for example
  - `<dbr:Audi_A4>  <rdf:type>  <dbo:MeanOfTransportation>`
- The corresponding resources's name is contained in the object element of this triple:
  - `<dbo:MeanOfTransportation>  <rdfs:label>  “mean of transportation"`

Ranking Entities

- ➔ With the term-based entity representations, we now need to rank entities with respect to their relevance to a search query.
- ➔ This can be viewed as a the problem of assigning a score to each entity in the catalog.
- ➔ The retrieval models from document ranking — e.g. vector space model, language models, probabilistic models — are used for entity scoring by replacing the *document* with *entity* in the equations.

Entity Ranking Evaluation

- Evaluation of ad hoc entity retrieval is analogous to that of ad hoc document retrieval.
- Given a ranked list of items, the relevance of each item is judged with respect to the underlying information need.
  - Set-based measures, e.g. precision, recall, precision at rank cutoff k (P@10, P@20).
  - Rank-based measures, e.g. average precision (AvP), mean average precision (MAP).
- The use of text collections is the de facto standard standard for evaluation in IR. Some relevant collections that can be used for entity retrieval are: INEX Entity Ranking (XER), TREC Entity, Semantic Search Challenge, INEX Linked Data.

Entity Ranking Summary

- ➔ In this section we discussed approaches for ranking entities in various datasets, from unstructured documents, to structured knowledge bases.
- ➔ Most of the effort in this area has been in constructing term-based representations of entities, which can be ranked using traditional document retrieval techniques.
- ➔ Unstructured entity representations with a bag-of-words retrieval models usually provide solid performance and a good starting point.
- ➔ Other approaches exist, most notably the use of specific characteristics of entities, such as relationships.

Entity Linking

Entity Linking

➔ Entity linking is the task of recognizing entity mentions in text and linking them to the corresponding entries in a knowledge base.

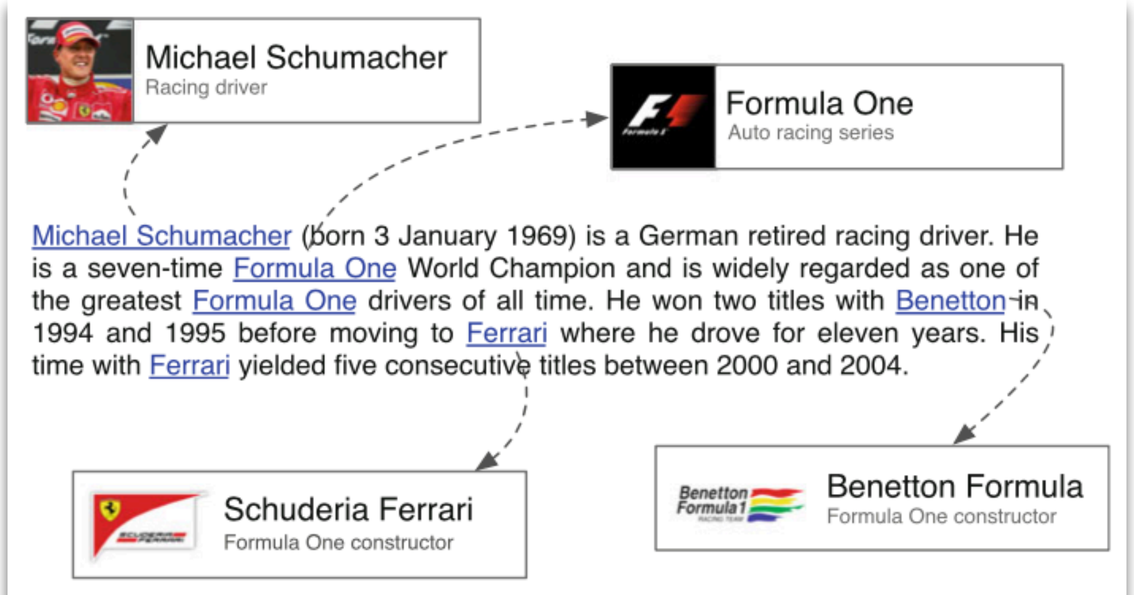
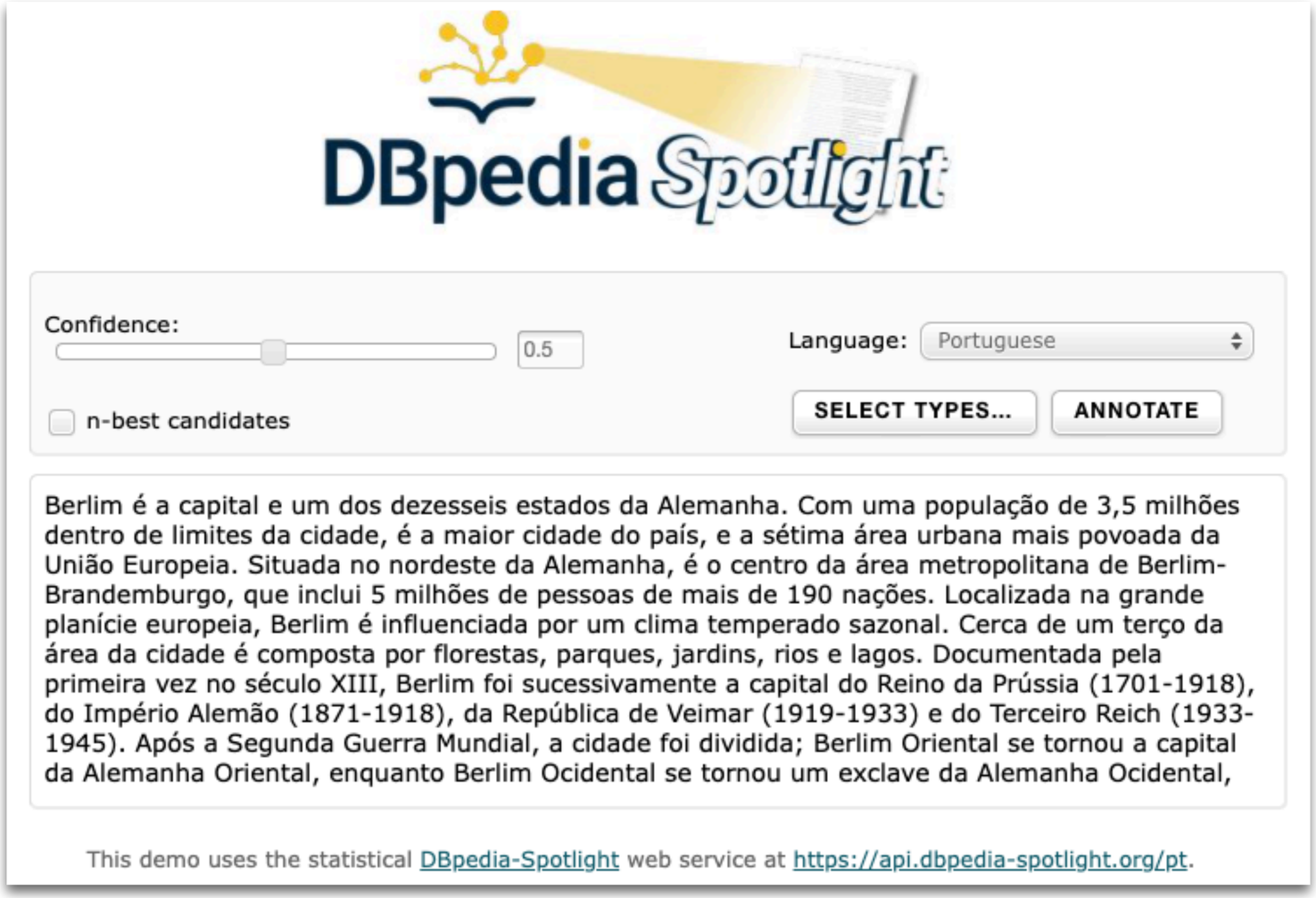


Fig. 5.1 Example of text annotated with entities from Wikipedia. Blue underlined text indicates the linked entity mentions. Text is taken from https://en.wikipedia.org/wiki/Michael_Schumacher

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Practical Example (Text)



Available online at: https://demo.dbpedia-spotlight.org/

Practical Example (Annotated)

Available online at: https://demo.dbpedia-spotlight.org/

Anatomy of an Entity Linking System

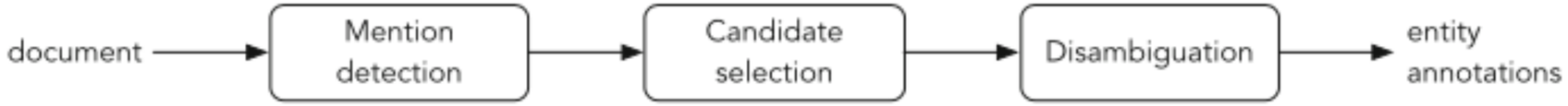


Fig. 5.2 Entity linking pipeline

- ➔ **Mention detection**, identification of text snippets that can potentially be linked to entities.
- ➔ **Candidate selection**, generating a set of candidate entities for each mention.
- ➔ **Disambiguation**, selecting a single entity (or none) for each mention, based on the context.

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Common Approach for Mention Detection

- ➔ Build a dictionary o entity surface forms, i.e. entities with all names variants.
- ➔ Check all document n-grams against the dictionary
- ➔ Filter out undesired entities

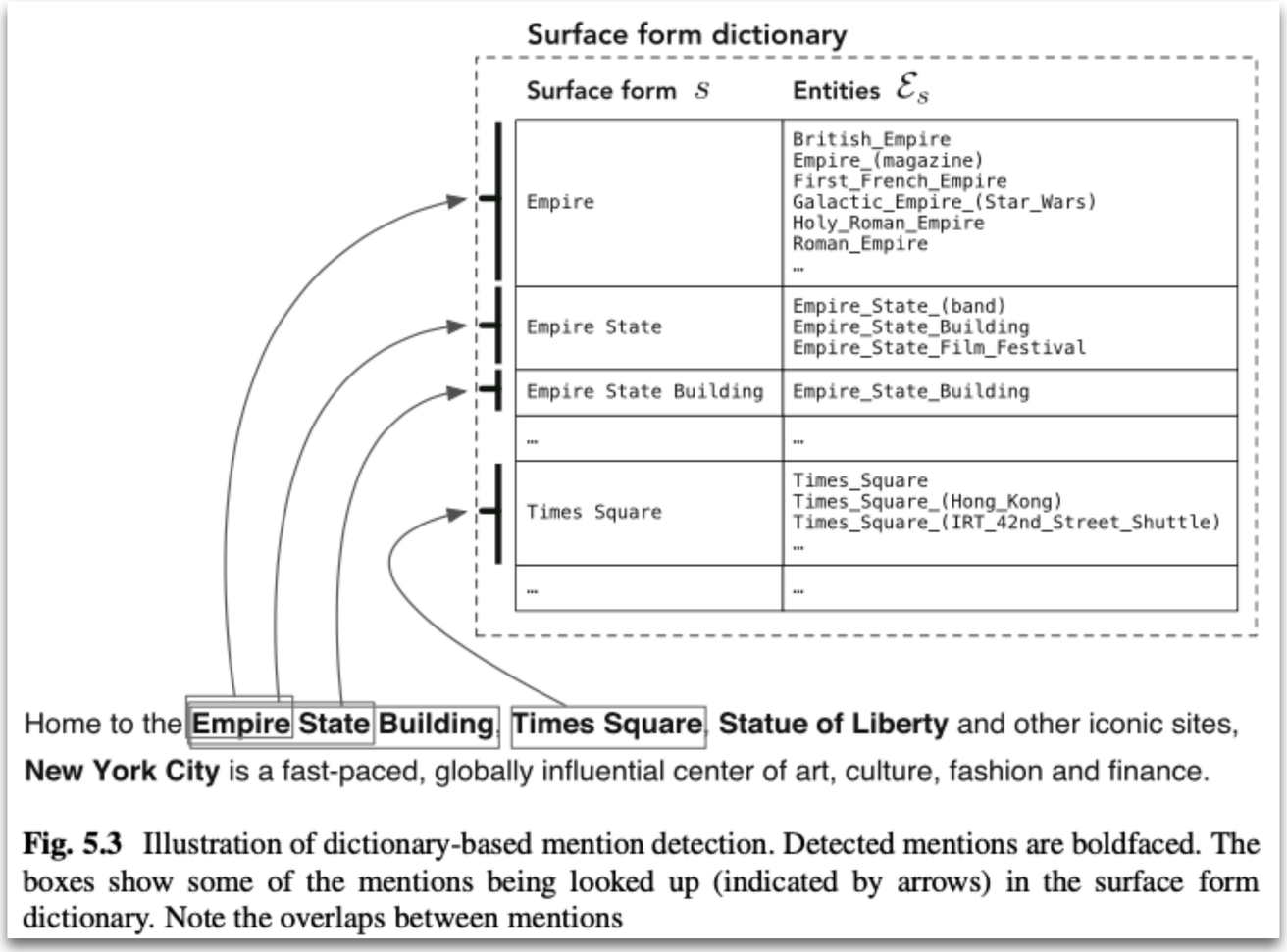


Fig. 5.3 Illustration of dictionary-based mention detection. Detected mentions are boldfaced. The boxes show some of the mentions being looked up (indicated by arrows) in the surface form dictionary. Note the overlaps between mentions

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Entity Linking Systems

Table 5.5 Overview of publicly available entity linking systems

System	Reference KR	Online demo	Web API	Source code
AIDA[a]	YAGO2	Yes	Yes	Yes (Java)
DBpedia Spotlight[b]	DBpedia	Yes	Yes	Yes (Java)
Illinois Wikifier[c]	Wikipedia	No	No	Yes (Java)
TAGME[d]	Wikipedia	Yes	Yes	Yes (Java)
Wikipedia Miner[e]	Wikipedia	No	No	Yes (Java)

- ➔ http://www.mpi-inf.mpg.de/yago-naga/aida/
- ➔ http://spotlight.dbpedia.org/
- ➔ http://cogcomp.cs.illinois.edu/page/download_view/Wikifier
- ➔ https://tagme.d4science.org/tagme/
- ➔ https://github.com/dnmilne/wikipediaminer

Table from Entity-Oriented Search, Springer, K. Balog (2018)

References and Further Reading

- K. Balog. Entity-Oriented Search. Springer, 2018. https://eos-book.org/
- H. Bast, B. Buchhold, and E. Haussmann. Semantic search on text and knowledge bases. Foundations and Trends in Information Retrieval, 10(2–3):119–271, 2016.

