Overview of Text Data Access

CiTIUS Centro Singular de Investigación en Tecnoloxías Intelixentes

Tecnologías de Gestión de Información No Estructurada

Prof. Dr. David E. Losada

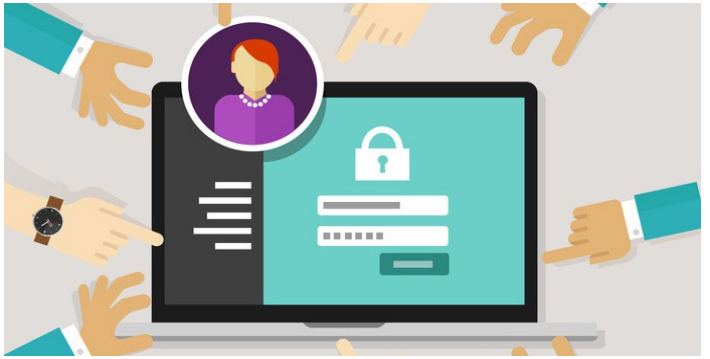
USC Universidade de Santiago de Compostela

CAMPUS VIDA

Máster Interuniversitario en Tecnologías de Análisis de Datos Masivos: Big Data

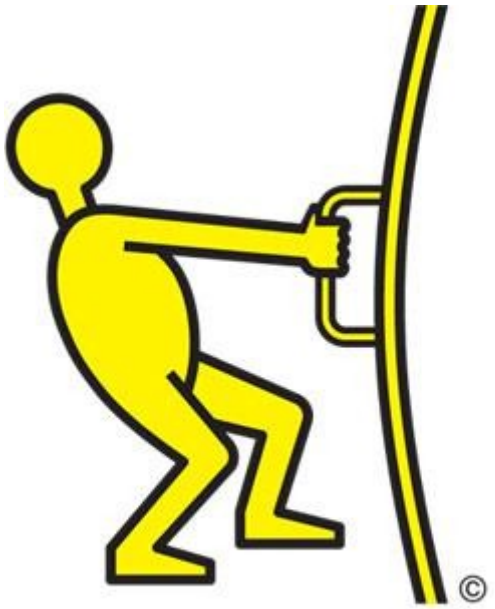
Text Data Access

foundation for **text analysis**



enables **retrieval** of the most **relevant** text data to a particular problem

enables **interpretation** of any analysis, results or discovered knowledge in appropriate **context** and provides data **provenance** (origin)



pull: the **users** take the **initiative** to fetch relevant information out from the system

push: the **system** takes the **initiative** to offer relevant information to users

selecting **relevant text data** from a large collection is the **basic task of text access**

generally based on a specification of the **information need** of an analyst (a user), and can be pull and push

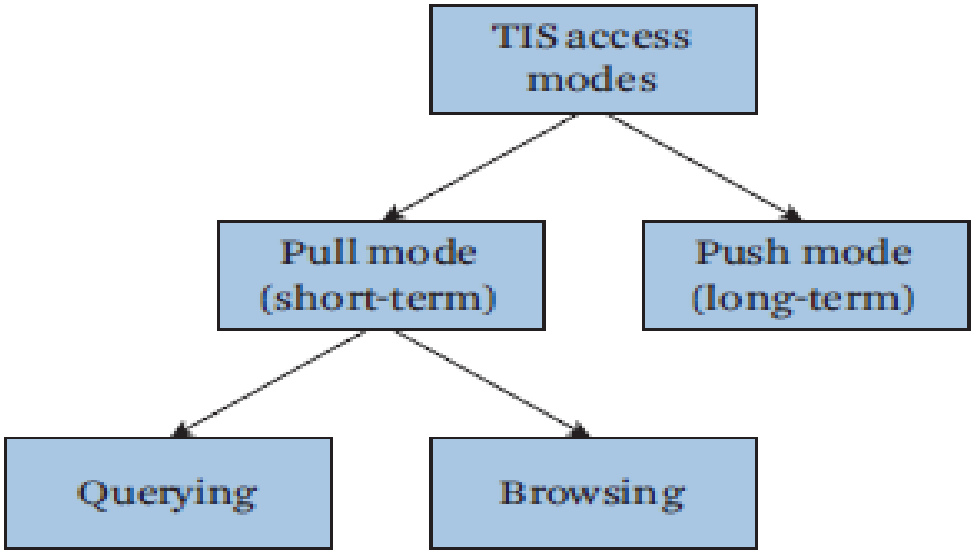


Figure 5.1 The dichotomy of text information access modes.

