INFORMATION RETRIEVAL

Prof. Marco Viviani
marco.viviani@unimib.it

The «information overload» problem

- The term "**information overload**" was coined by Bertram Gross, Professor of Political Science at Hunter College, in his 1964 work – The Managing of Organizations.

- It was popularized by Alvin Toffler, the American writer and futurist, in his book "Future Shock" in 1970.

"Information overload occurs when the amount of input to a system exceeds its processing capacity. Decision makers have fairly limited cognitive processing capacity. Consequently, when information overload occurs, it is likely that a reduction in decision quality will occur."

Automatic access to information

- Development of systems that help the user to identify information **relevant** to their **needs** (to inform, i.e., reduce ignorance).

- The definition of such systems is based on the solution of a **decision problem**: how to identify and define the importance of information that satisfies the user's preferences? It is necessary to:
  - interpret the content of texts, images, video, audio.
  - interpret the user's needs.

- Central role of the notion of **relevance**: relevance is a **subjective** property: difficult to define and measure!

Main systems for accessing information

- **DataBase Management Systems** (DBMS)
  - They requires the formulation of a **query**.

- **Information Retrieval Systems** (IRS, Search Engines)
  - They requires the formulation of a **query**.

- **Information Filtering Systems** (IFS, Recommender Systems)
  - They requires **user profiles**, i.e., descriptions of specific needs dynamically updated, also based on the user behavior (**NO query**).

Main systems for accessing information

- **DataBase Management Systems** (DBMS)
  - They requires the formulation of a **query**.

- **Information Retrieval Systems** (IRS, Search Engines)
  - They requires the formulation of a **query**.

- **Information Filtering Systems** (Recommender Systems)
  - They requires **user profiles**, i.e., descriptions of specific needs dynamically updated, also based on the user behavior (**NO query**).

BASICS OF IR

Defining Information Retrieval (IR)

- IR is the computer science discipline that deals with the storage and retrieval of **documents**.

- Its goal is the creation of software systems that allow the storage of **large** quantities of documents in an archive, to allow an **efficient** and **effective** retrieval of documents **relevant** to users' **information needs**.

"IR is the name for the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him (…). IR embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, techniques, and machines that are employed to carry out the operation."
[Mooers, 1951]

Main Information Retrieval tasks

- **Ad-hoc retrieval**
  - Ranked document retrieval.
  - In an operational search engine, the retrieval system uses specialized index structures to search potentially billions of documents.
  - The results ranking is presented in a search engine results page (SERP).

- **Question-answering**
  - Choosing between multiple choices.
  - Ranking spans of text or passages.
  - Synthesizing textual responses by gathering evidence from one or more sources.

Glossary

- **Document** (d): unit of information retrievable, expressed in free format (without the application of specific formats or schemes). The documents have informative content.
- **Archive** (D): set of documents accessible by means of an IRS; it can be centralized or distributed.
- **Textual IR** → scientific articles, letters, newspaper articles, legends of images or graphics, audio transcriptions, …
- **Multimedia IR** → images, graphics, audio (spoken or not spoken), video, …, stored in digital format.

Glossary

- **Large size**: distributed systems and the spread of the Web generated **very large bases of documents** (archives) (e.g., Google is now indexing billions of Web pages) → Big data.

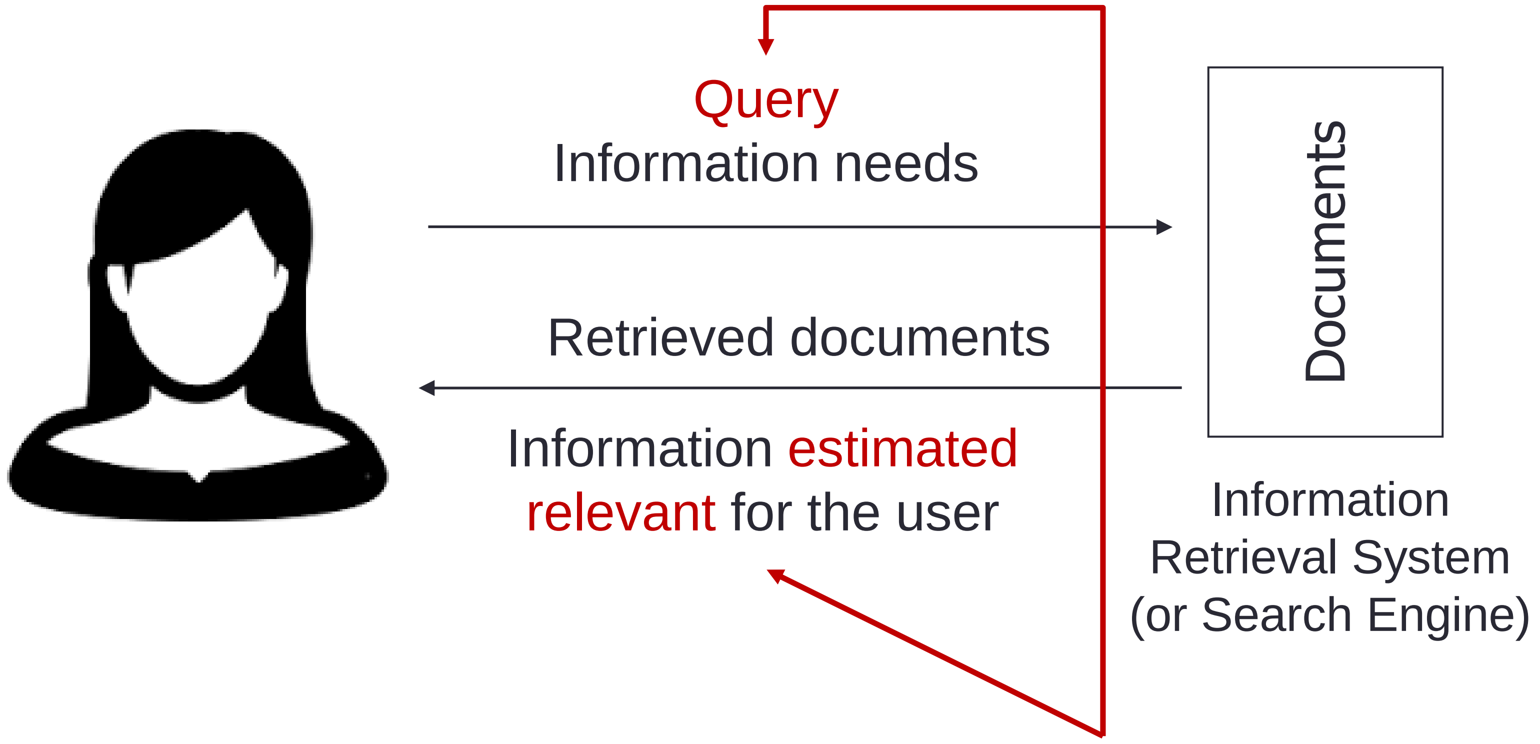
- **Information needs**: need for useful information to solve a problem; leads to the formulation of a **query**.

- **Relevance**: utility of a document for the user, estimated by the search engine with respect to the query that the user has formulated (but not only…).

Primary objective of an IR System

- Find all documents that are relevant to the user by neglecting non-relevant documents → **HOW?**