Search User Interfaces

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Search User Interface Design, Max L. Wilson (2011)

Outline

- ➔ Introduction
- ➔ Early Search UIs
- ➔ Modern Search UIs
- ➔ Evaluation of Search UIs

Introduction and Context

Introduction

- ➔ Search systems are built to be used by humans, thus a search user interface (SUI) is necessary for users to interact with the system.
- ➔ Information retrieval started with a system-centered view, and evolved to a both system- and user-centered view with the observation that more specific queries improved the relevance of results.
- ➔ Multiple factors concur to the overall user experience, the implemented algorithms, the existing metadata, the design of the user interfaces, its aesthetics and visual design.
- ➔ Search user interface development is a multidisciplinary area, requiring multiple and diverse expertises working together.

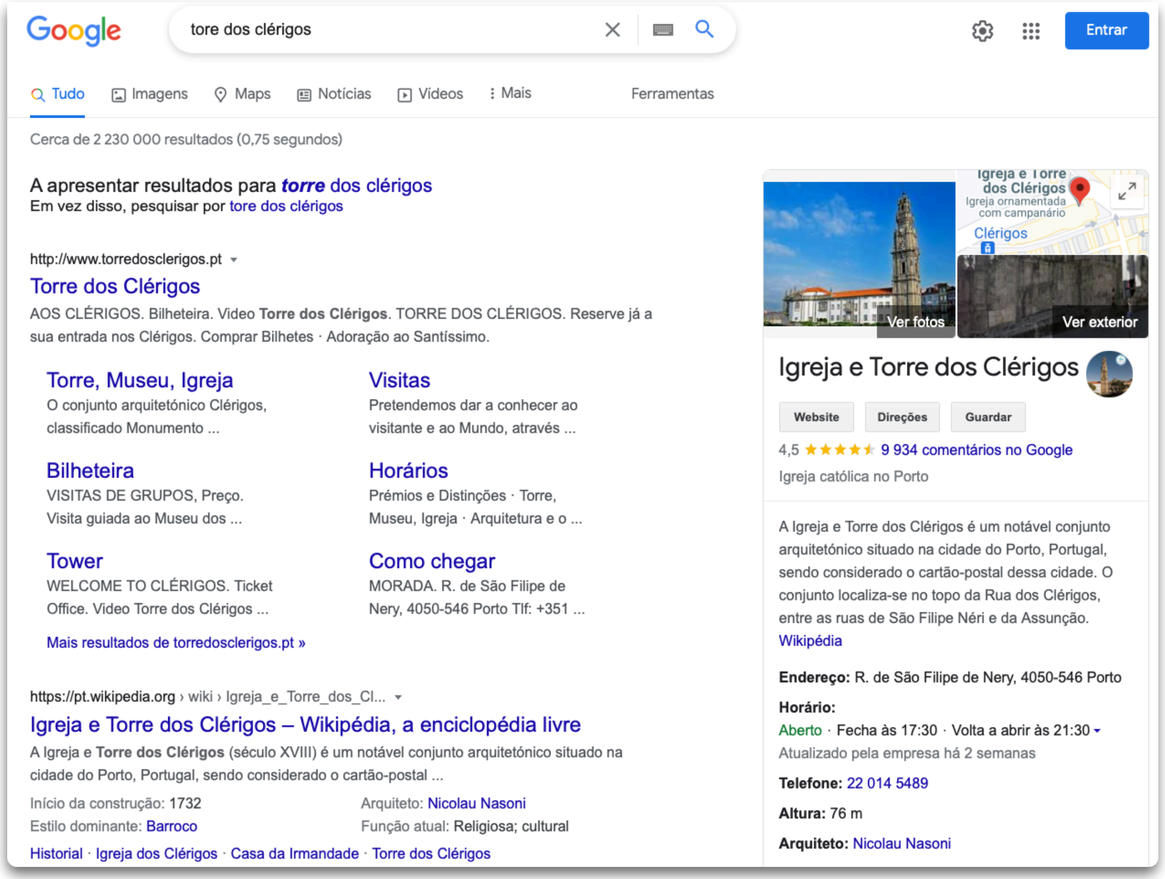
Search in early 2011

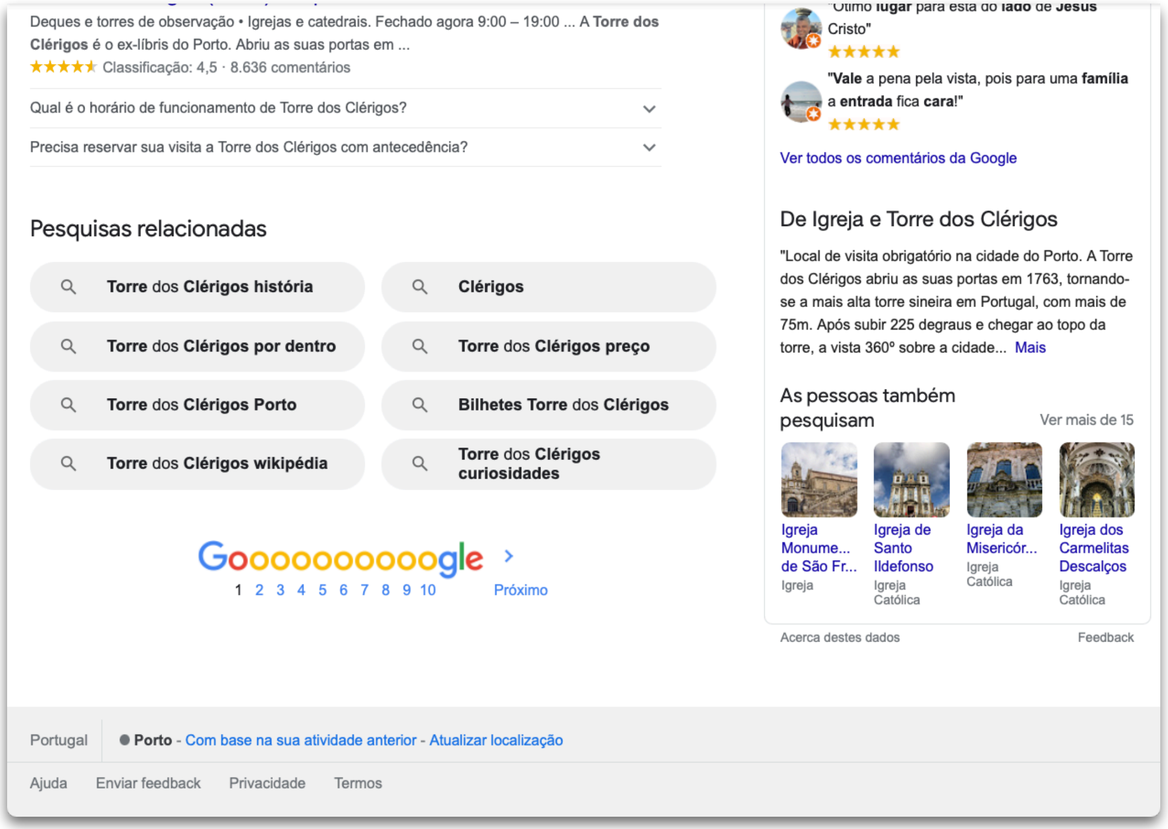
1. Search box
2. Results
3. Pagination
4. Specialized search (imagens, maps …)
5. Contextual filtering options
6. Scope restriction based on range time
7. Scope restriction based on characteristics
8. Advanced search
9. Related queries
10. Information about the volume of results
11. Automatic correction / suggestion
12. Previous searchers
13. Personalization of search settings
14. User accounts

Figure 1.1: 14 notable features in the Google Search User Interface (SUI) in early 2011, described below.

Image from Search User Interface Design, Max L. Wilson (2011)

Search in early 2022





A Framework for Thinking About Search Features

- Max L. Wilson proposes a framework to help on the discussion of search user interfaces, elements and features.
- It divides the elements of a search user interface in 4 main groups:
  - **Input** - features that allow the searcher to express what they are looking for.
  - **Control** - features that help searchers to modify, refine, restrict, or expand their input.
  - **Informational** - features that provide results or information about results.
  - **Personalization** - features that relate specifically to searchers and their previous interactions.

Features in Google SUI (early 2011)

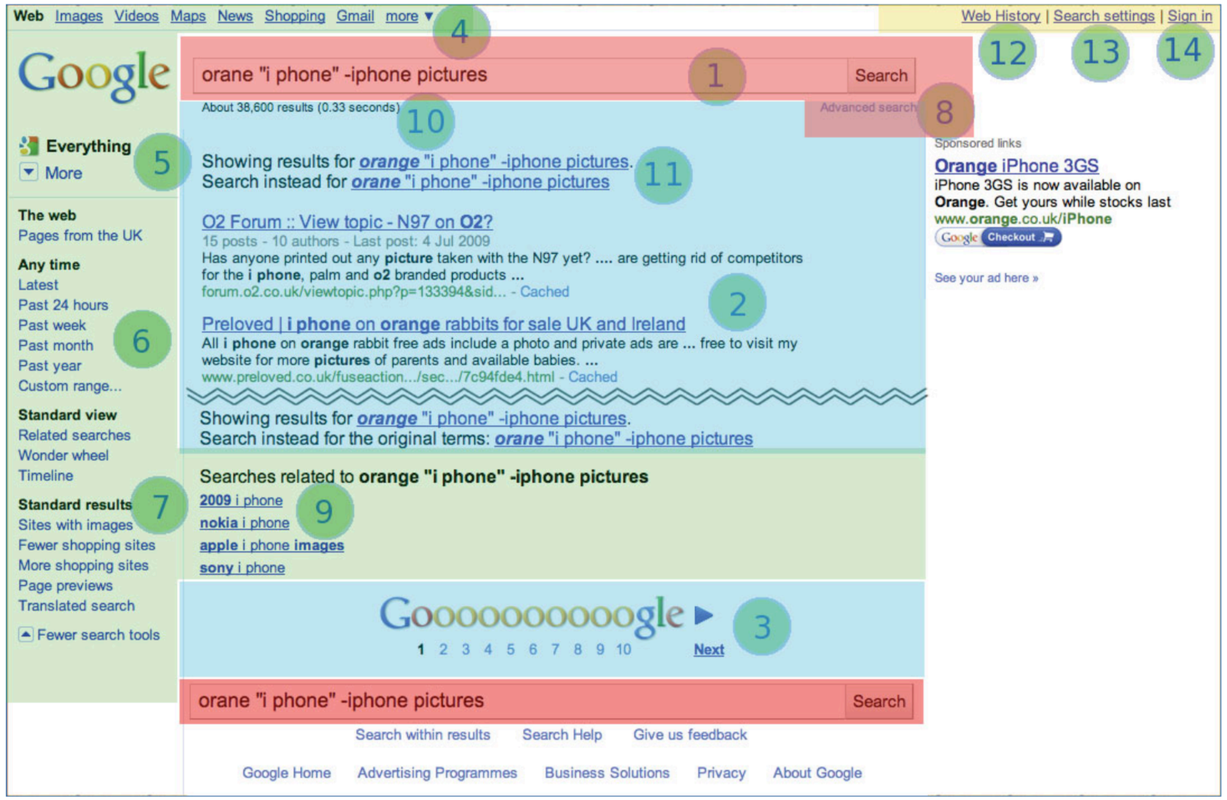
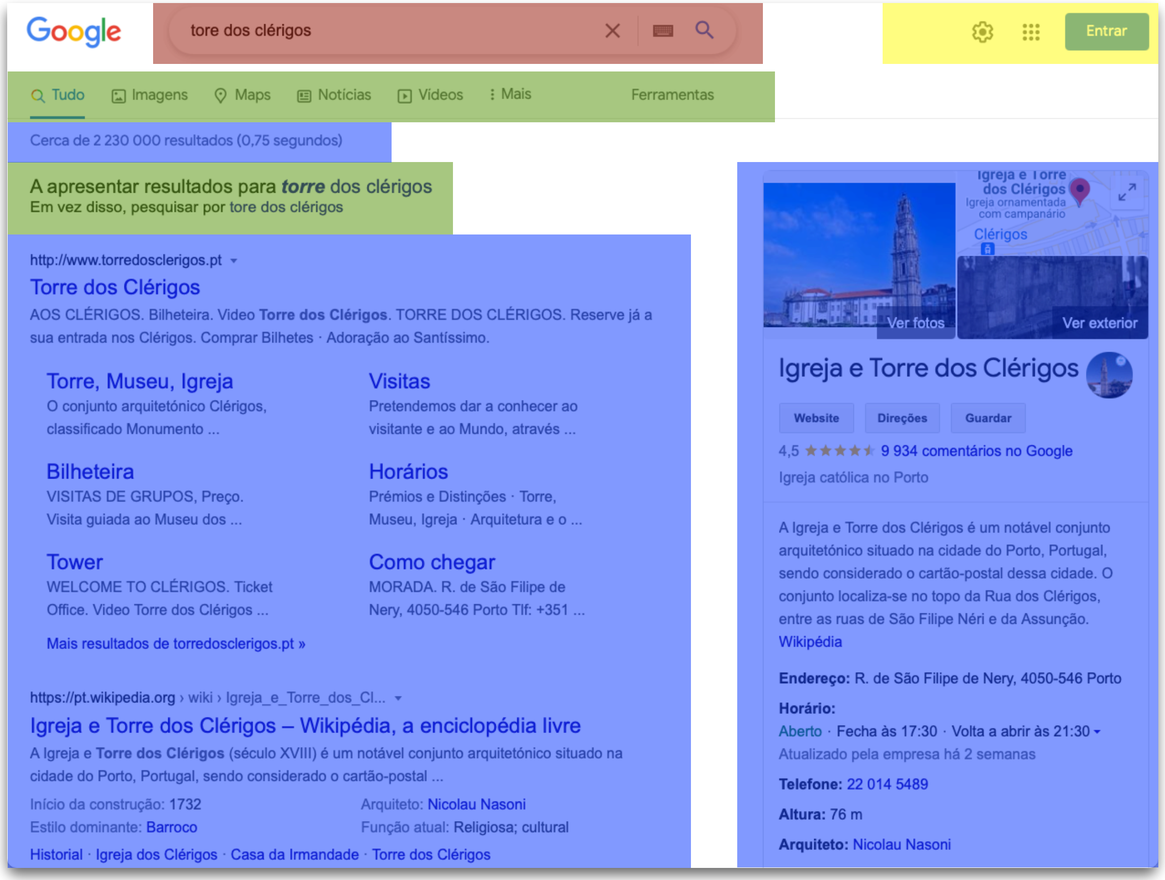