pull mode

the user initiates the access process to find the relevant text data, typically by using a **search engine**.

essential when a user has an **ad hoc information need**, i.e., a temporary information need that might disappear once the need is satisfied.

e.g., a user wants to buy a product and is interested in retrieving all the relevant opinions; after the user has purchased the product, the user would generally no longer need such information

pull mode

browsing

can be very useful when a user does not know how to formulate an effective query

or finds it inconvenient to enter a keyword query (e.g., through a smartphone), or simply wants to explore a topic with no fixed goal

users tend to **mix querying and browsing** (e.g., while traversing through links)

information seeking vs sightseeing

a tourist knows the exact address of an attraction (=> transport)

similar to a user who knows exactly what he is looking for

(formulates a query with the “right keywords”)

a tourist doesn’t know the exact address of an attraction (may want to go to an approximate location and then walk around to find the attraction)

similar to a user who does not have a good knowledge about the target pages, he can also use an approximate query to reach some related pages and then browse into truly relevant information.

when querying does not work well, browsing can be very useful

push mode

the system initiates the process to **recommend** a set of relevant information **items** to the user.

generally more useful to satisfy **long-standing information needs**

e.g., a researcher's research interests can be relatively stable over time.

the information **stream** (i.e., published research articles) is **dynamic**

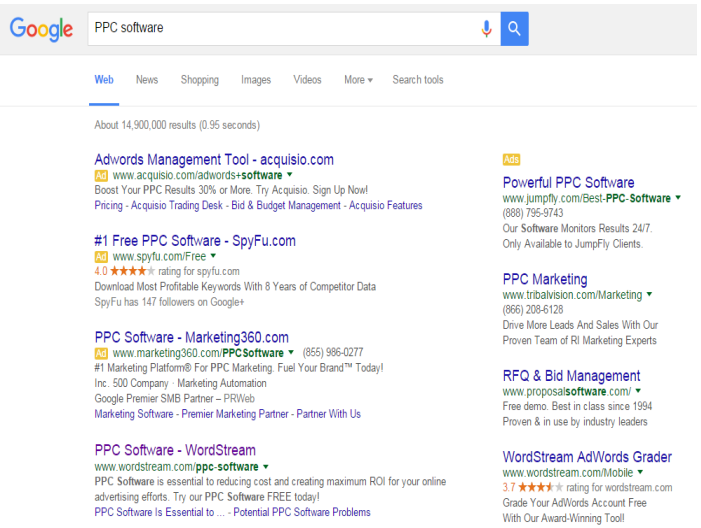
although a user can regularly search for relevant literature, it is more desirable for a recommender (also called filtering) system to monitor the dynamic information stream and "push" any relevant articles to the user based on the matching of the articles with the user's interests (e.g., in the form of an email).

push mode

producer-initiated recommendation (or selective dissemination of information, SDI).

the producer of information has an interest in disseminating the information among relevant users, and would push an information item to such users.

e.g. advertising of products on SERP



the recommendations can be delivered through email notifications or recommended through a SERP

short/long term information needs

short-term: most often associated with pull mode

temporary and usually satisfied through search or navigation in the information space

long-term: most often associated with push mode.

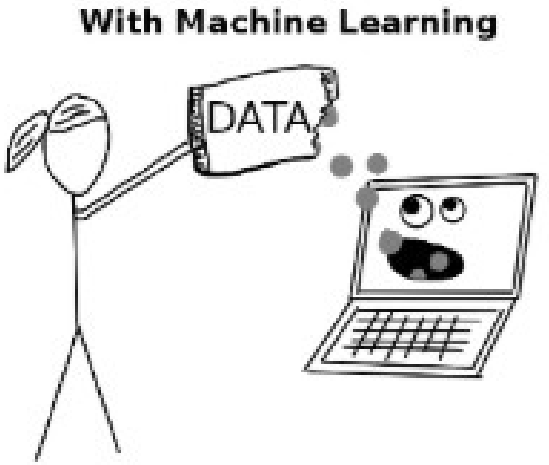
can be better satisfied through filtering or recommendation where the system would take the initiative to push the relevant information to a user.

ad-hoc retrieval extremely important

ad hoc information needs **show up far more frequently** than long-term info needs

the techniques effective for ad hoc retrieval can usually be re-used for filtering and recommendation as well.

in the case of long-term information needs, it is possible to collect and exploit **user feedback**



due to the availability of training data,

the problem of filtering/recommendation

can usually be solved by using supervised ML

in adhoc retrieval, we do not have much feedback information from a user (i.e., little training data for a particular query).

ad-hoc retrieval difficult!

multimode interactive access

push/pull modes **integrated** in the same information access environment

querying and browsing also seamlessly integrated to provide maximum flexibility

Figure 5.2 Sample interface of browsing with a topic map where browsing and querying are naturally integrated.