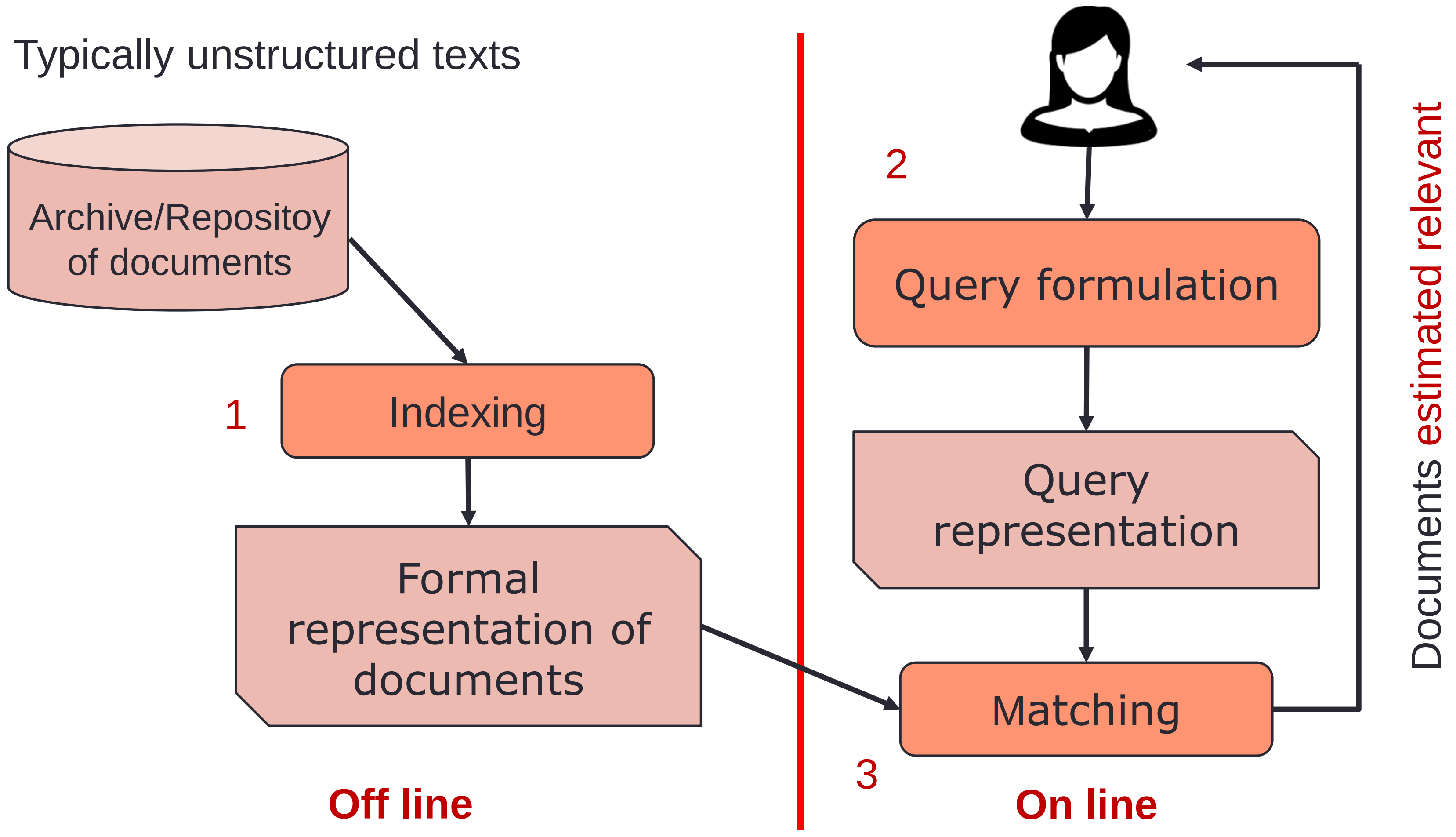
Information Retrieval System



Query
Information needs
Retrieved documents
Information estimated relevant for the user
Documents
Information Retrieval System (or Search Engine)

Subjectivity!

Basic structure of an IRS


Typically unstructured texts
Archive/Repositoy of documents
1
Indexing
Formal representation of documents
Off line
2
Query formulation
Query representation
Matching
3
On line
Documents estimated relevant

An IRS is based on a mathematical model!

Components of an IR System

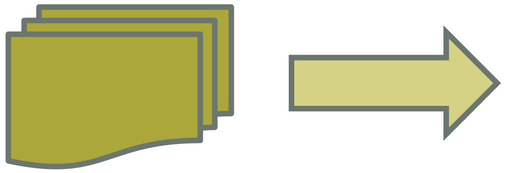
- **Document repository**: the document is the unit of information retrievable. It can consist of a text or be composed of narrative, pictorial, coded parts, etc. (Multimedia).
- **Formal representation of documents**: it formally represent the information content of documents. It is based on the output produced by the indexing process.
- **Query language**: in a query the conditions for the selection of documents are expressed.
- **Matching mechanism**: it "compares" the representation of the documents in the document repository with the selection conditions expressed in the query.
  - → (Topical) Relevance → **Retrieval Status Value** (RSV)

Indexing

The «indexing» process

“*The experimental evidence accumulated over the past 20 years indicates that text indexing systems based on the assignment of appropriately weighted single terms produce retrieval results that are superior than those obtainable with other more elaborate text representations. These results depend crucially on the choice of effective term-weighting systems*”.

[Salton, 1988]



{information, retrieval, computer, science, discipline, keywords, …}

The «indexing» process

- **Problem**: how to describe the “semantic” content of a document in an automatically manageable way?

- The indexing process is based on the extraction of **“elements” (features)** that form the basis of the **description (representation)** of the document.

- For texts, these elements (called **indexes**) are generally words.

- The documents are represented as **(weighted) sets of words**.
  - We have seen this already.

Automatic indexing of text documents

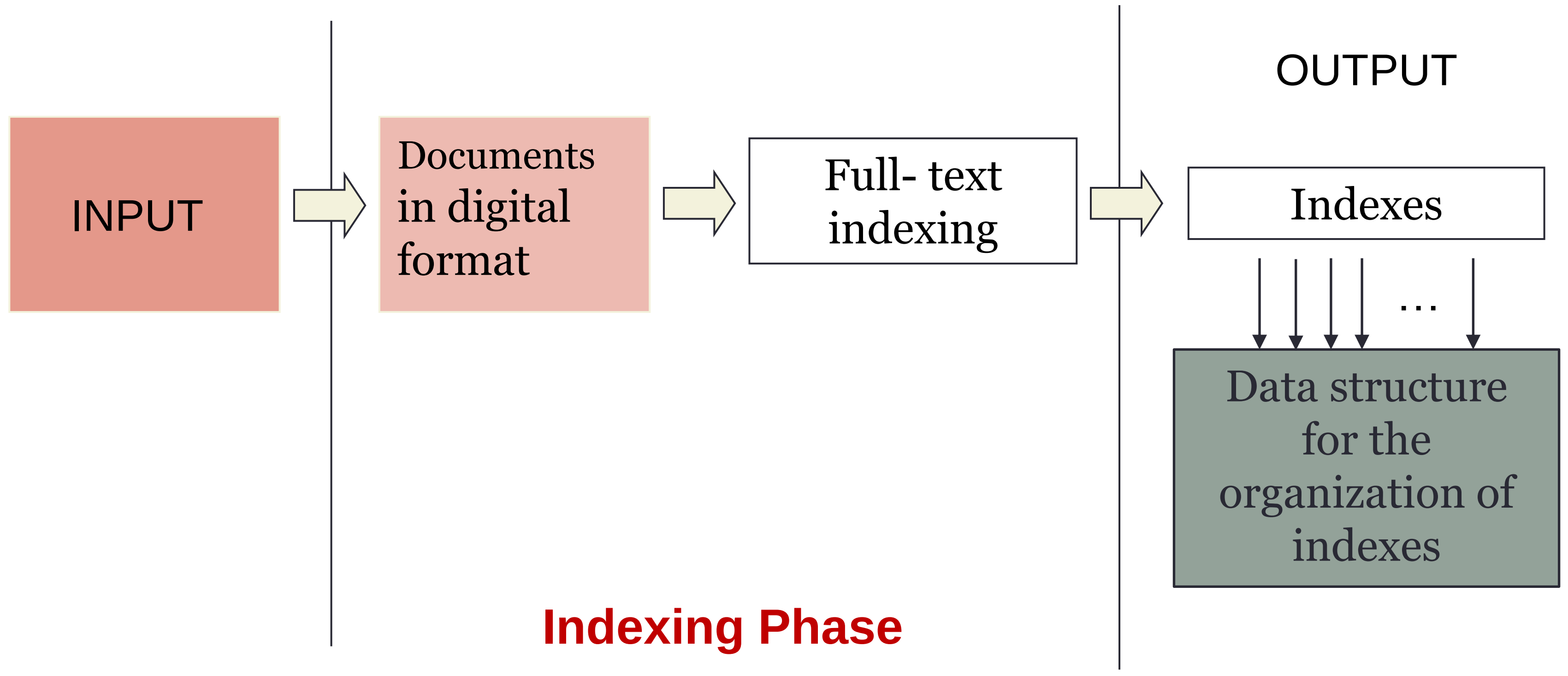
- The **automatic indexing** of a textual document is the process aimed at the association of **indexes** (index terms) with a text.

- Typically: **full-text indexing**.

- The use of the indexes makes the retrieval **efficient** based on the keywords specified in a query (example: analytical index of a book).

- **DICTIONARY** → All the indexes extracted / associated with all the documents of the collection considered constitute the dictionary of the collection.

Scheme of automatic indexing process


OUTPUT
INPUT
Documents in digital format
Full- text indexing
Indexes
…
Data structure for the organization of indexes
Indexing Phase

«Inverted file» structure (motivations)

DocID	Term1	Term2	Term3	Term4	Term5	Term6	Term7
Doc1	3	4	-	-	-	-	-
Doc2	-	-	1	2	6	5	8
Doc3	-	2	-	1	3	-	-
Doc4	-	-	-	-	4	7	6

- The organization of the indexes in a "**static**" data structure, for example a matrix of document/term occurrences would be inefficient (**forward index**).
  - Since the matrix is **sparse**, space would be wasted to store also the "non-occurrence" of the terms (e.g., Term3 in Doc1).
  - **Getting** the list of documents that contain a specific term would be burdensome.