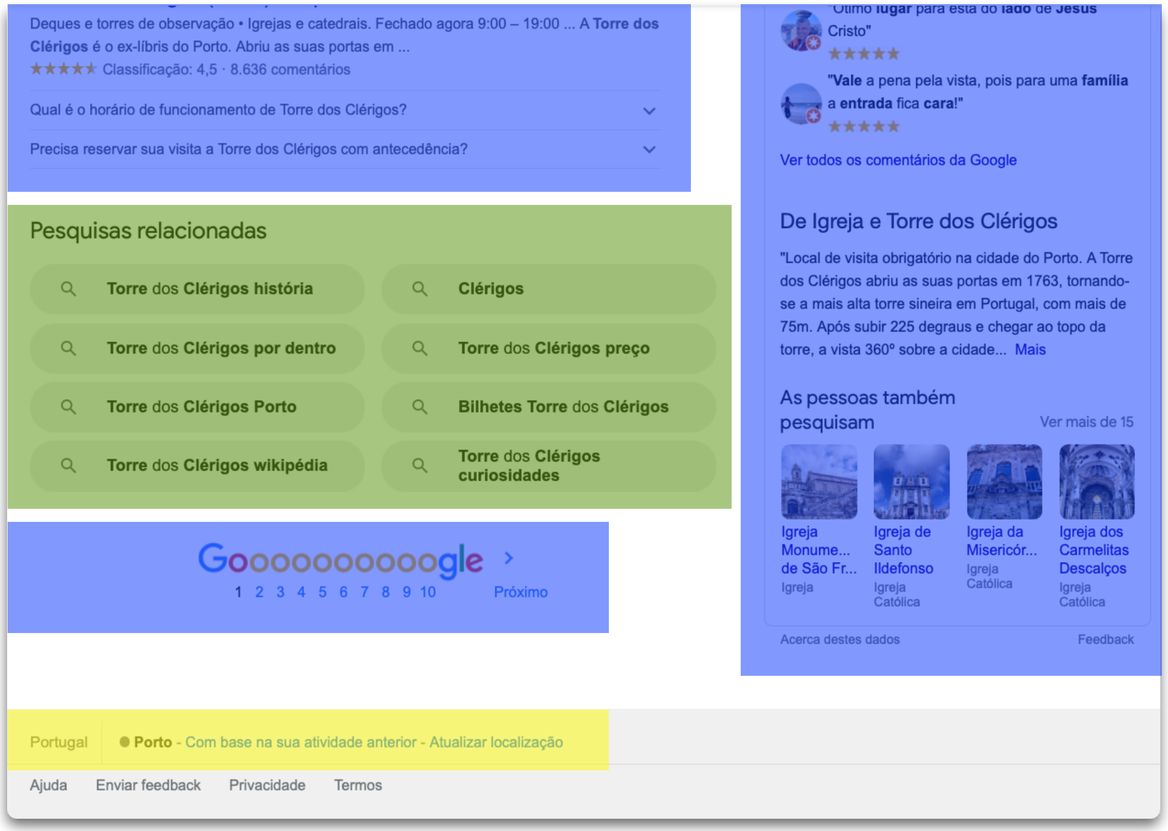
Figure 1.2: The Google SUI, from early 2011, zoned by the different types feature categories: Input as Red (including features #1 and #8), Control as Green (#4–#7, #9), Informational as Blue (#2, #3, #10, #11), and Personalisable as Yellow (#12–#14).

Image from Search User Interface Design, Max L. Wilson (2011)

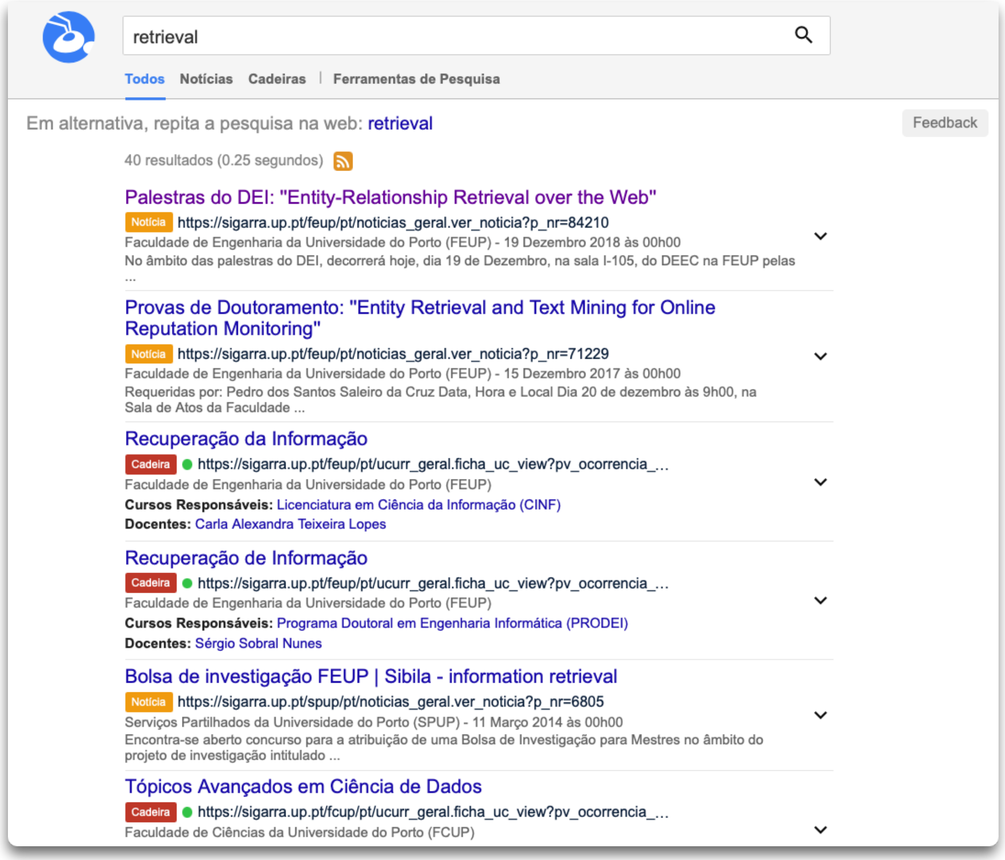
Features in Google SUI (early 2022)



Informational	Control	Input	Personalization



SIGARRA Search



Related Disciplines

- Six disciplines, or factors, contribute to the design of a search user interface.
  - User Experience
  - Graphic Design
  - Information Retrieval
  - Library & Information Science
  - Info-Seeking
  - Human-Computer Interaction

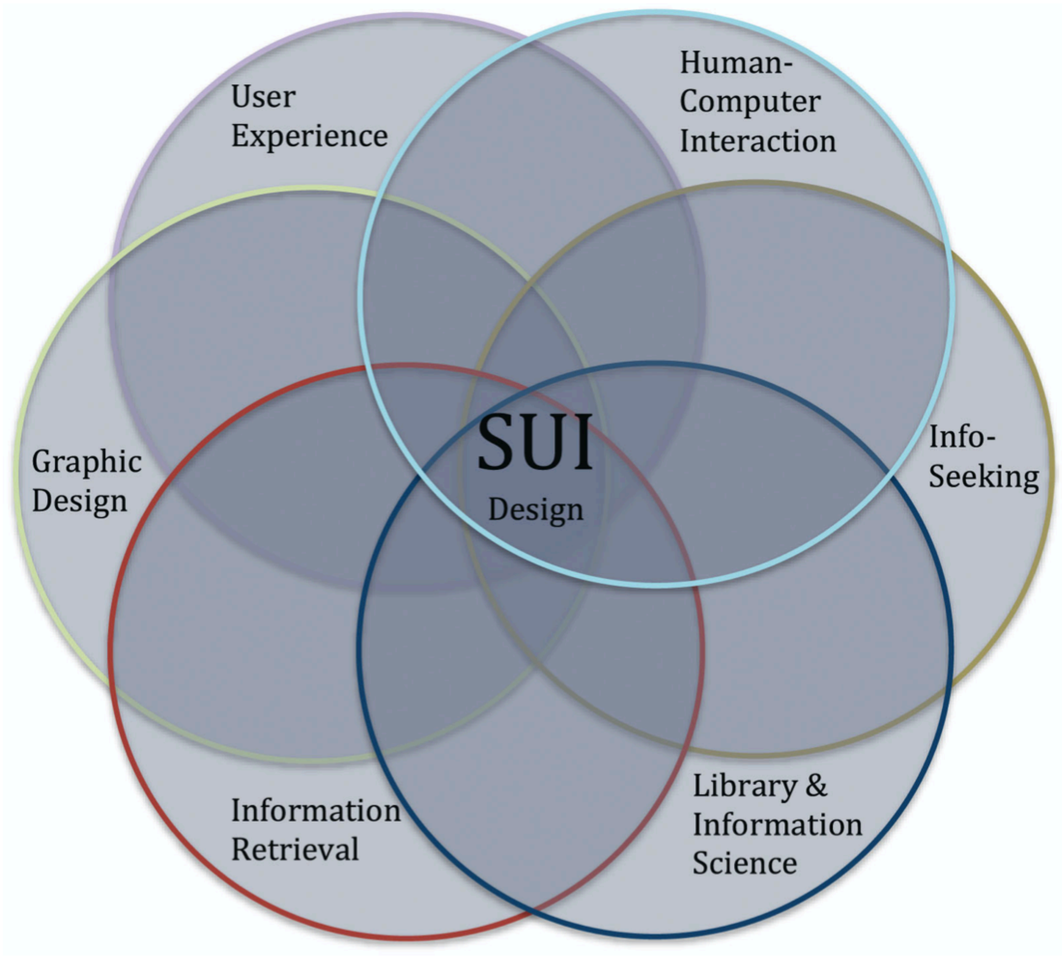


Figure 2.2: Search User Interface (SUI) design is affected by 6 related disciplines.

Image from Search User Interface Design, Max L. Wilson (2011)

Impact Factors

- ➔ Organization and properties of the data sources, e.g. available metadata.
- ➔ IR algorithms adopted, considering both speed and effectiveness.
- ➔ Graphic design, aesthetics impact how people judge the trustworthiness of a website. Simple use of color and other simple visual cues can have significant impacto on SUI success.
- ➔ How people search for information — field of Information Seeking Behavior — also impacts on search user interfaces design.

UI Design Principles (1)

➔ General users interface design guidelines from the HCI and UX communities.

1. **Visibility**, keep the user informed about what is happening at any given time. Example: keep the current search terms in the query box.

2. **Language**, adopt language that the user can understand. Example: instead of 'query' use the term 'search', or instead of 'query expansion' use 'related searches'.

3. **Control and freedom**, do not block users in a hole or fixed pathway, instead provide mechanisms for users to recover from them. Example: highlight spelling errors or mistakes but do not force them on users.

4. **Consistency**, adopt a consistent design that follows the same conventions. Example: always use 'search' and not 'query' or 'keyword search' in other places.