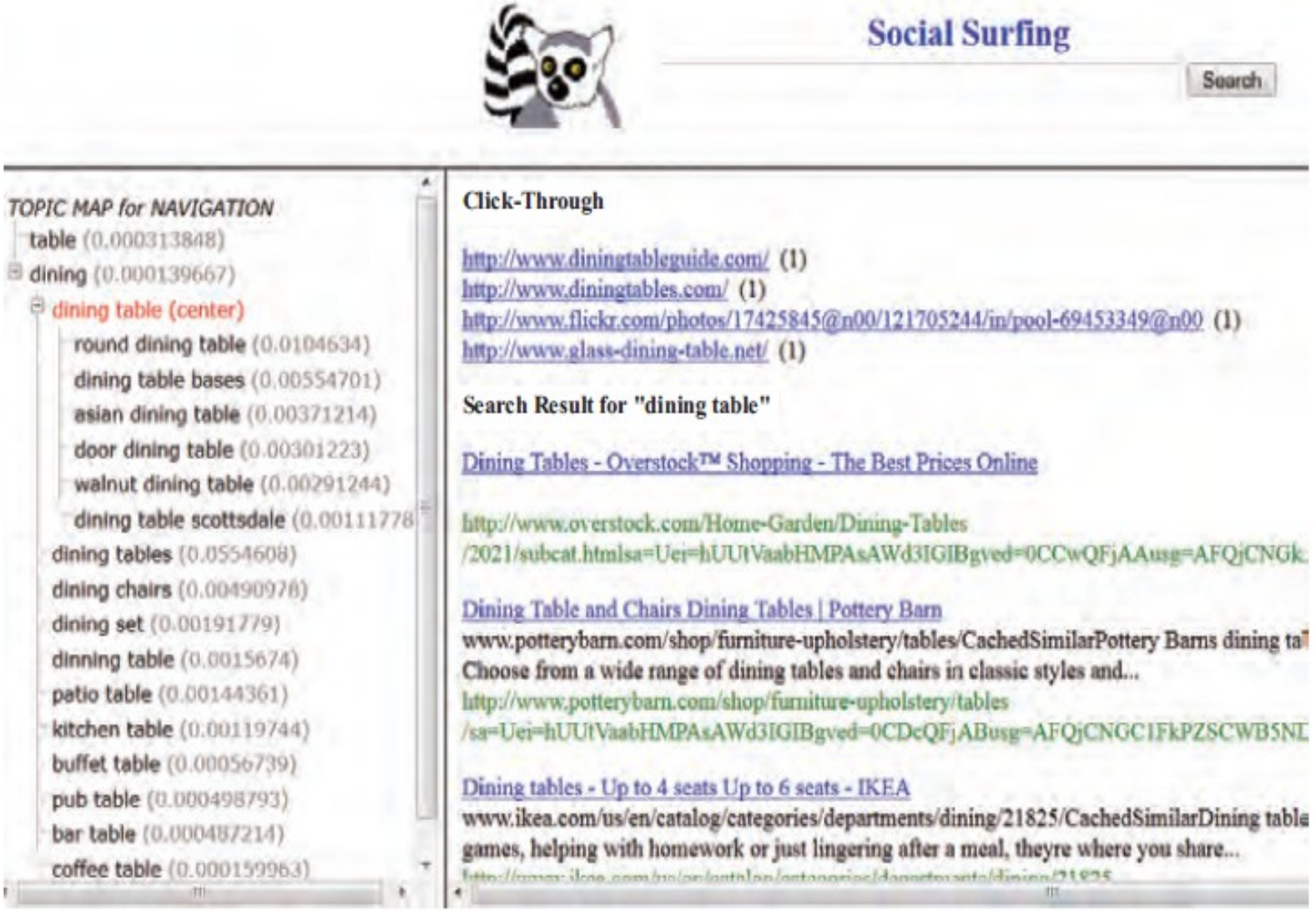


Figure 5.2 Sample interface of browsing with a topic map where browsing and querying are naturally integrated.