«Inverted file» structure (motivations)

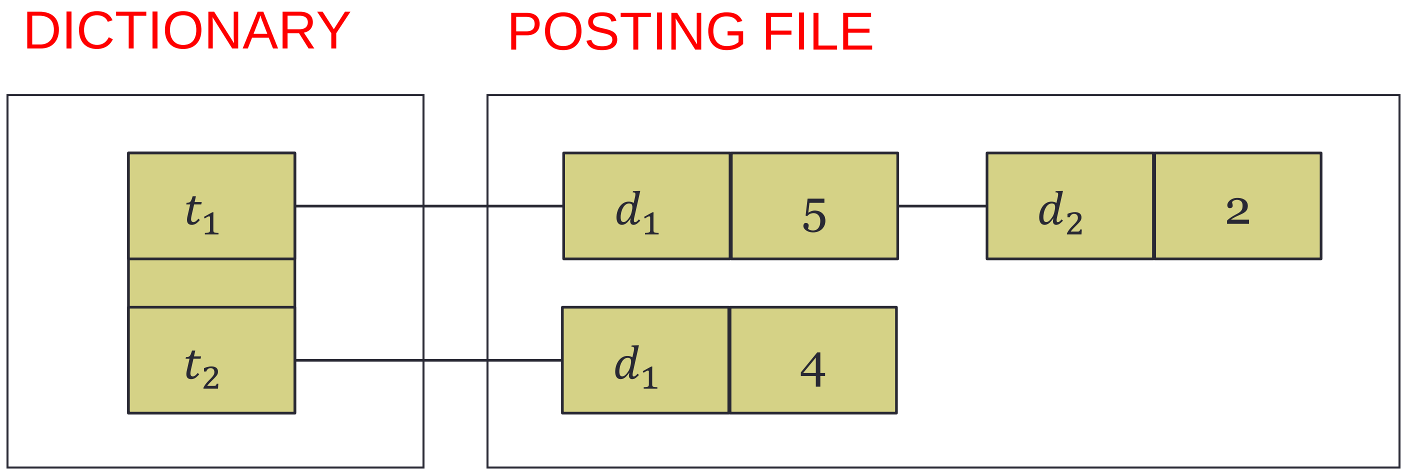
Term	Doc1	Doc2	Doc3	Doc4
Term1	3	-	-	-
Term2	4	-	2	-
Term3	-	1	-	-
Term4	-	2	1	-
Term5	-	6	3	4
Term6	-	5	-	7
Term7	-	8	-	6

- The table of occurrences must be **inverted** (term/document) and organized in a **dynamic** data structure.

«Inverted file» data structure

- The **index terms** are organized in a dictionary. Each term "points" to a list containing the references to documents of which the term is an index.

- Use of two files: **dictionary** and **posting file** (it contains the posting lists of all index terms).



«Inverted file» data structure

- Information associated with each **index**:
  - **The list of documents containing it** (in the posting file).
  - **Its occurrences in the collection** (in the dictionary) for the calculation of the IDF and to optimize the evaluation of Boolean queries).

Simple representation of the inverted file

- d_1: The GDP incrased 2 percent this quarter.
- d_2: The spring economic slowdown continued to spring downwards this quarter.

Dictionary File		Posting File
2	1	→ d_1
continued	1	→ d_2
downwards	1	→ d_2
economic	1	→ d_2
GDP	1	→ d_1
increased	1	→ d_1
percent	1	→ d_1
quarter	2	→ d_1, d_2
slowdown	1	→ d_2
spring	1	→ d_2
this	2	→ d_1, d_2
the	2	→ d_1, d_2
to	1	→ d_2

«Inverted file» data structure

- The **dictionary** (may) contain <u>for each index</u>: term, global frequency in the dictionary, pointer to the posting list in the posting file.

- The **posting list** (may) contain <u>for each element</u>:
  - The unique document identifier (DocID) that is associated with a file name or URL location.
  - The frequency of the term in the document (tf).
  - The position in the document of each occurrence of the term (optional, only for query with context). The position can be expressed as:
    - word number from the beginning of the document,
    - number of bytes from the beginning of the document,
    - section number, paragraph, sentence, word number in the sentence.

Improved representation of the inverted file

- **D1**: The GDP incrased 2 percent this economy quarter.

- **D2**: The spring economic slowdown continued to spring downwards this economy quarter.

Dictionary File		Posting File
2	1	→ D1:4
continued	1	→ D2:5
downwards	1	→ D2:8
economic	1	→ D2:3
economy	2	→ D1:7, D2:10
GDP	1	→ D1:2
increased	1	→ D1:3
percent	2	→ D1:5
quarter	2	→ D1:8, D2:11
slowdown	1	→ D2:4
spring	2	→ D2:2

Matching

Information Retrieval models

- An IR system is based on a **mathematical model** that provides a formal description:
  - of the document,
  - of the query,
  - of how to compare the query and the document representations to estimate the relevance of documents to the query.

- It should be noted that the use of the **same formal framework** to represent both documents and queries guarantees a correct matching.

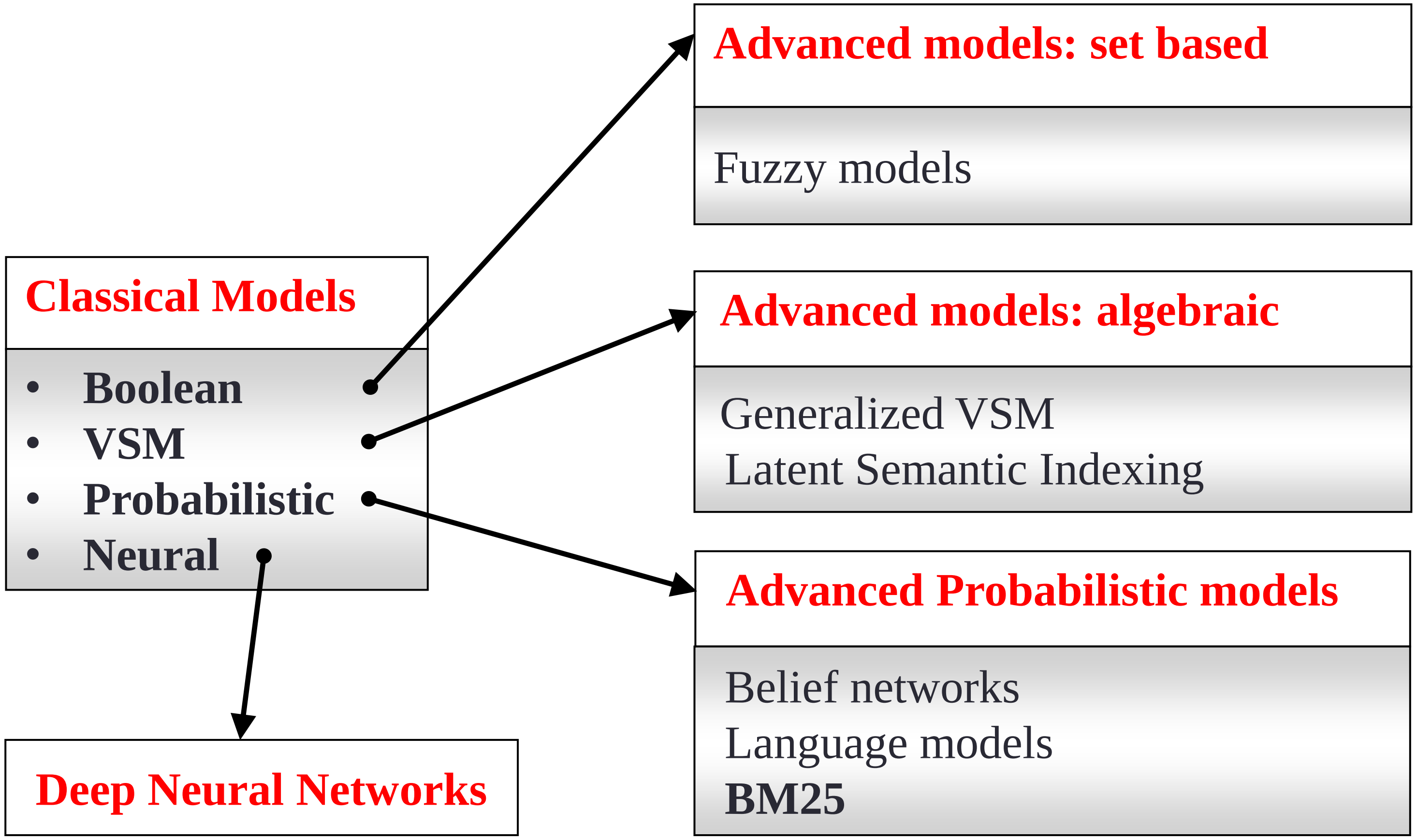
- IMPORTANT: An IR system simplifies the complexity of the retrieval activity → the results produced are not «perfect» (**estimate of relevance**).