UI Design Principles (2)

5. **Error prevention**, make it hard to do unproductive things, i.e. avoid the need to undo actions.

6. **Support recognition**, help users not have to remember what they have done or need to do. Example: provide related searchers, keep the query in the search box.

7. **Flexibility and efficiency**, provide features and shortcuts for experiences users to be more productive and efficient. Example: let users navigate search results with keyboard shortcuts.

8. **Aesthetics and minimalism**, keep design simple and minimalist; make sensible use of white space.

UI Design Principles (3)

9. **Clear error messages**, provide informative and useful error messages. Example: not only state the lack of results, but provide alternative searches.

10. **Help and documentation**, provide help as documentation, FAQs, and examples.

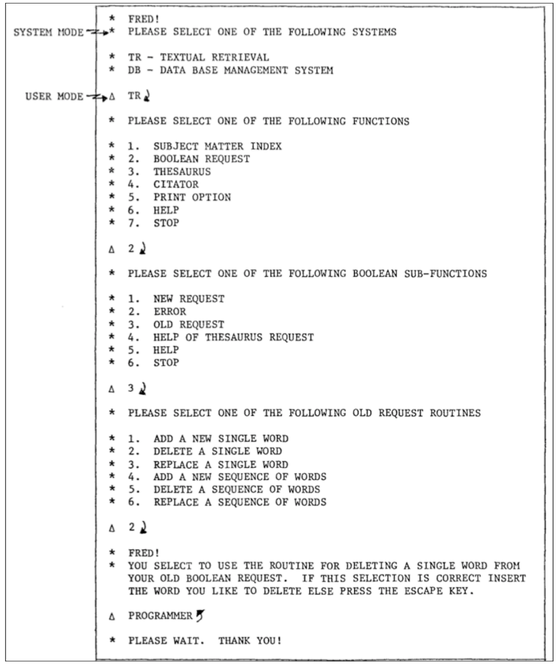
➔ Other UI design principles:

- ➔ Shneiderman's "Eight Golden Rules"
  https://www.cs.umd.edu/~ben/goldenrules.html
- ➔ Bruce Tognazzini's "First Principles of Interaction Design"
  https://asktog.com/atc/principles-of-interaction-design/
- ➔ Jakob Nilsen's "10 Usability Heuristics for User Interface Design"
  https://www.nngroup.com/articles/ten-usability-heuristics/

Early Search User Interfaces

Conversation and Dialogues

➔ Given the UI limitations, early (1970s) SUIs were modeled around conversations or dialogues.



```
                 *  FRED!
SYSTEM MODE ---> *  PLEASE SELECT ONE OF THE FOLLOWING SYSTEMS

                 *  TR - TEXTUAL RETRIEVAL
                 *  DB - DATA BASE MANAGEMENT SYSTEM

USER MODE ---->  Δ  TR ↵

                 *  PLEASE SELECT ONE OF THE FOLLOWING FUNCTIONS

                 *  1.  SUBJECT MATTER INDEX
                 *  2.  BOOLEAN REQUEST
                 *  3.  THESAURUS
                 *  4.  CITATOR
                 *  5.  PRINT OPTION
                 *  6.  HELP
                 *  7.  STOP

                 Δ  2 ↵

                 *  PLEASE SELECT ONE OF THE FOLLOWING BOOLEAN SUB-FUNCTIONS

                 *  1.  NEW REQUEST
                 *  2.  ERROR
                 *  3.  OLD REQUEST
                 *  4.  HELP OF THESAURUS REQUEST
                 *  5.  HELP
                 *  6.  STOP

                 Δ  3 ↵

                 *  PLEASE SELECT ONE OF THE FOLLOWING OLD REQUEST ROUTINES

                 *  1.  ADD A NEW SINGLE WORD
                 *  2.  DELETE A SINGLE WORD
                 *  3.  REPLACE A SINGLE WORD
                 *  4.  ADD A NEW SEQUENCE OF WORDS
                 *  5.  DELETE A SEQUENCE OF WORDS
                 *  6.  REPLACE A SEQUENCE OF WORDS

                 Δ  2 ↵

                 *  FRED!
                 *  YOU SELECT TO USE THE ROUTINE FOR DELETING A SINGLE WORD FROM
                    YOUR OLD BOOLEAN REQUEST.  IF THIS SELECTION IS CORRECT INSERT
                    THE WORD YOU LIKE TO DELETE ELSE PRESS THE ESCAPE KEY.

                 Δ  PROGRAMMER ↵

                 *  PLEASE WAIT.  THANK YOU!
```

Figure 3.1: A series of dialogue-style questions aimed to help the searcher describe what they are searching for; taken from [179].

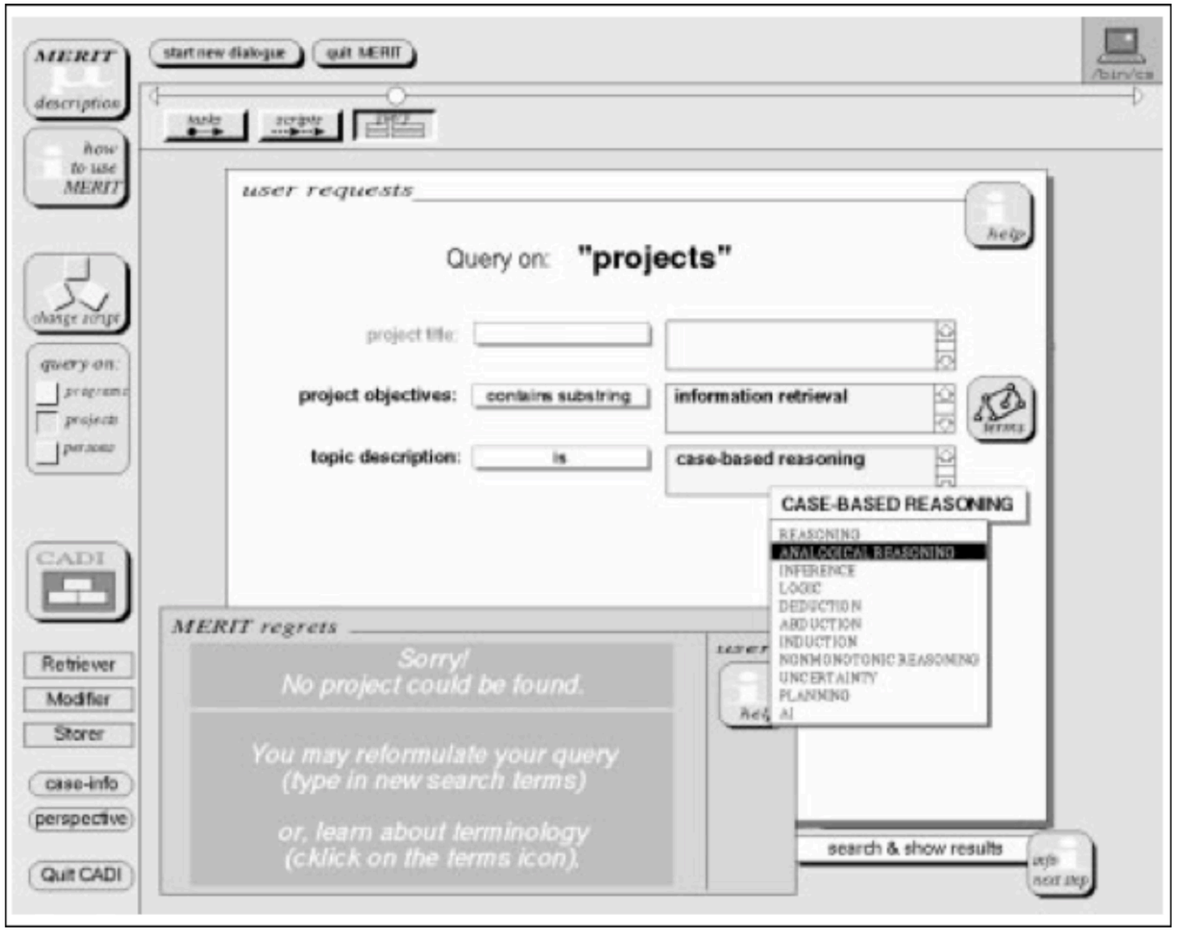
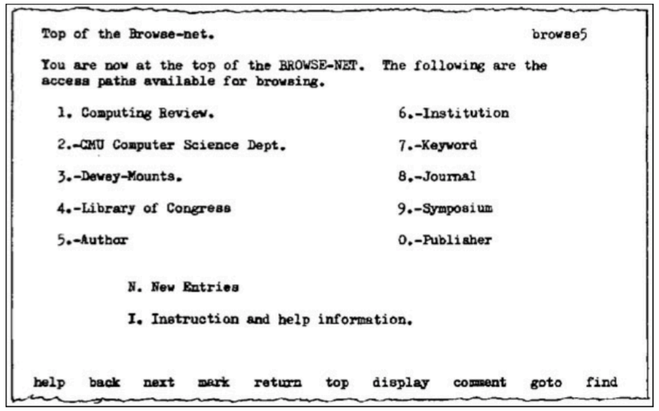


Figure 3.2: An advanced IIR SUI, called Merit, that responds based on a complex model of human conversations; taken from [16].

Images from Search User Interface Design, Max L. Wilson (2011)

Browsing

- ➔ Another type of system, given the command line technology available, was represented by browsing SUIs.
- ➔ Directories were a popular solution for SUIs in the 1990s, with the growth of the web.



```
Top of the Browse-net.                                    browse5

You are now at the top of the BROWSE-NET.  The following are the
access paths available for browsing.

  1. Computing Review.                    6.-Institution

  2.-CMU Computer Science Dept.           7.-Keyword

  3.-Dewey-Mounts.                        8.-Journal

  4.-Library of Congress                  9.-Symposium

  5.-Author                               0.-Publisher

          N. New Entries

          I. Instruction and help information.

help   back   next   mark   return   top   display   comment   goto   find
```

Figure 3.3: An early browsing interface for databases that provides options for different ways of accessing the documents; taken from [147].

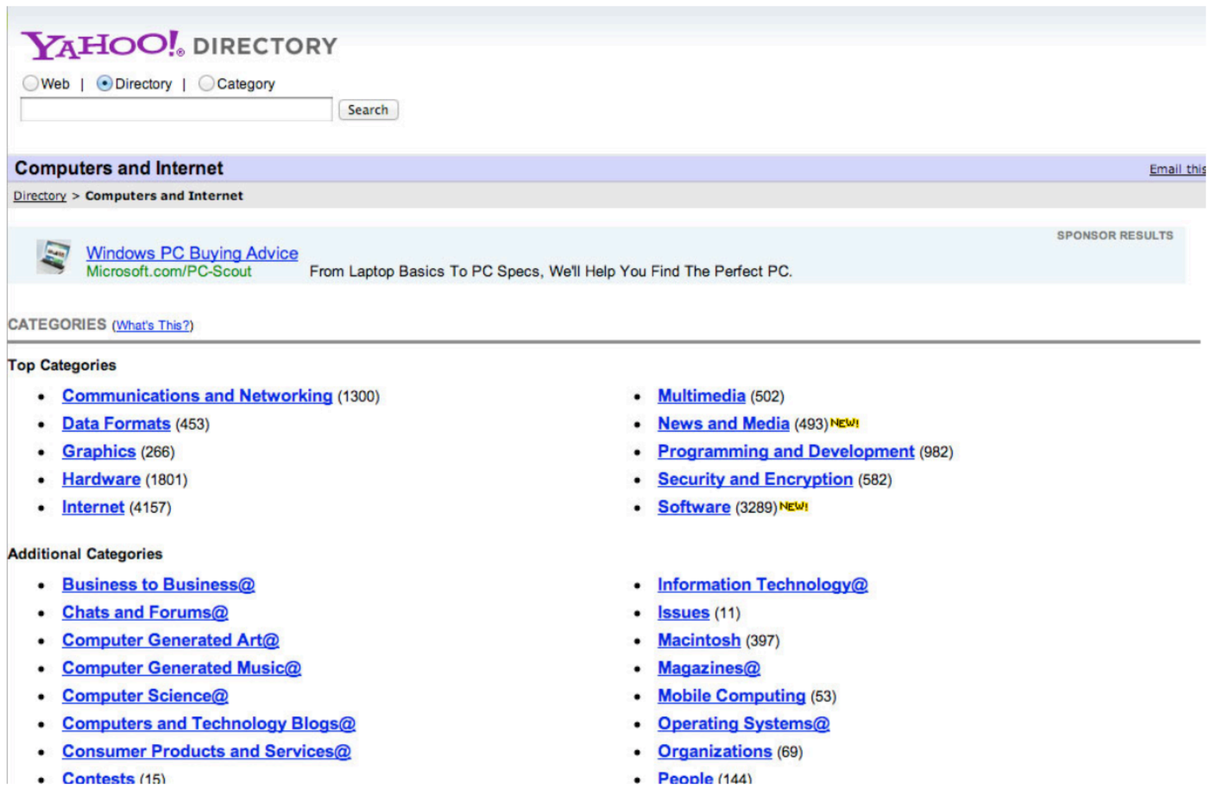


Figure 3.4: Yahoo! Directory, like other web directories, was extremely popular until the efficiency and scale of search engines like Google improved.