Querying (long-range jump)

a user submits a new query through

the search box

the search results from a search engine

will be shown in the right pane.

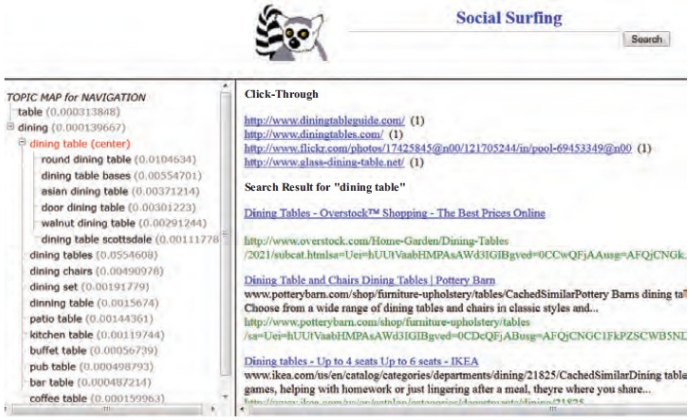


Figure 5.2 Sample interface of browsing with a topic map where browsing and querying are naturally integrated.

Navigating on the map (short-range walk)

the relevant part of a topic map is also shown on the left pane to facilitate browsing

when a user clicks on a map node, this pane will be refreshed and a local view with the clicked node as the current focus will be displayed

a user can thus zoom into a child node, zoom out to a parent node, or navigate into a horizontal neighbor node. Such a map enables the user to "walk" in the information space to browse into relevant documents without needing to reformulate queries.

Viewing a topic region

Figure 5.2 Sample interface of browsing with a topic map where browsing and querying are naturally integrated.

the user may double-click on a topic node on the map

to view the documents covered in the topic region

the search result pane would be updated with new results corresponding to the documents in the selected topic region.