Information Retrieval models

PURPOSE: Retrieving the relevant documents for the user: the relevance of a document is relative to the formulated query (i.e., **topical relevance**).

- **Exact comparison**: “binary” notion of relevance
  - Relevant / Not relevant.
- **Partial comparison**: “gradual” notion of relevance:
  - Idea: comparison between the document and the query that tolerates mismatches (e.g., **similarity** of the query to the document).

Information Retrieval models



The Boolean model

- It is based on **Set Theory**
- A document is formally represented as a set of index terms → **binary weights** associated with index terms.
- A query is formally represented as a **Boolean expression on terms** (use of Boolean operators AND, OR and NOT).
- The matching mechanism applies **set operations**.
- Relevance is modeled as a **binary property** of documents (**Retrieval Status Value** either 0 or 1).

Boolean language

- **Boolean query**: two or more search terms (keywords) connected by Boolean connectives:

  AND **OR**

- Terms can be negated using the **NOT** operator.

- **EXAMPLES:**
  - abacus AND actor
  - abacus OR actor
  - (abacus AND actor) OR (abacus AND atoll)
  - NOT actor

Indexing

- Inverted file (with simple posting lists).

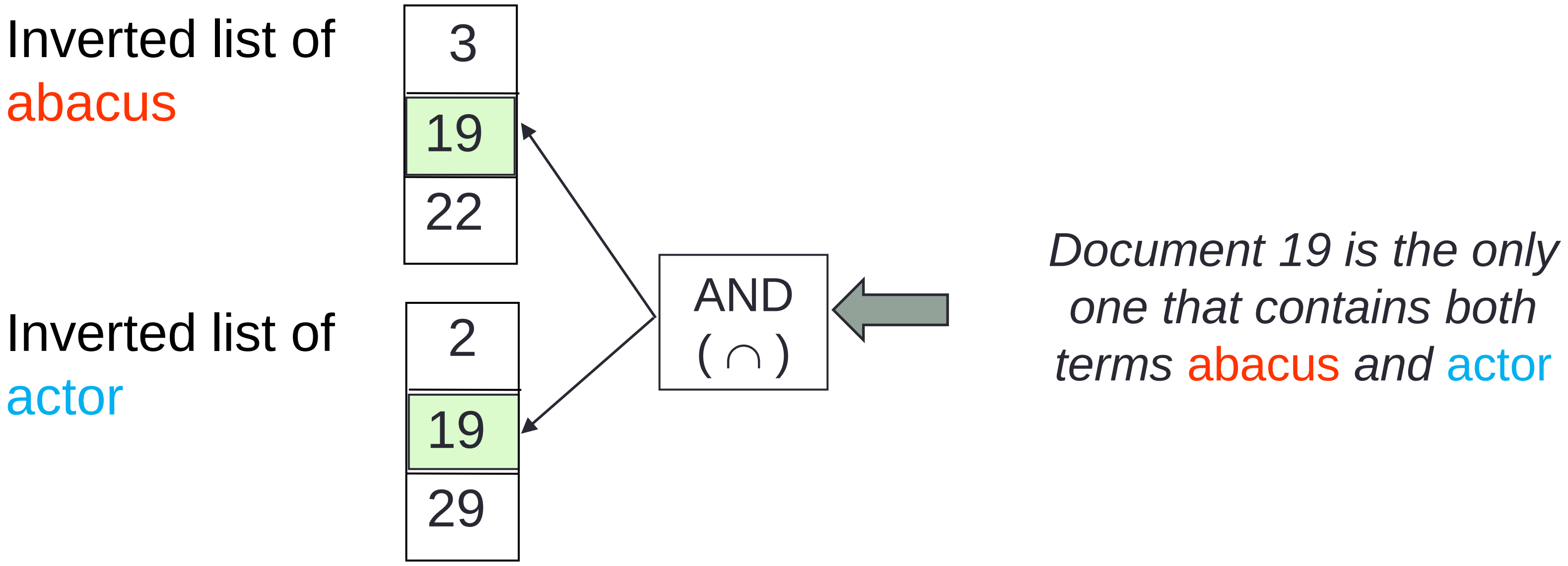
Dictionary	Posting file
term	*docID*
abacus	3 19 22
actor	2 19 29
aspen	5
atoll	11 34

$\}$ Posting list of abacus

Words are index terms selected in the indexing phase

Evaluation of Boolean queries

- The dictionary file is accessed. The posting list corresponding to the term is retrieved.
- Esempio: abacus AND actor


Inverted list of abacus
3
19
22
Inverted list of actor
2
19
29
AND
(∩)
Document 19 is the only one that contains both terms abacus and actor

Evaluation order

- The evaluation order of Boolean queries is important and must be specified. For example:
  - posting energy = d_1, d_3, d_5, d_7
  - posting nuclear = d_2, d_3, d_4, d_5, d_6
  - posting solar = d_4, d_6, d_8

- q = energy AND nuclear OR solar

- **From the left:**
  - $\{d_3, d_5\} \cup \{d_4, d_6, d_8\} = \{d_3, d_5, d_4, d_6, d_8\}$

- **From the right:**
  - $\{d_2, d_3, d_4, d_5, d_6, d_8\} \cap \{d_1, d_3, d_5, d_7\} = \{d_3, d_5\}$

Definition of priority

Operators	Priority
ADJ, NEAR	highest
AND, NOT	⇕
OR	lowest

- Example
  - *a* AND *b* OR *c* AND *b*

- It is evaluated as
  - (*a* AND *b*) OR (*c* AND *b*)

Limitation of the Boolean model

- The matching is based on a **binary decision criterion** (exact comparison).

- It is not able to produce a **ranking** of the results.
  - ***q*** **= a AND b AND c AND d AND e**

    both d_1={a} and d_2={a, b, c, d} are excluded
  - ***q*** **= a OR b OR c OR d OR e**

    the rank of d_1={a} is equal to the rank of d_2={a,b,c,d,e}

The Vector Space Model (VSM)

- It is based on **linear algebra**:
  - both documents and queries are represented in an ***n*-dimensional vector space**, where n is the number of index terms in the considered collection.
- **Relevance** is modeled as a **gradual notion** and it is proportional to the **proximity** of the vector that identifies a document to the vector that identifies the query:
  - proximity = similarity of vectors;
  - proximity ≈ inverse of distance.
- **Term independence**: terms that appear simultaneously in a document are are assumed not correlated.

The Vector Space Model (VSM)

- A **document** is represented as a vector, where weights can be either binary (0 and 1) or positive numbers:

$$\overrightarrow{d_j} = (w_{1j}, w_{2j}, \dots, w_{Nj})$$

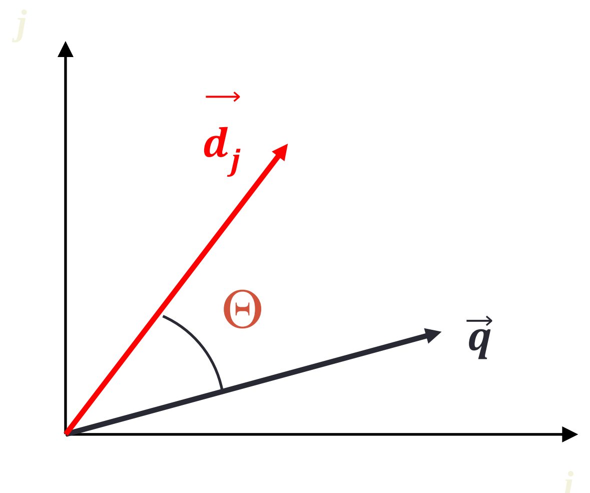
- A **query** is also represented as a vector:

$$\vec{q} = (w_{1q}, w_{2q}, \dots, w_{Nq})$$

The Cosine similarity

$$sim(\overrightarrow{d_j}, \vec{q}) = \frac{\overrightarrow{d_j} \bullet \vec{q}}{\|\overrightarrow{d_j}\| \cdot \|\vec{q}\|} = \frac{\sum_{i=1}^{n} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{n} (w_{ij})^2} \sqrt{\sum_{i=1}^{n} (w_{iq})^2}}$$

- if $w_{ij} > 0$ and $w_{iq} > 0 \rightarrow 0 \leq sim(\overrightarrow{d_j}, \vec{q}) \leq 1$
- A document is retrieved even if it does not satisfy completely the query.
- $\|\overrightarrow{d_j}\|, \|\vec{q}\|$ normalization factors.



WEB SEARCH

The Web graph

- We can think at the World Wide Web as a graph, in which:
  - The nodes are the URLs.
  - There is an edge between node x and node y when the page corresponding to URL x contains a link to the URL y.

- This graph is called the **Web graph**. Obviously, it is a dynamic graph that changes frequently.