Images from Search User Interface Design, Max L. Wilson (2011)

Form Filling

- ➔ SUIs became more interactive with the emergence of Graphical User Interfaces (GUIs) in the 1980s.
- ➔ The "form filling" paradigm, that we still use today, became popular.

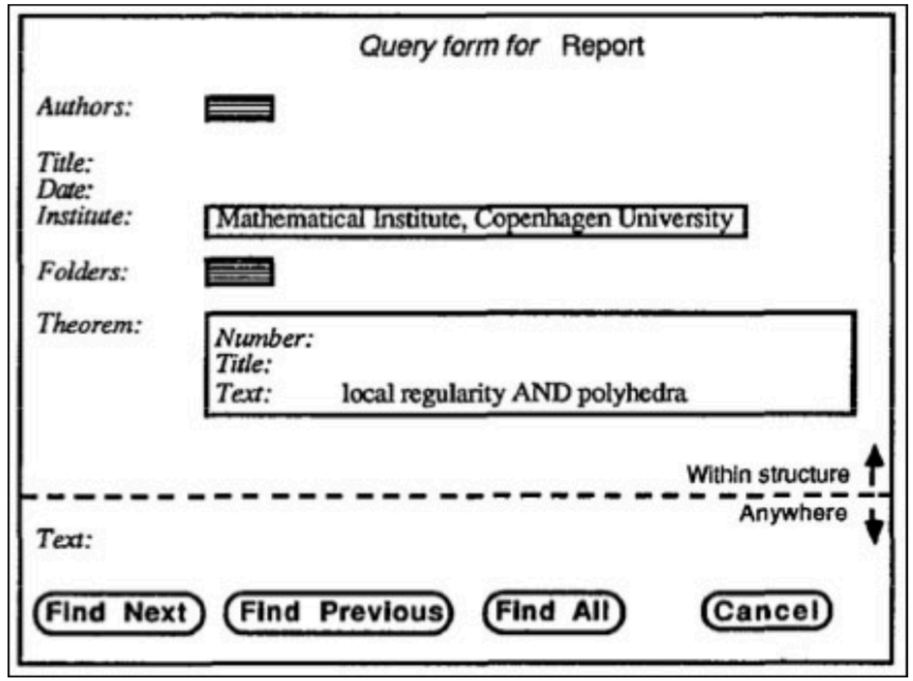


Figure 3.5: The EUROMATH interface had custom forms for each type of record in the system; taken from [123].

Images from Search User Interface Design, Max L. Wilson (2011)

Boolean Searching

- ➔ Boolean queries are on of the early advancement in the way users can specify their needs, e.g. [kings OR queens]
- ➔ The advent of GUIs provided an opportunity to help people construct Boolean queries more easily and visually.
- ➔ Boolean search is still prevalent in modern search user interfaces.

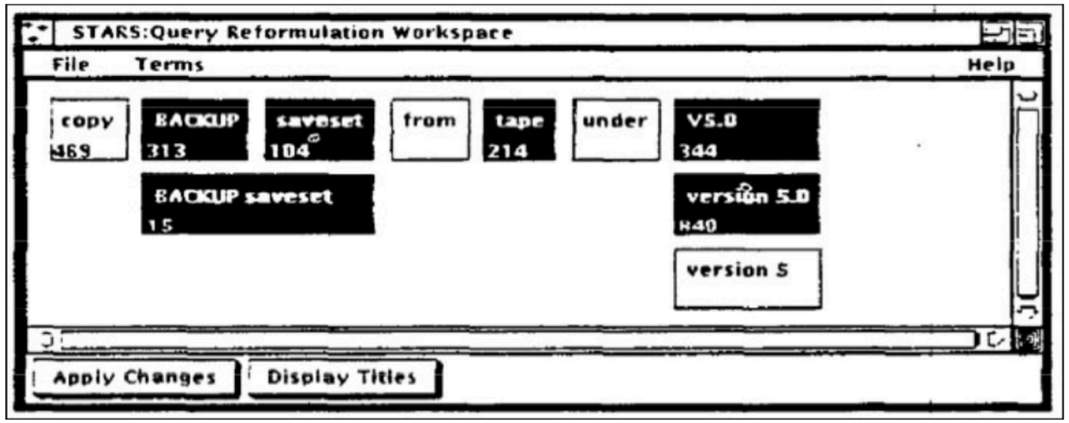


Figure 3.6: The query builder in the STARS interface allowed searchers to arrange query terms in a 2D space to build a Boolean query. Items aligned vertically are ORed and items aligned horizontally are ANDed; taken from [7].

Images from Search User Interface Design, Max L. Wilson (2011)

Modern Search User Interfaces

Modern Search User Interfaces

- ➔ Advances in modern search user interfaces designs are describes according to the framework presented before, where elements and features are grouped in: input; control; informational, and personalization.

- ➔ For each search user interface design, examples and design recommendations are highlighted.

Input Features

- Input features typically try to provide either suggestions for keywords searchers, or metadata that can be browsed.

- The **search box** pervades search user interfaces and users expect to find a white text field to insert their search terms into.
  - Advantages: extremely flexible, and uses the searchers language.
  - Search box also informs the user of the search being made, thus an informational feature.
  - Additionally, keeping the search box visible allows users to use it as a control feature too.

Adding Metadata

- ➔ Metadata provided by the user can vary in both depth and complexity.
- ➔ Metadata can provide input, control, and informational support.
- ➔ An example is faceted metadata, e.g. grouping results in multiple different dimensions (facets).

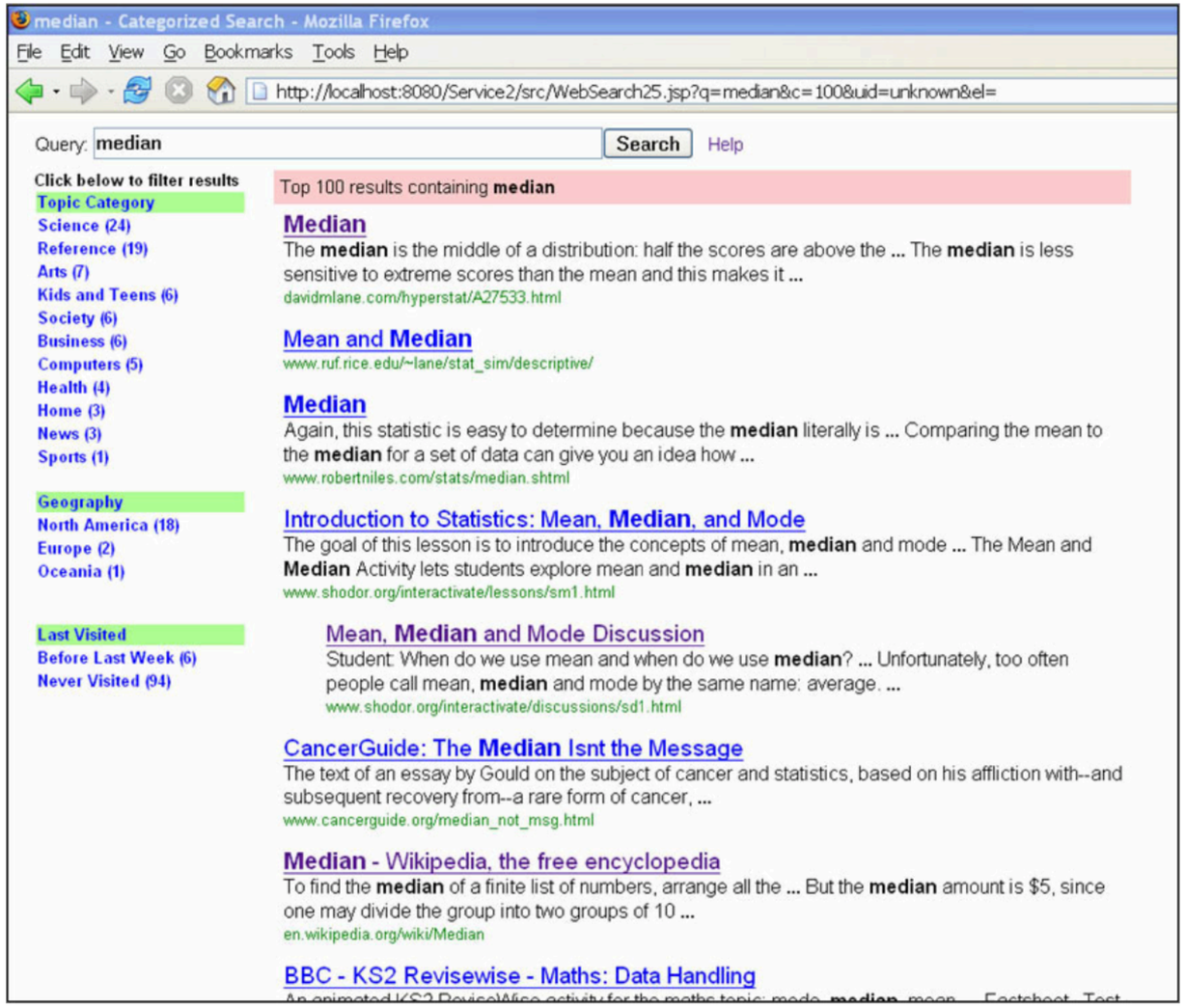


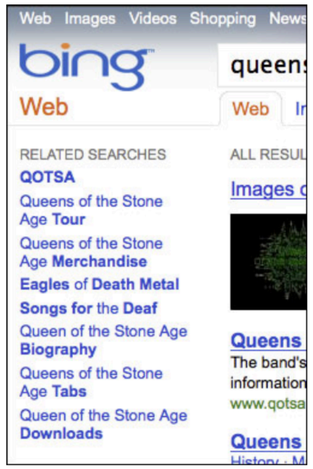
Figure 4.8: SERVICE provides faceted browsing over generic web search results.

Images from Search User Interface Design, Max L. Wilson (2011)

Control Features

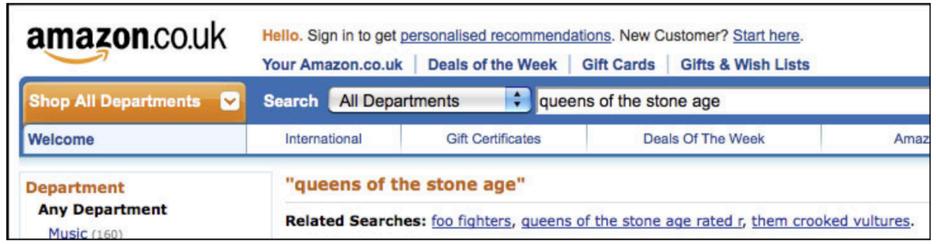
- Control features can affect either the search that was being submitted, or can organize or rearrange the results.
- Examples of control features:
  - Interactive query changes, e.g. "related searches", "more like this".
  - Corrections, e.g. auto-correcting errors.
  - Sorting, e.g. provide features that allow users to re-order results.
  - Filters, e.g. filter results based on properties.

Interactive Query Changes



(a) Bing – IQEs on left of SERP

(b) Google – IQEs for some searches at the bottom of the results.

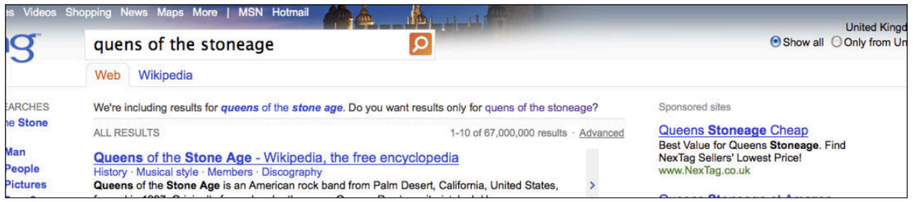


(c) Amazon provides fewer IQEs above the results.

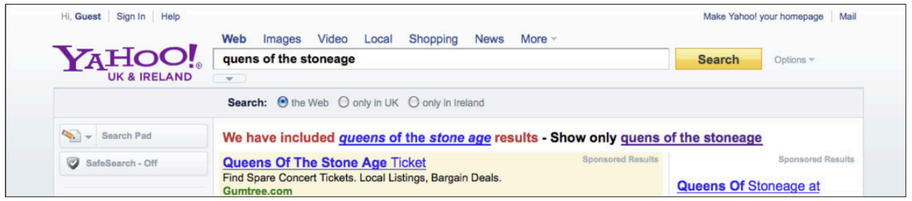
Figure 4.10: Interactive Query Suggestions (both refinements and alternatives) are a familiar sight in most search systems.

Images from Search User Interface Design, Max L. Wilson (2011)

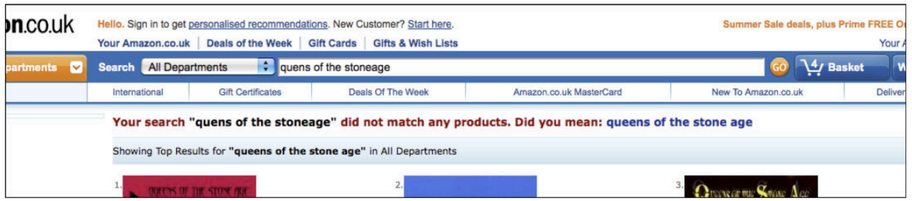
Corrections



(a) Bing (shown here) and Google often auto-corrects errors, but provides a link to results on the exact query, if it is confident that the corrected version is more likely.



(b) Yahoo! currently only auto-corrects errors, but provides a link to results on the exact query.