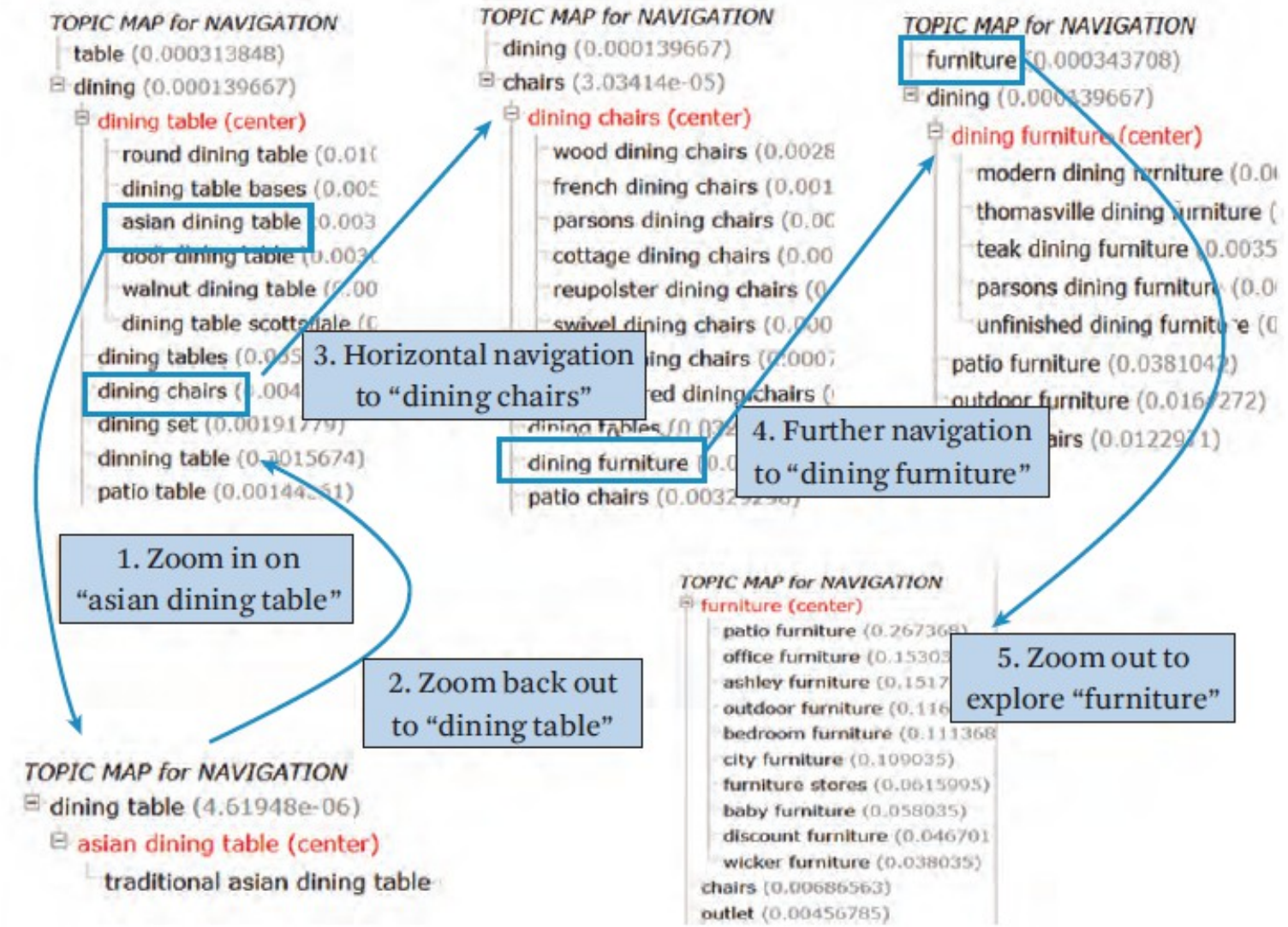


Figure 5.3 A sample trace of browsing showing how a user can navigate in the information space without querying.

text retrieval

the most important tool for supporting **text data access** is a **search engine**

SEs directly provide support for **querying** and can be easily extended to provide **recommendation/browsing**

the **techniques** used to implement an effective SE are often also useful for implementation of a **recommender** system as well as many **text analysis** functions

the problem of text retrieval (tr)

TR: to use a **query** to **find relevant documents** in a **collection** of text documents.

frequently needed task

(users often have temporary ad hoc information needs)

retrieval techniques for **non-textual data** are less mature (and tend to rely on TR to match a keyword query with companion text data)

For example, the current image search engines on the Web are essentially a TR system where each image is represented by a text document consisting of any associated text data with the image (e.g., title, caption, ….).

easy or hard

depending on **specific queries** and **specific collections**

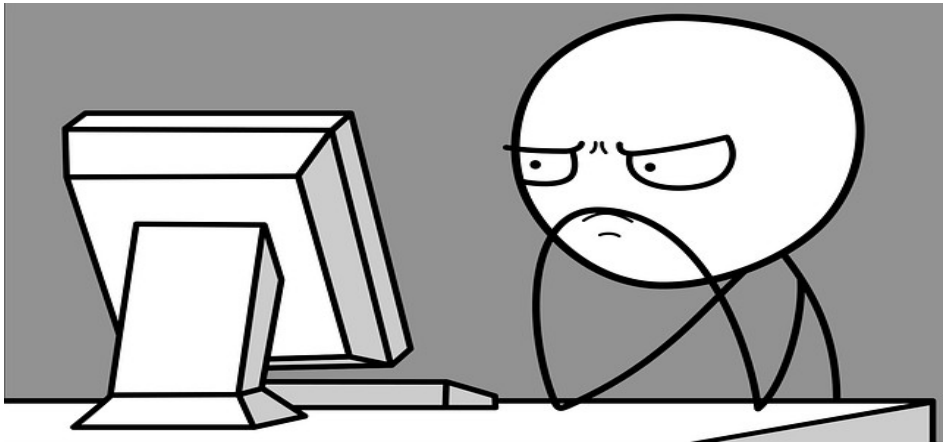
in web search, finding homepages is generally easy

but finding out people's opinions about some topic (e.g., U.S. foreign policy) is much harder

why is text retrieval difficult?

a **query** is usually quite **short** and **incomplete** (no formal language like SQL)

the **information need** may be difficult to describe precisely, especially when the user isn't familiar with the topic



why is text retrieval difficult?

precise **understanding** of the document **content** is difficult.