Relevant information on the Web

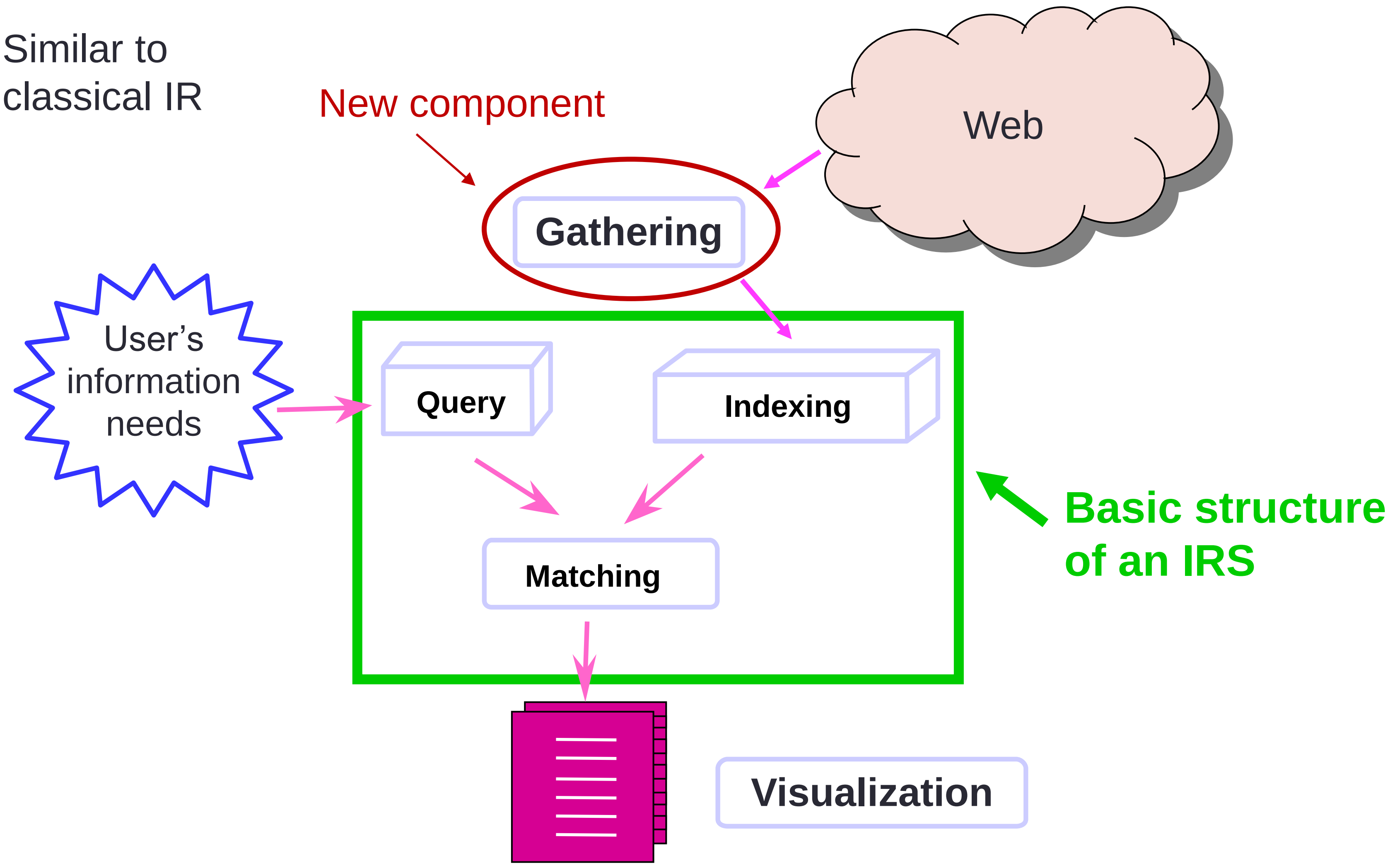
- Various **search methods**:
  - Direct search given URL.
  - Search by link (browsing).
  - Use of Web services for search:
    - Search engines that index a portion of Web documents and allow the user to formulate queries and retrieve relevant webpage addresses (Google, Bing, ...).
    - Web portals that in addition to providing a search engine, classify Web documents by directory and provide an interface for browsing the document catalog.

- **Systems supporting electronic commerce** (Information Filtering Systems: Recommender Systems).

Web search

- Terminology and definitions:
  - A Web page corresponds to a document in traditional IR.
  - Web pages differ in size, structure, type of files that are contained (text, graphics, sounds, images, video, format (HTML, GIF, JPEG, ASCII, PDF, etc.).
  - Web search considers as a collection of documents the part of the Web that is publicly indexable and excludes pages that require authorizations, dynamic pages, etc.

Structure of a Web Search Engine



Deep Web

- **Deep Web** (or Hidden Web) indicates the parts of the World Wide Web whose contents are not indexed by standard Web search engines.

- Some examples:
  - Dynamic pages, which are dynamically generated by specific applications, then discarder. E.g., flight reservation pages.
  - Limited access content pages, where the access is limited in a technical way (e.g., using CAPTCHAs).
  - Private Web pages, which require registration and login (password-protected resources).
  - Unlinked content, i.e., pages which are not linked to by other pages.

Document gathering

The crawling process

- The search engine is equipped with a software agent (information agent), namely

  crawler, (spider, worm, robot, Web search agent)

  that browses the Web to send new or updated pages to a server that indexes them.

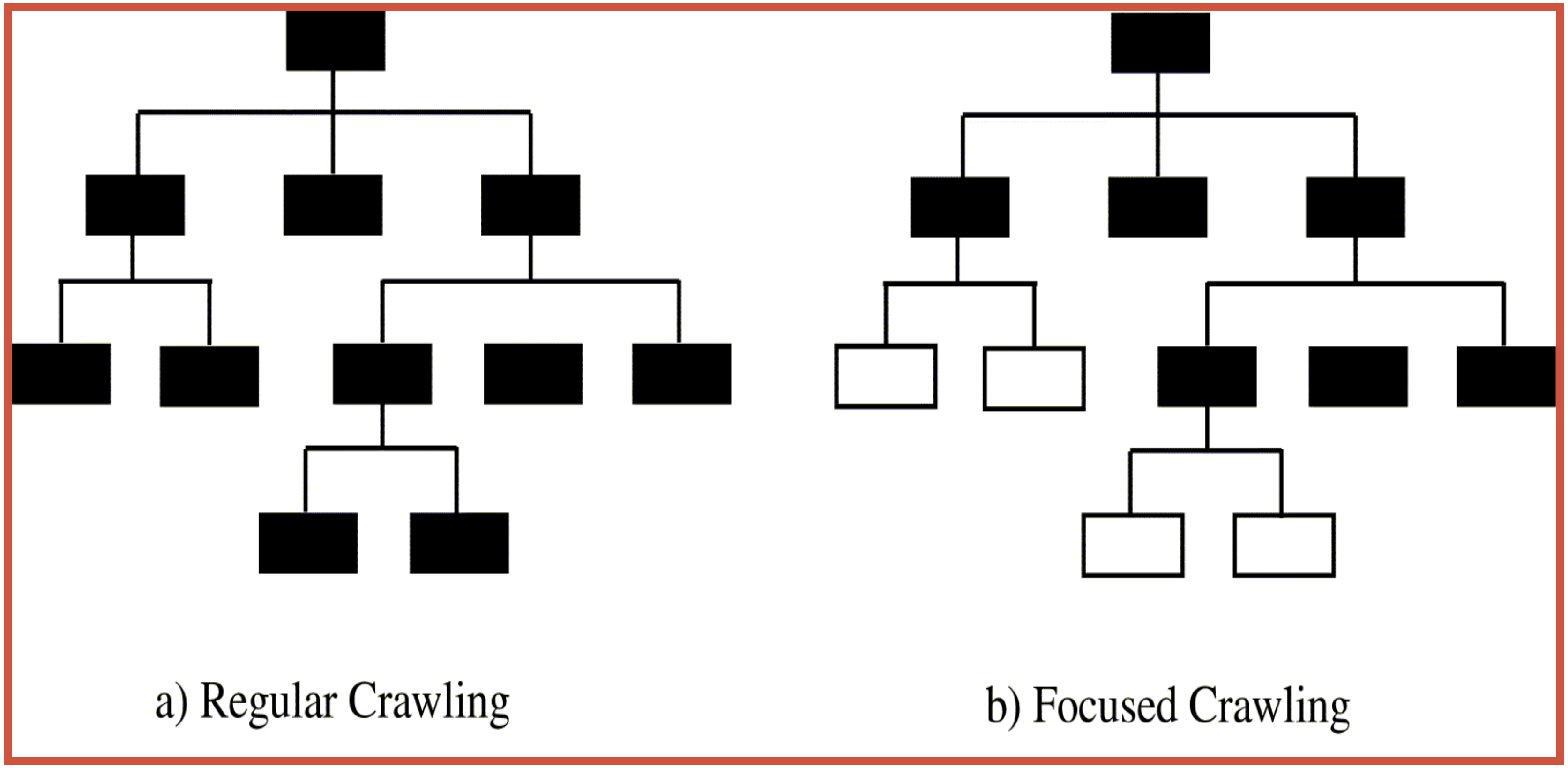
- The crawler surfs the Web using known URLs as starting points (well-known interesting points of access) and then visits other Web pages through links that go from one page to another.
  - The collection starts from one (or more) pages, called seed(s) of collection (seed set).