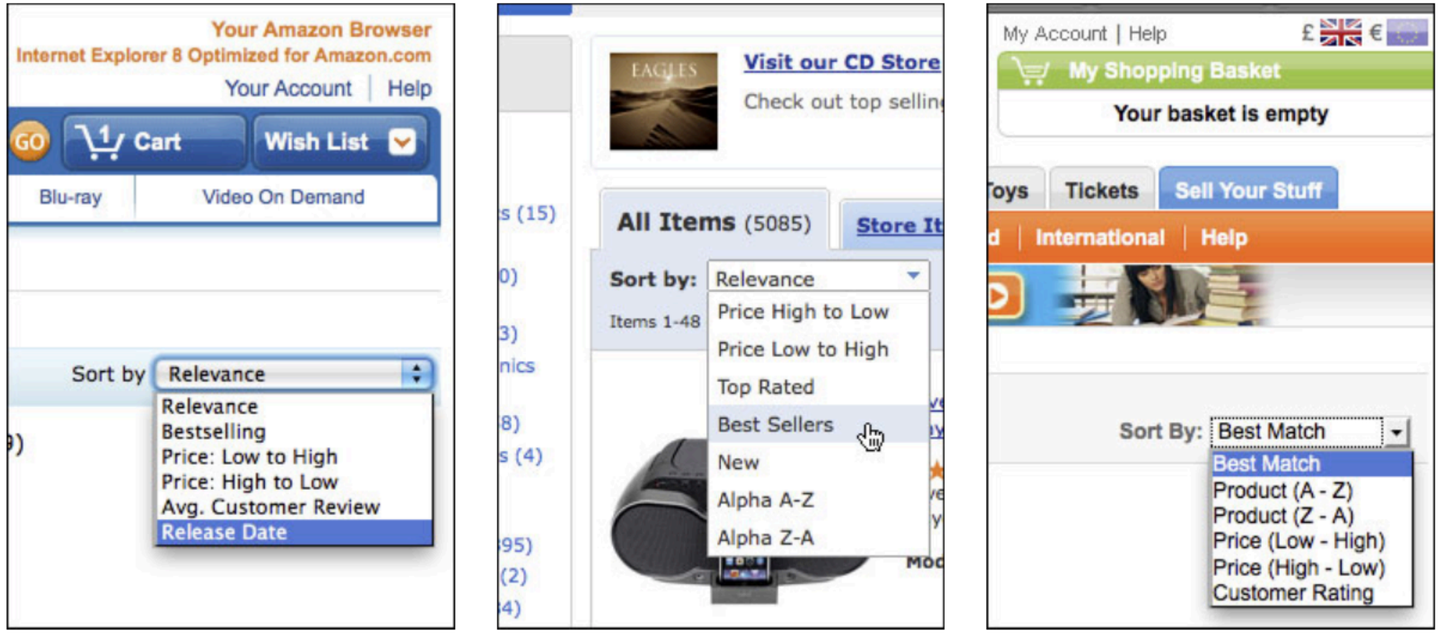


(c) Amazon says it cannot find results for an incorrect query, but provides a link and example results for a corrected version.

Figure 4.11: Correcting errors in searcher queries to avoid dead ends.

Images from Search User Interface Design, Max L. Wilson (2011)

Sorting



(a) Sorting in Amazon (b) Sorting in Walmart (c) Sorting in Yahoo!

Images from Search User Interface Design, Max L. Wilson (2011)

Filters

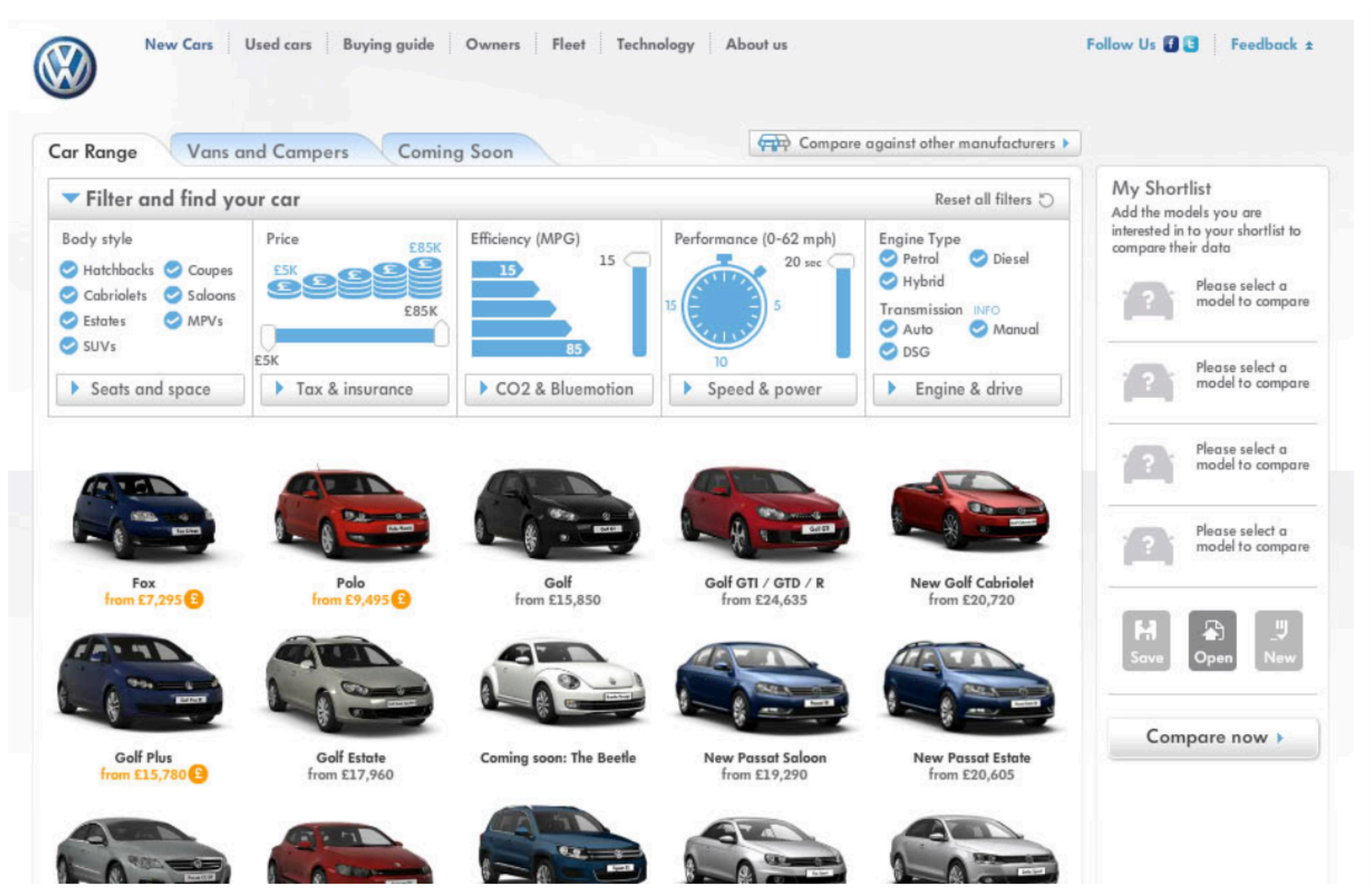


Figure 4.14: Volkswagen use a dynamic and interactive set of sliders and checkboxes to filter their cars.

Images from Search User Interface Design, Max L. Wilson (2011)

Informational Features

- ➔ Informational features relate to how results are organized and displayed.
- ➔ Individual results are presented in a Search Engine Results Page (SERP).
- ➔ An individual result typically includes: (1) the title of the result; (2) a snippet of text; and (3) the URL for the result.

https://pt.wikipedia.org › wiki › Recuperação_de_infor...

Recuperação de informação – Wikipédia, a enciclopédia livre

É uma ciência de pesquisa sobre busca por **informações** em documentos, busca pelos documentos propriamente ditos, busca por metadados que descrevam documentos e ...

Histórico acadêmico · Palavras-chave · Principais passos · Esquema global

https://www.marilia.unesp.br › EdbertoFerneda PDF

Recuperação de Informação

14/08/2018 — de **recuperação de informação** dos sistemas gerenciadores de bancos de dados, estudados e implementados no âmbito da Ciência da Computação.

26 páginas

Personalization Features

- With personalization features the search experience is tailored to the searcher.
- Personalization can be based on:
  - Individual data, e.g. previous user interactions, previous searchers, user location;
  - Aggregated data, e.g. based on activity profiles, location data, topical data.
- Personalization can impact:
  - Results ranking;
  - Search suggestions;
  - Search engine features.

Evaluating Search User Interfaces

Evaluating Search User Interfaces

- Three approaches: IR style, empirical user studies, analytical approaches.
- IR style
  - Traditionally, IR systems are evaluated within a TREC-style environment, based on datasets, specific tasks, and known 'best results' for each task (calculated by human experts).
  - Given the success of the TREC approach, an interactive track was created to take search interaction into account. But success was limited. Simple evaluation measures like precision and recall were not sufficient.
- Other approaches are: empirical and analytical.

Empirical User Studies

- ➔ HCI user studies methods instead focus on how well a system, including the SUI, allows searchers to complete a task.
- ➔ It became common to evaluate search systems by creating user studies using tasks that are specifically oriented towards search.
- ➔ Empirical methods are about observing and recording actual user performance.
- ➔ Common measurements used in user studies: number of searches, number of terms per search, number of results visited, search times, task accuracy, etc. Qualitative methods such as interviews and observations are also possible.
- ➔ Designing and conducting user studies is hard. Results can be impacted by many factors such as the motivation of participants, software bugs, and even small user experience differences, e.g. slight differences in colors.

Analytical Approaches

- ➔ In analytical approaches, typically low-cost or "discount" inspection methods are used to allow evaluators to assess a design and make well informed predictions about SUI.
- ➔ There are many analytics methods for UI and UX — e.g. heuristic evaluation, cognitive walkthrough —, but fewer specifically designed for SUI.
- ➔ Analytical approaches can leverage the wealth of models and experience previously developed by experts.
- ➔ Analytics methods can only make estimates about how suitable a design is, before a formal evaluation.

Choosing an Evaluation Approach

- The HCI community has developed the DECIDE process to help on this decision:
  - D - Determine the goals of the evaluation;
  - E - Explore the specific questions to be answered;
  - C - Choose an evaluation paradigm, such as systematic IR or empirical user studies;
  - I - Identify practical issues in performing such an evaluation;
  - D - Decide how to deal with any ethical issues;
  - E - Evaluate, Interpret, and present the data.

Summary of Design Recommendations

Summary of Design Recommendations (1)

1. Keep the search box and the current query clearly visible for the searcher at all times.
2. Help searchers to create useful queries whenever possible.
3. Make it clear how results relate to metadata in your system, to help searchers to judge the results and make sense of the whole collection.
4. Carefully curated metadata is better than automatically generated, but both are better than no metadata at all.
5. If appropriate, support searchers in reviewing their decisions and their options quickly and easily.
6. Always return results based on the first interaction, as subsequent interactions may never be needed.
7. Never let searcher reach a dead end, where they have to go back or start over.
8. Help searchers avoid mistakes wherever possible, but do not force that help upon them.

Summary of Design Recommendations (2)

9. Give searchers control over the way results are ordered.
10. Make sure it is obvious exactly how results are ordered and which, therefore, are most important.
11. Avoid unnecessary information, which can be distracting during search.
12. Searchers rarely scroll, so get ‘important' information above the first-scroll point.
13. Provide actionable features in the SERP results directly so that searchers do not have to interrupt their search.
14. Images and Previews can help searchers make better browsing decisions.
15. Make sure the dimensions and layout of a visualization are clear and intuitive to the searcher.
16. Guiding numbers help searchers to make better browsing decisions.

Summary of Design Recommendations (3)

17. Animation should be used carefully and purposefully to convey a message, such as change.
18. Track and reuse information about the behavior of a system's searchers.
19. Help searchers to return to previously viewed SERPs and results.
20. Help searchers to recover their previous search sessions, as they may be back to finish a task.

Summary

Summary

- ➔ **Multidisciplinary**, addressing these challenges requires knowledge and expertise from multiple areas, e.g. information science, visual design, human-computer interaction.
- ➔ **Different types of features**, search user interfaces features can be grouped in different types — input features; control features; informational features; personalization features.
- ➔ **Heuristics**, existing guidelines and best practices can help on the design of successful search user interfaces.
- ➔ **User studies**, evaluation of search user interfaces is done by observing users conducting specific tasks.

References

- Max Wilson, Search User Interface Design (2011)
- Marti A. Hearst, Search User Interfaces (2009)
  - Available online at: https://searchuserinterfaces.com
- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval (2011)
  - Chapter 2: User Interfaces for Search + Slides

IR Topics

PRI 22/23 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Outline

- ➔ Learning to Rank
- ➔ Neural Information Retrieval
- ➔ Vector Search

Learning to Rank

Based on Learning to Rank for Information Retrieval, Tie-Yan Liu (2009)

Learning to Rank

- *Learning to Rank (LtoR) (Liu, 2009)*
  - *is a task to automatically construct a ranking model*
  - *using training data, such that*
  - *the model can sort new objects according*
  - *to their degrees of relevance, preference, or importance.*

- Motivated by the large volume of information available
  - Challenge in ranking millions of documents
  - Opportunity to use existing data for ranking

Ranking Problems

- Many information retrieval problems are by nature ranking problems, such as:
  - document retrieval,
  - collaborative filtering,
  - key term extraction,
  - definition finding.
- Many different ranking scenarios:
  - Rank document purely according to their relevance with regards to the query.
  - Considering the relationships of similarity and diversity between documents (e.g. relational ranking).
  - Aggregate several candidate ranked lists (e.g. meta search, unified ranking).
  - Find to what degree a document property influences the ranking result.

Ranking in Information Retrieval

- Conventional document ranking models:
  - Query-dependent models:
    - Boolean model, set based model (no degree of relevance)
    - Vector space model, using term weighting such as TF-IDF.
    - Probabilistic models, such as language models and BM25.
  - Query-independent models:
    - Link-based models for the web, such as PageRank or HITS

Learning to Rank

- ➔ Many of the existing ranking models contain parameters that need to be tuned.
- ➔ Different ranking models can be combined to create a new ranking.
- ➔ In addition, models perfectly tuned using the validation set do not necessary perform well on unseen queries.

- ➔ How to define and combine parameters and models to produce an effective ranking?
- ➔ Machine learning methods have demonstrated their effectiveness in automatically tuning parameters combining multiple evidences.
- ➔ Learning to rank is the application of machine learning methods to information retrieval problems.

Typical Learning to Rank Flow

- The training data consists of:
  - n queries (q1, q2, …),
  - their associated documents represented as features vectors (x1, x2, …),
  - and the corresponding relevance judgements.
- A learning algorithm is employed to learn the ranking model h.
- The ranking model h is then evaluated with standard IR measures by producing ranked lists for the training data.

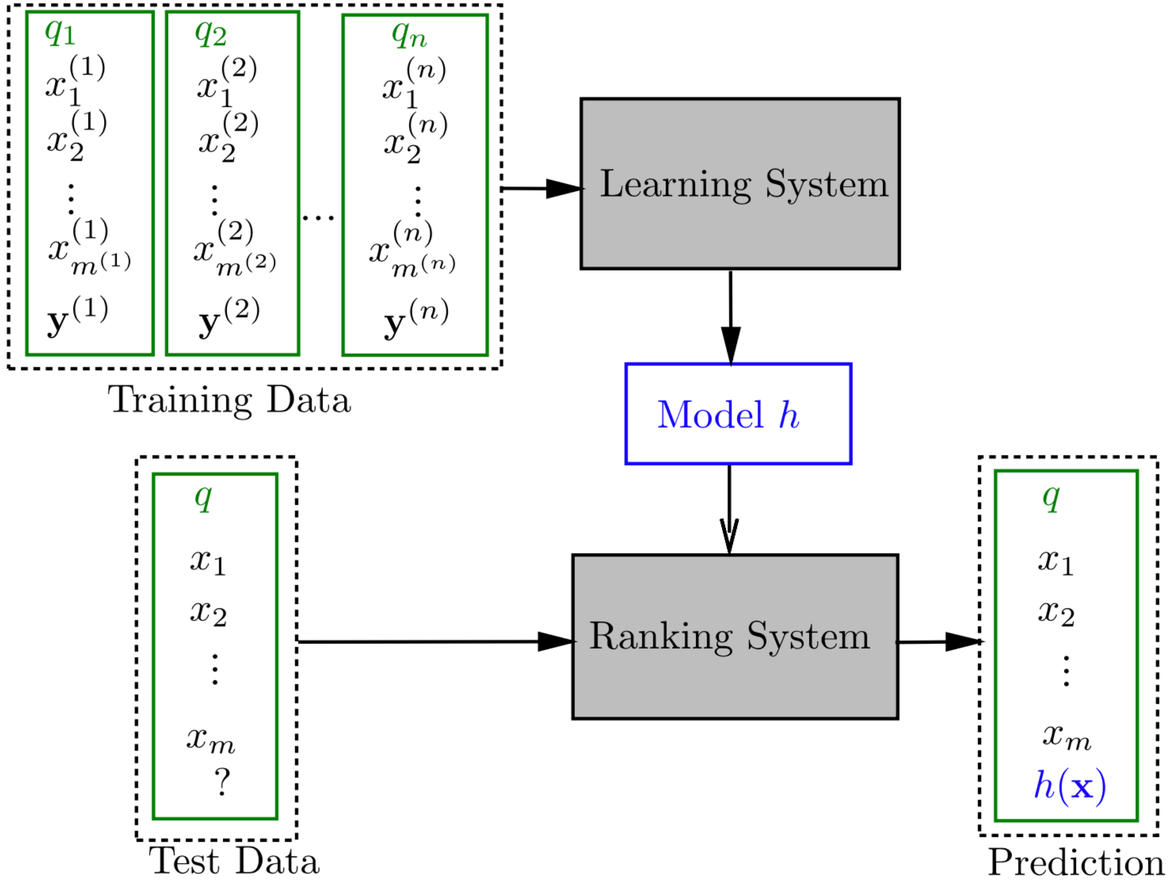


Fig. 1.1 Learning-to-rank framework.

Image from Learning to Rank for Information Retrieval, Tie-Yan Liu (2009)

Popular Features

Table 6.2 Learning features of TREC.

ID	Feature description
1	Term frequency (TF) of body
2	TF of anchor
3	TF of title
4	TF of URL
5	TF of whole document
6	Inverse document frequency (IDF) of body
7	IDF of anchor
8	IDF of title
9	IDF of URL
10	IDF of whole document
11	TF*IDF of body
12	TF*IDF of anchor
13	TF*IDF of title
14	TF*IDF of URL
15	TF*IDF of whole document
16	Document length (DL) of body
17	DL of anchor
18	DL of title
19	DL of URL
20	DL of whole document
21	BM25 of body
22	BM25 of anchor
23	BM25 of title

Table 6.2 (*Continued*)

ID	Feature description
40	LMIR.JM of whole document
41	Sitemap based term propagation
42	Sitemap based score propagation
43	Hyperlink base score propagation: weighted in-link
44	Hyperlink base score propagation: weighted out-link
45	Hyperlink base score propagation: uniform out-link
46	Hyperlink base feature propagation: weighted in-link
47	Hyperlink base feature propagation: weighted out-link
48	Hyperlink base feature propagation: uniform out-link
49	HITS authority
50	HITS hub
51	PageRank
52	HostRank
53	Topical PageRank
54	Topical HITS authority
55	Topical HITS hub
56	Inlink number
57	Outlink number
58	Number of slash in URL
59	Length of URL
60	Number of child page
61	BM25 of extracted title
62	LMIR.ABS of extracted title
63	LMIR.DIR of extracted title
64	LMIR.JM of extracted title

Table from Learning to Rank for Information Retrieval, Tie-Yan Liu (2009)

Learning to Rank Approaches

- The learning to rank process can be modeled in different ways.
  - Pointwise approach
  - Pairwise approach
  - Listwise approach

- Different learning techniques can be employed, including SVM, Boosting, Neural Nets, etc.

- Effectiveness from literature: Listwise > Pairwise > Pointwise
- Different methods can be combined with successful results.

Pointwise Approach

- ➔ Input: feature vectors for single documents, e.g. scores for query-document pairs.
- ➔ Output: relevance degree of each single document (e.g., relevant / non-relevant).
- ➔ Methods: OC SVM, McRank, Prank.

- ➔ Treat the problem as a regression problem, i.e. estimate a continuous variable (i.e. each document's score).
- ➔ Straightforward approach.
- ➔ Ambitious goal — produce document scores, but only rankings are needed.

Pairwise Approach

- ➔ Pairwise classification problem, i.e. decide pairwise preferences for documents.
- ➔ Input: pairs of documents, both represented as feature vectors.
  - ➔ Manually annotated data, e.g. (q,d,d’) — d is more relevant to q than d’ (harder to get, high quality).
  - ➔ Log data, e.g. if, for query q, document d and d’ are listed, and user clicked d and not d’, then (q,d,d’) can be inferred (easy, lower quality).
- ➔ Output: pairwise preferences (ranging from 1 to -1) between document pairs.

- ➔ The problem is treated as a classification problem, i.e. given a query and two documents, determine which is better.
- ➔ Methods: Ranking SVM, LambdaMART, LambdaRank

Listwise Approach

- Input: entire group of documents associated with a query, i.e. ranking lists.
  - Offline: obtain relevance judgements (q,d,s), where s is a score indicating the relevance of document d to query q.
  - Online: present different rankings to users (or interleave lists) and observe (from logs) which users prefer.
- Output: full document predicted ranking for query.
- Methods: PermuRank, AdaRank, SoftRank.

- Problem is modeled in a more natural “IR” way but not directly adaptable to conventional machine learning techniques.
- Best performing approaches.

Reference Datasets for Learning to Rank

- LETOR: Learning to Rank for Information Retrieval
  - www.microsoft.com/research/project/letor-learning-rank-information-retrieval
- Microsoft Learning to Rank Datasets
  - www.microsoft.com/research/project/mslr

Support in Solr

- Solr includes support for Learning to Rank.
- https://solr.apache.org/guide/learning-to-rank.html
  - Includes a step-by-step practical example.

Neural Information Retrieval

Based on An Introduction to Neural Information Retrieval, B. Mitra and N. Craswell (2018).
Based on Lecture Notes on Neural Information Retrieval, N. Tonellotto (2022).

Neural Information Retrieval

- Neural IR is the application of shallow or deep neural networks to IR tasks.
  - Neural models have been employed in many IR scenarios—including ad-hoc retrieval, recommender systems, multimedia search, and even conversational systems that generate answers in response to natural language questions.
- Unlike Learning to Rank approaches that train machine learning models over a set of hand-crafted features, neural models accept the raw text of a query and document as input.

- Neural networks are typically used in two different ways:
  - Learn the ranking functions combining the relevance signals to produce an ordering of documents.
  - Learn the abstract representations of documents and queries to capture their relevance information.

Data in Neural Information Retrieval

- ➔ Neural models for IR use vector representations of text, which usually contain a large number of parameters that need to be tuned.
- ➔ ML models with large set of parameters typically benefit from large quantity of training data.
- ➔ Learning suitable representations of text demands large-scale datasets for training.
- ➔ Therefore, unlike classical IR models, these neural approaches tend to be data hungry, with performance that improves with more training data.

Document Ranking

- Document ranking comprises three steps (recall IR models):
  - Generate a representation of the query that specified the information need.
  - Generate a representation of the document.
  - Match the query and the document representations to estimate their mutual relevance.

- Neural approaches can influence one or more of these three steps.
- Unlike traditional learning to rank models, neural architectures depend less on manual feature engineering on more on automatically detecting regularities in matching patterns.

Document Ranking

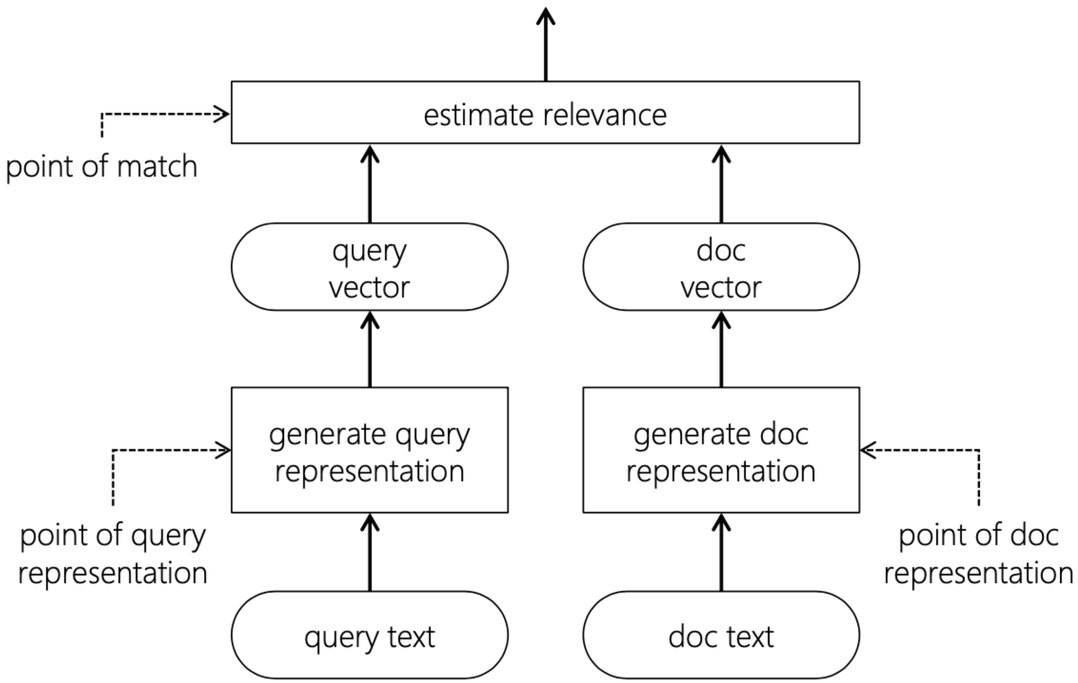
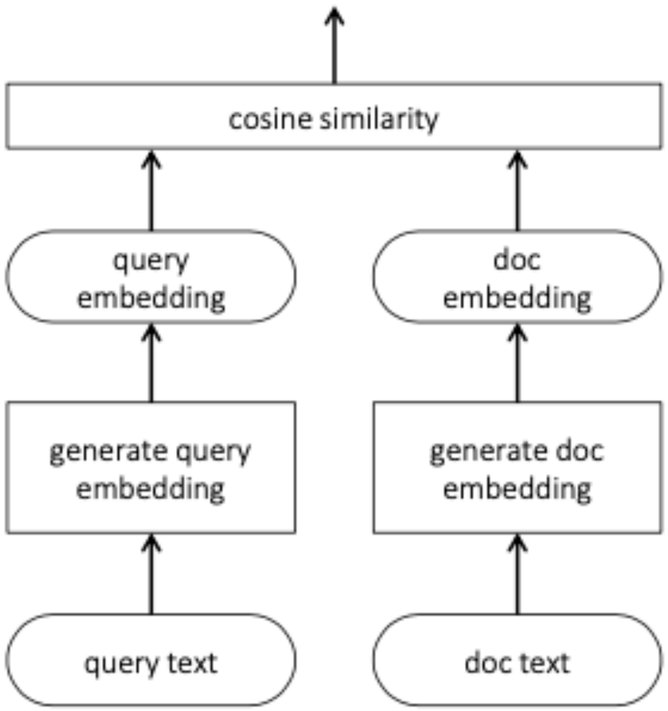


Figure 2.3: Document ranking typically involves a query and a document representation steps, followed by a matching stage. Neural models can be useful either for generating good representations or in estimating relevance, or both.