The crawling process

1. Initialize a page queue with some known URLs (popular or sent by users)
2. Select a URL address from the queue
3. Select the page
4. Search for other URLs in the Web page
   - For example, <a href=../pubs/index.html >publications</a>
5. Discard the URLs that:
   - Can not be analyzed, e.g., .exe
   - Have already been visited
6. Add URLs to the queue
7. If the time has not expired, go back to step 2

Focused crawling

- Focused crawlers only visit pages that are considered relevant to a topic.



Link analysis

Ranking documents on the Web

- The main difference between Web search and traditional search is the fact that links pointing to a page provide a measure of its **popularity**.

- Links between pages indicate the existence of relationships between the pages.

- The 2 main criteria for the estimation of relevance are:
  - Topicality (thematic relevance): content of the pages.
    - The most common models are the Boolean model and the Vector Space Model.
  - Popularity: link analysis.
    - PageRank, HITS, which determine the popularity of Web pages (such as the citations analysis for Impact factor calculation of scientific publications).

Link analysis

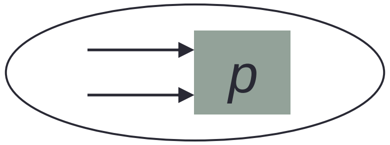
- It uses the **Web structure** to estimate page popularity.

- The popularity of a page grows as the number of its in-links increases (and may depend on the number of its out-links).

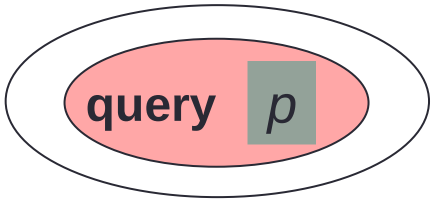
- Popular pages are more likely to contain relevant information with respect to non-popular ones.

Link analysis

- Popularity independent from the query - global analysis
  - PageRank (Google): it simulates a random walk through the Web and computes the "degree" of probability of reaching page *p (the page rank of p)*



- Query-dependent popularity - local analysis
  - HITS (*Hyperlink Induced Topic Search*): it identifies authoritative pages based on those retrieved from a query or those connected to them (to the retrieved pages)

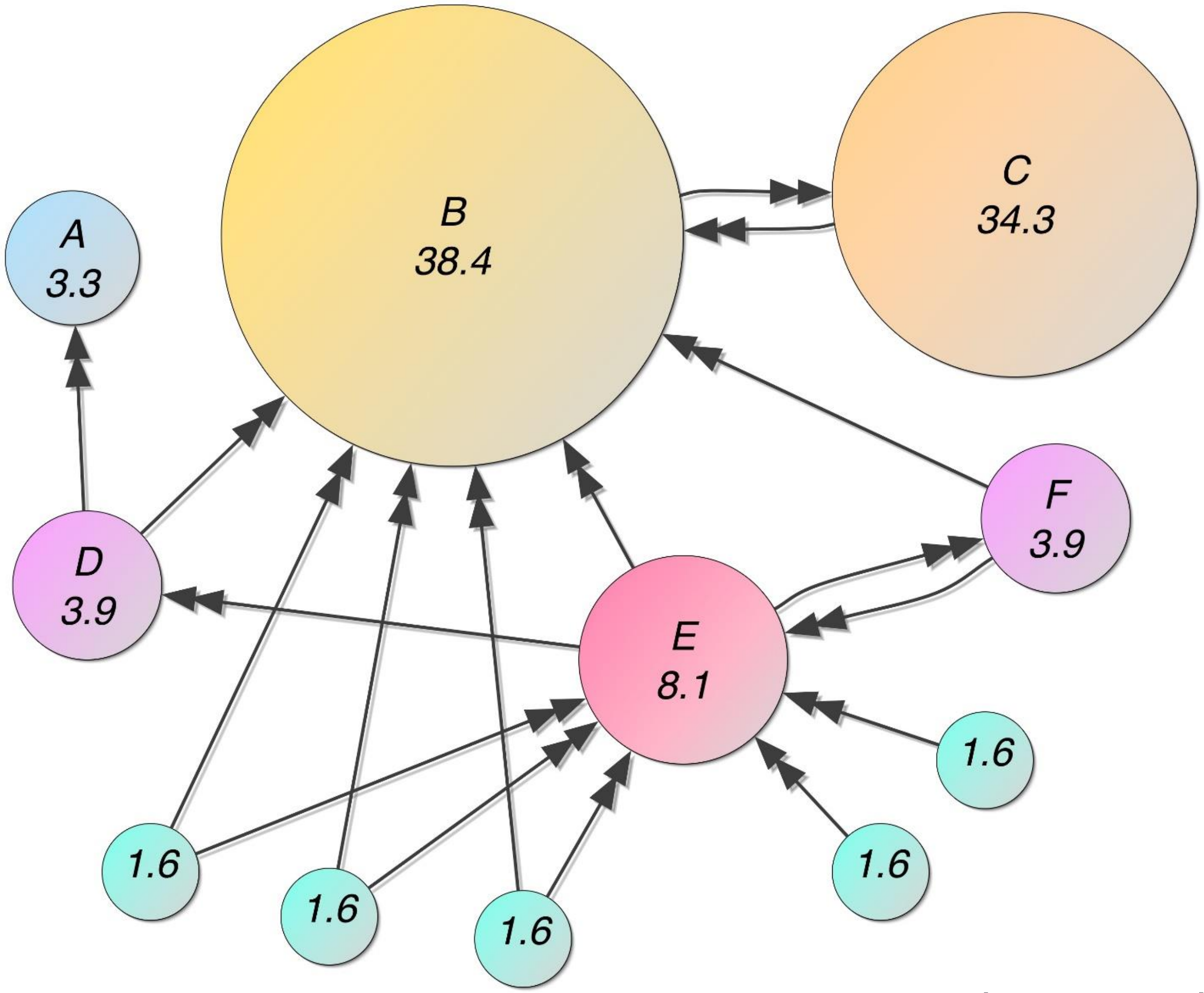


PageRank

- The **Google** algorithm.
- Basic idea: PageRank simulates a user walk on the Web.
  - A page with high PageRank value has:
    - either several in-links,
    - or a few in-links with a high PageRank value.
- Retrieval (originally) is a combination of:
  - Topicality estimate.
  - PageRank value.

(Sergey Brin, Larry Page)

PageRank



(Sergey Brin, Larry Page)

CONTEXTUAL SEARCH

Information needs and user queries

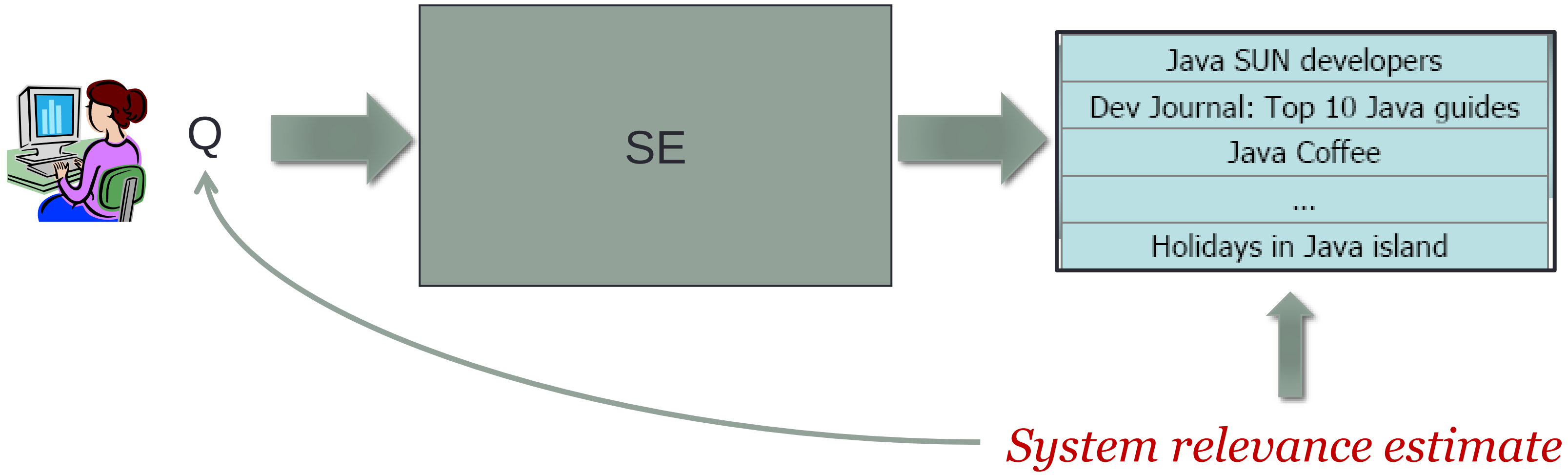
- Search Engines require users to synthesize/translate their information needs into an expression of a formal query language (generally keyword-based, with Boolean operators)

  → **query-centered approach**, where the system strongly relies on user queries as explicit signals of the user's interests.