Image from An Introduction to Neural Information Retrieval, B. Mitra and N. Craswell (2018).

Document Ranking

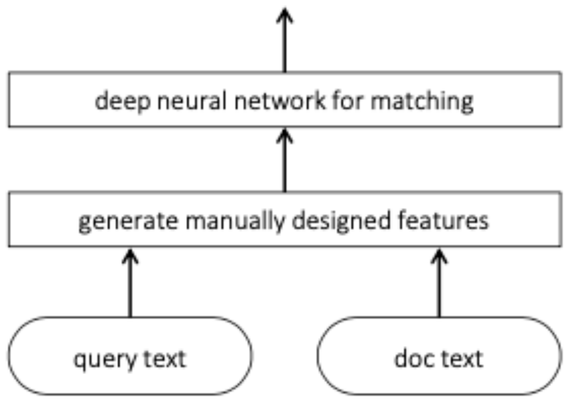
- ➔ Many neural IR models depend on learning low-dimensional vector representations (or embeddings) of query and document text.
- ➔ And then use them with traditional IR models in conjunction with simple similarity metrics (e.g., cosine similarity).



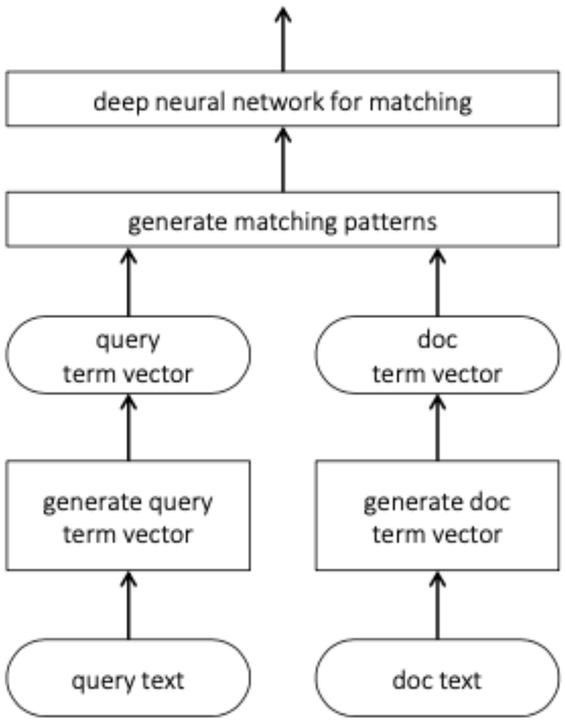
(c) Learning query and document representations for matching (*e.g.*, (Huang *et al.*, 2013; Mitra *et al.*, 2016a))

Image from An Introduction to Neural Information Retrieval, B. Mitra and N. Craswell (2018).

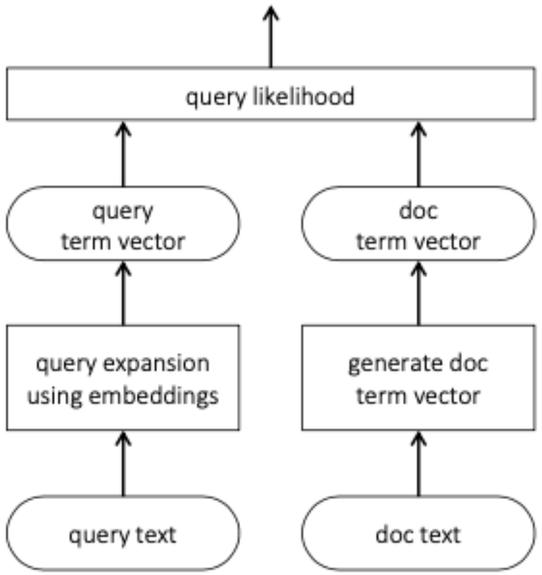
Document Ranking



(a) Learning to rank using manually designed features (*e.g.*, Liu (2009))



(b) Estimating relevance from patterns of exact matches (*e.g.*, (Guo *et al.*, 2016a; Mitra *et al.*, 2017a))



(d) Query expansion using neural embeddings (*e.g.*, (Roy *et al.*, 2016; Diaz *et al.*, 2016))

Image from An Introduction to Neural Information Retrieval, B. Mitra and N. Craswell (2018).

Word Embeddings

- ➔ In the 1950s, many linguists formulated the distributional hypothesis: words that occur in the same contexts tend to have similar meanings.
- ➔ According to this hypothesis, the meaning of words can be inferred by their usage together with other words in existing texts.
- ➔ “You shall know a word by the company it keeps.”, John Firth (1957)
- ➔ In recent years, the distributional hypothesis has been applied to create semantic understandings of terms and term sequences through word embeddings.
- ➔ A word embedding is a numerical vector that represents the semantic meaning of a given term sequence.

Recall Sparse Vectors Representation

query | exact term lookup in inverted index

	apple	caffeine	cheese	coffee	drink	donut	food	juice	pizza	tea	water
latte	0	0	0	0	0	0	0	0	0	0	0
cappuccino	0	0	0	0	0	0	0	0	0	0	0
apple juice	**1**	0	0	0	0	0	0	**1**	0	0	0
cheese pizza	0	0	**1**	0	0	0	0	0	**1**	0	0
donut	0	0	0	0	0	**1**	0	0	0	0	0
soda	0	0	0	0	0	0	0	0	0	0	0
green tea	0	0	0	0	0	0	0	0	0	**1**	0
water	0	0	0	0	0	0	0	0	0	0	**1**
cheese bread sticks	0	0	**1**	0	0	0	0	0	0	0	0
cinnamon sticks	0	0	0	0	0	0	0	0	0	0	0

Figure 2.10 Vectors with one dimension per term in the inverted index. Every query on the left maps to a vector on the right, with a value of "1" for any term in the index that is also in the query, and a "0" for any term in the index that is not in the query.

Image from AI Powered Search, T. Grainger et al. (2022).

Word Embeddings / Dense Vectors

	food	drink	dairy	bread	caffeine	sweet	calories	healthy
apple juice	0	5	0	0	0	4	4	3
cappuccino	0	5	3	0	4	1	2	3
cheese bread sticks	5	0	4	5	0	1	4	2
cheese pizza	5	0	4	4	0	1	5	2
cinnamon bread sticks	5	0	1	5	0	3	4	2
donut	5	0	1	5	0	4	5	1
green tea	0	5	0	0	2	1	1	5
latte	0	5	4	0	4	1	3	3
soda	0	5	0	0	3	5	5	0
water	0	5	0	0	0	0	0	5

Figure 2.11 Word embeddings with reduced dimensions. In this case, instead of one dimension per term (exists or missing), now higher-level dimensions exist that score shared attributes across items such as "healthy", contains "caffeine" or "bread" or "dairy", or whether the item is "food" or a "drink".

Image from AI Powered Search, T. Grainger et al. (2022).

Transformers

- Static word embeddings map words with multiple senses into an average or most common-sense representation based on the training data used to compute the vectors.
  - The vector of a word does not change with the other words used in a sentence around it.
- A transformers is a neural network designed to explicitly take into account the context of arbitrary long sequences of text.
  - Transformer compute contextualized word embeddings, where the representation of each input token is conditioned by the whole input text.
  - The most popular contextualized word embeddings are learned with deep neural networks such as the Bidirectional Encoder Representations from Transformers (BERT).
- With fine-tuning, the parameters of a pre-trained language model can be updated for the domain data and target task.
  - Pre-training typically requires a huge general-purpose training corpus, and long and expensive computation resources.
  - On the other side, fine-tuning requires a small domain-specific corpus focused on the downstream task, affordable computational resources and few hours or days of additional training.

Retrieval Architectures

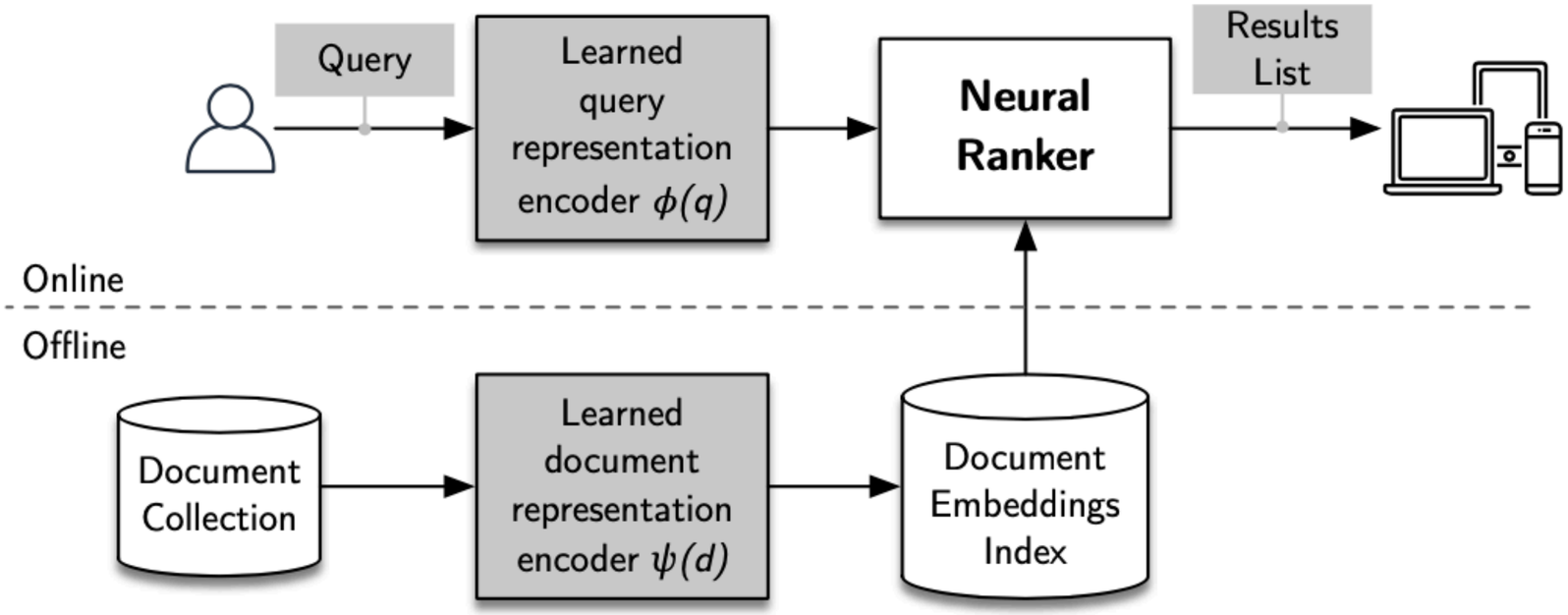


Figure 8: Dense retrieval architecture for representation-focused neural IR systems.

Image from Lecture Notes on Neural Information Retrieval, N. Tonellotto (2022).

Vector Search Systems

- ➔ Also called vector databases.
- ➔ In dense retrieval systems, document embeddings are pre-computed, thus the need for storing and searching through these document embeddings.
- ➔ Given a query, represented as a dense vector, instead of computing the distance between this vector and all document dense vectors (too expensive), the strategy is to use approximate nearest neighbor search.

- ➔ Solr (since version 9) includes support for Dense Vector Search.
- ➔ https://solr.apache.org/guide/solr/latest/query-guide/dense-vector-search.html

Vector Search Open-Source Frameworks

- Milvus, https://milvus.io
- Vespa, https://vespa.ai
- Qdrant, https://qdrant.tech

- Vector Search benchmarks
  - http://ann-benchmarks.com

References

- Liu, Tie-Yan. Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3.3 (2009): 225-331.
- Mitra, Bhaskar, and Nick Craswell. An introduction to neural information retrieval. Foundations and Trends in Information Retrieval 13.1 (2018): 1-126.
- Tonellotto, Nicola. Lecture Notes on Neural Information Retrieval. 2022.
- Grainger, Trey, Doug Turnbull, and Max Irwin. AI Powered Search. Manning (2022).