- **Poor representation** of the information needs as:
  - Users may find it difficult to express by a few keywords a complex information need.
  - Users are encouraged to enter short queries (both by the search engine interface, and by the fact that long queries do not produce good results).
  - Users sometimes have no clear idea of what they are looking for.

Query-centered Search Engines

- The query is the unique, static manifestation of the users' needs, and the system behaves as a black box. **This model is not realistic**.



- User's relevance? **Subjective**, related to the cognitive background of the user and to other variables…

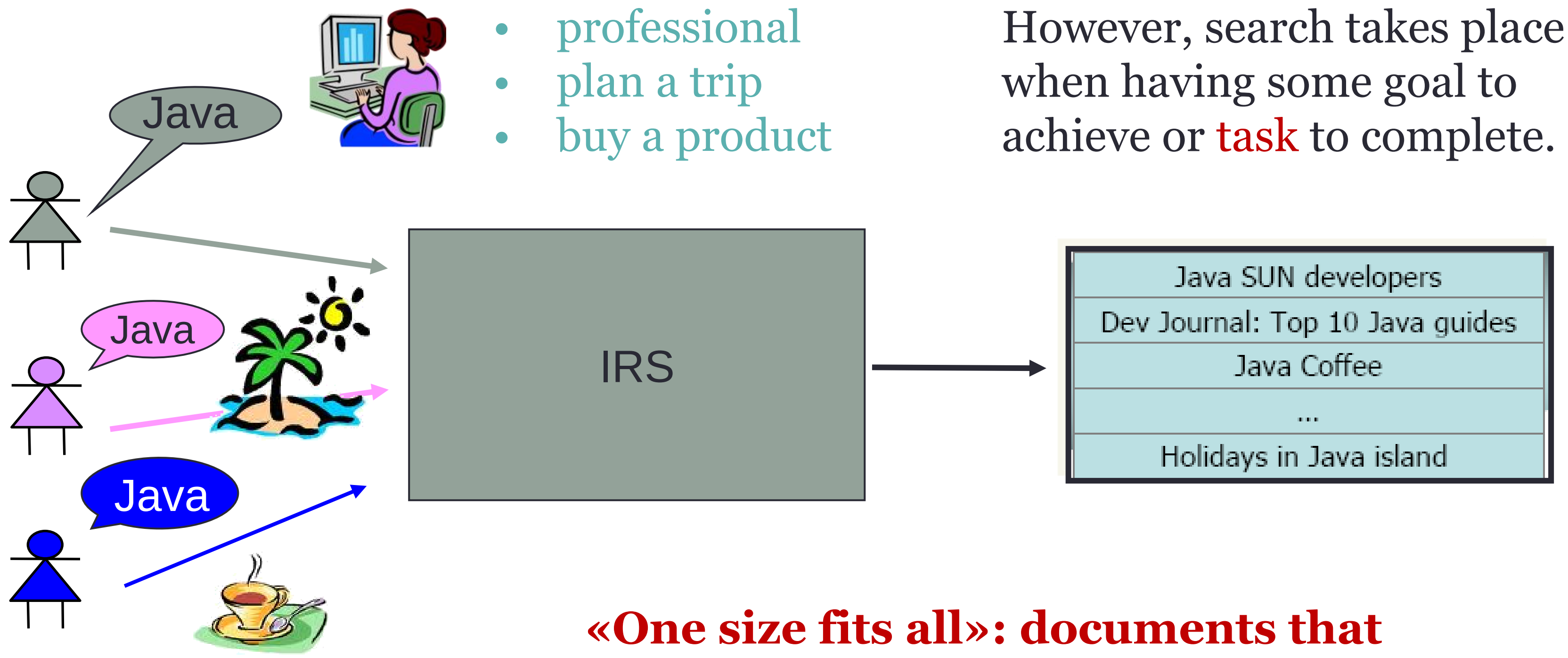
Limitations of query-centered SE

- A same query by distinct users produces the same results.

Task?
- professional
- plan a trip
- buy a product

Search Context Ignored

However, search takes place when having some goal to achieve or task to complete.

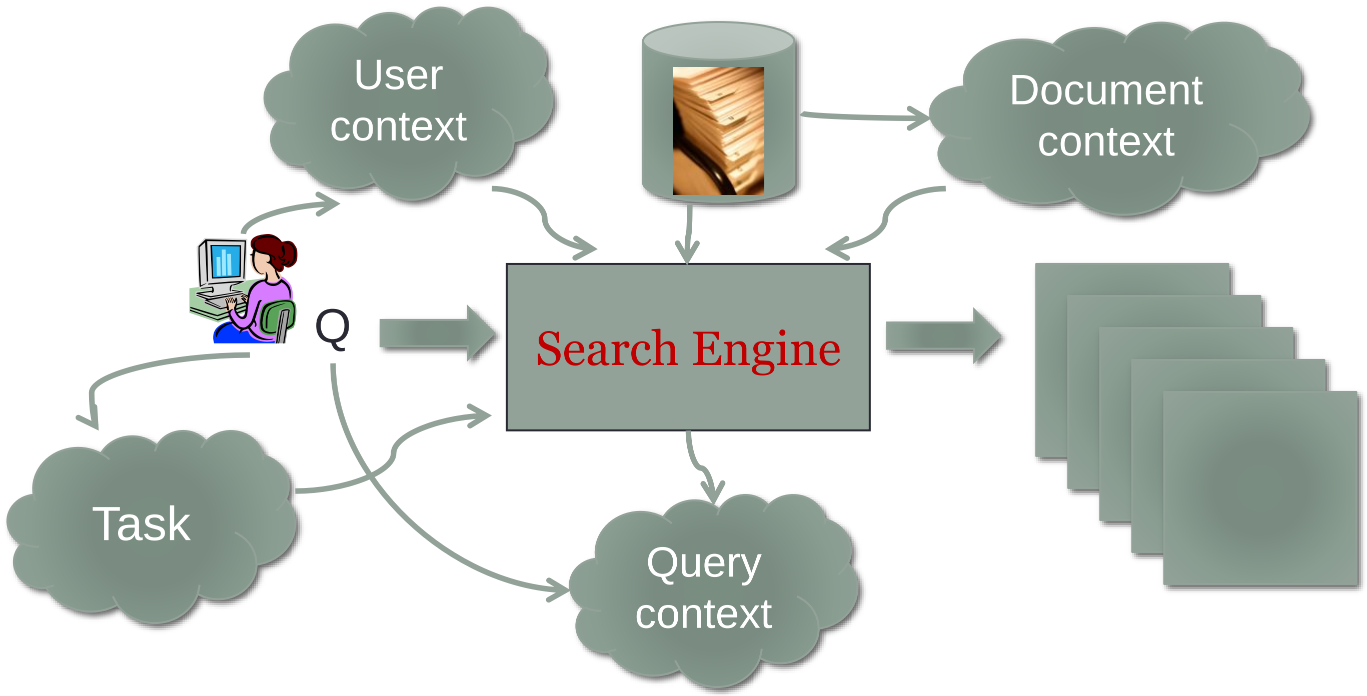


Java SUN developers
Dev Journal: Top 10 Java guides
Java Coffee
...
Holidays in Java island

«One size fits all»: documents that best fit the query are returned

Contextual search

- Aim: integrating (various kind of) **context** in the IR process
- A shift from query-centered IR to **context-aware IR**
  - Information need = query + ?
  - Relevance: beyond topical relevance to contextual (situational) relevance



Contextual search

~~Web search~~

Contextual search

Google needs to **understand** rather than **index**

From the presentation "*predicting the future of Google – Brightonseo*", 2014
https://www.slideshare.net/millerian/predicting-the-future-of-google-brightonseo

Contextual search: which information?

- **User context**
  - Age, Gender
  - Topical interests
  - Expertise
  - Social Context
- **Task (activity)**
  - E.g., Planning a travel
  - Making a survey
  - Exploratory search
- **Query context**
  - Some proposals in the literature makes use of previous queries to better understand the user needs in a query session and to personalize the search outcome (Xiang et al, Context Aware Ranking in Web Search, 2010).
- **Environment**
  - Device: PDA….
  - Geographic
  - Temporal
- **Document context**
  - Author
  - Genre
  - Tags
  - Creation date
  - Usage
  - …
- …

Personalized search

Defining personalized search

Personalized search: it produces a search outcome fitting the user context ("*user's characteristics and preferences*"). The user context is represented by a **user model** (**user profile**).

↓

Personalized IR may be considered as an « instance » of context-based IR

The user context

- **Demographic data**
  - Age, sex, education, language, country, …

- **Content-related information**
  - Explicitly defined by the user
    - List of keywords, generated documents.
    - Desktop information.
  - Generated within social activities (annotations, tags, posts, blogs, …)
  - Usage and activity based
    - Search history: past queries, visited pages.
    - Browsing history.
    - Content derived from usage of applications.
    - Interaction activities (eye gaze tracking, time spent on pages/documents, downloading, printing, …).

Short- and long-term profiles

- **Long-term profile**
  - Modeling of users' persistent interests and preferences.

- **Short-term profile**
  - Usually related to query sessions and then to specific information needs.
  - Session segmentation question: given a sequence of user queries, where to cut the session boundary (each query, several queries ...).
    - Timeout threshold.
    - Common words or edit distance between queries.
    - Adjacency of two queries in user input sequences.
    - Similarity between the top-*k* search results of two queries.

User profile representation

- **Words**

- **Concepts**
  - Use of external knowledge resources to identify concepts in the user related information (existing taxonomies or ontologies, e.g. OPD –taxonomy-, Wordnet, Yago –Yet another Great Ontology).
  - Need a collection of information that represent the user interests.
  - Need a mapping of the user information to the external resource.

- **Several formal representations** of user profiles have been proposed, among which:
  - Bag of words $\searrow$
  - Vectors $\rightarrow$ Unstructured User Models
  - Graphs
    - Hierarchy
    - Network
  - Ontologies
  - Language models, topic models, word embeddings, …

Personalization processes

1. **Pre-processing approaches**
   - Query modification/expansion.

2. **Post-processing approaches**
   - Result re-ranking.

3. **In-process approaches**
   - Personalization integrated in the ranking process.

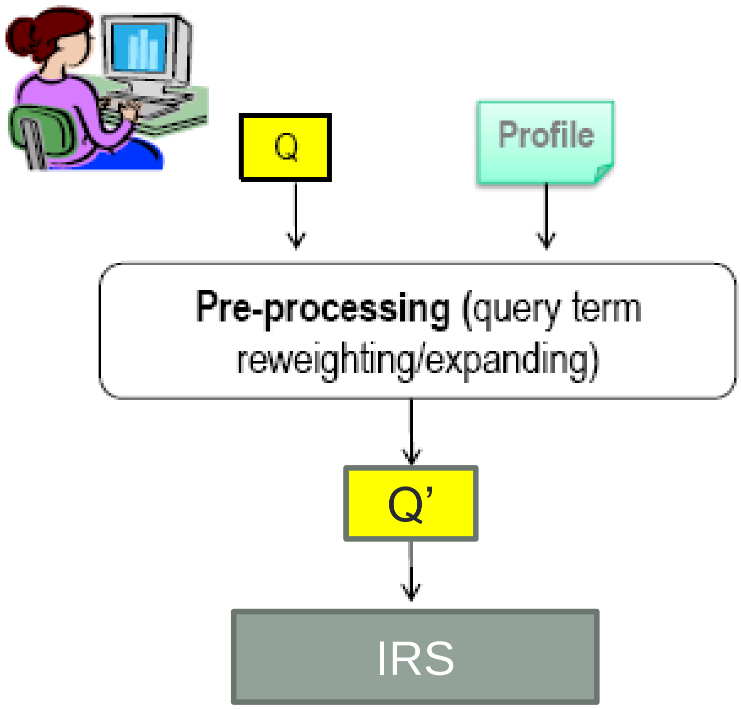
- Personalized presentation of search results.

1. Pre-processing

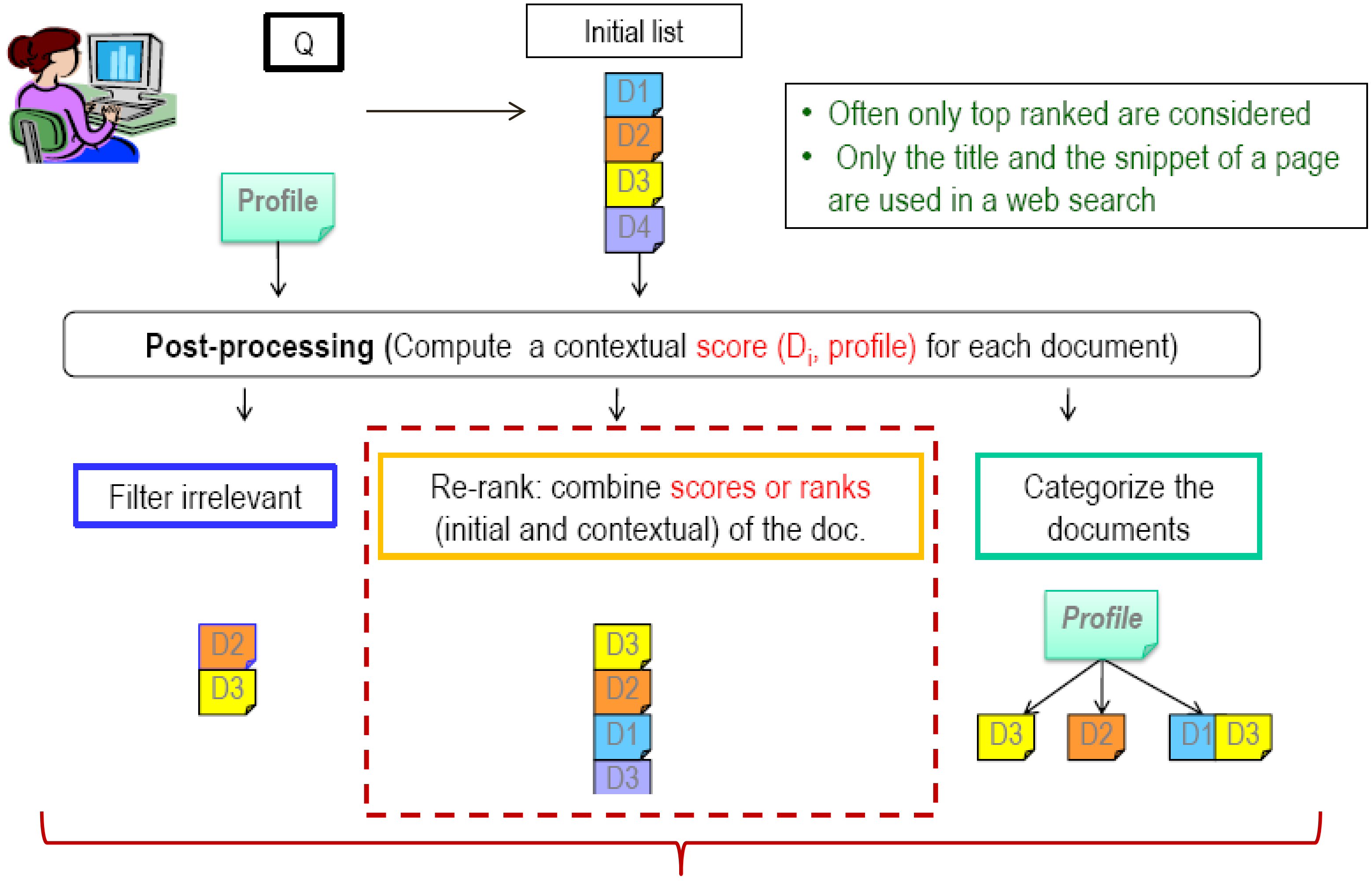
Query modification/ expansion

- The query is modified/expanded by adding and/or reweighting terms (categories) issued from the user profile.

- E.g., query expansion:
  - Query: sport
  - User profile: soccer, volleyball, basketball
  - Expanded query: sport soccer volleyball basketball



2. Post-processing


Q
Initial list
D1
D2
D3
D4
Profile
Post-processing (Compute a contextual score (D_i, profile) for each document)
Filter irrelevant
D2
D3
Re-rank: combine scores or ranks (initial and contextual) of the doc.
D3
D2
D1
D3
Categorize the documents
Profile
D3
D2
D1
D3

- Often only top ranked are considered
- Only the title and the snippet of a page are used in a web search

Possible post-processing strategies

3. In-process

Personalised retrieval processes

- Personalization is incorporated in the retrieval process.
- Ranking is a unified process where user preferences/context are employed to score the documents in a query evaluation process.