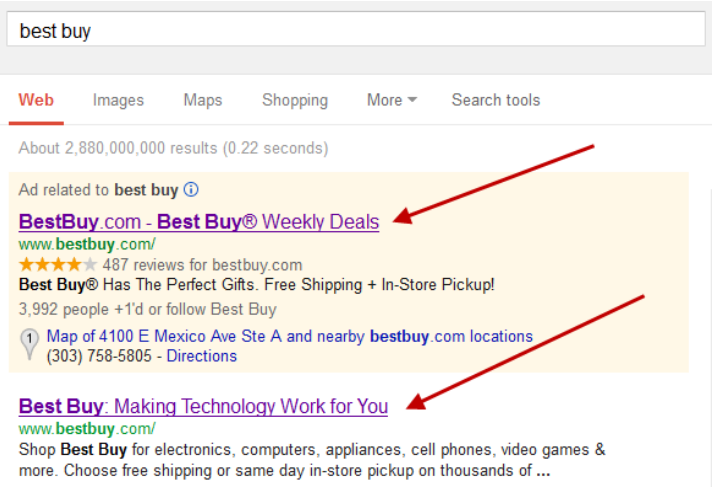
what counts as the correct answer is **subjective**

even when human experts judge the relevance of documents, they may disagree with each other

lack of clear semantic structures and difficulty in NL understanding

why is text retrieval difficult?

current SEs work very well for **navigational** queries and **simple, popular informational queries**



but in the case where a user has a **complex information need** (e.g. analyzing opinions about products to buy, or researching medical information about some symptoms), they often work **poorly**

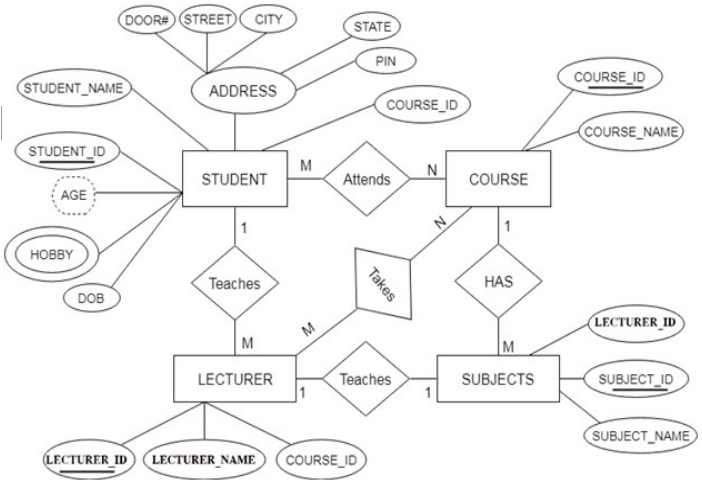
why is text retrieval difficult?

little or **no support** to help users **digest** and exploit the retrieved information

a user would still have to sift through a long list of documents and read them in detail to fully digest the knowledge buried in text data in order to perform their task at hand

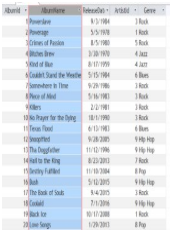
text retrieval vs data retrieval

both tasks help users **find relevant info**



the **data managed** by a search engine and a DB system are **different**

In **DBs**, the data are **structured**, each **field** has a clearly defined meaning according to a **schema**. Tables with well-specified columns.

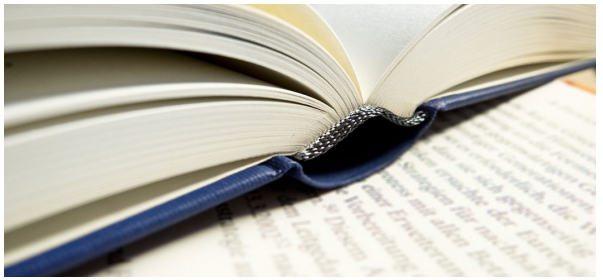


For example, in a bank DB, one field may be customer names, another may be the address, and yet another may be the balance

text retrieval vs data retrieval

the data managed by a search engine are **unstructured**

text is **difficult** for computers to **understand**

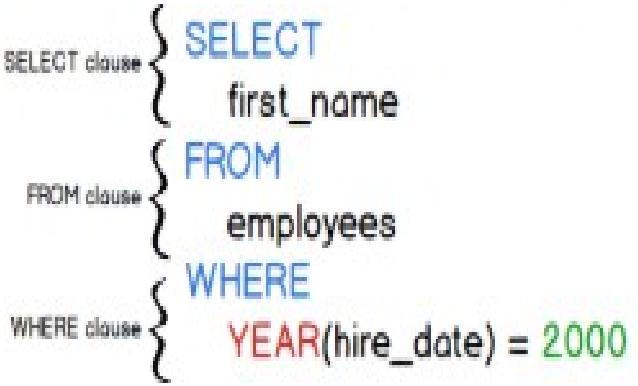


even if a sentence says a person lives in a particular address, it remains difficult for the computer to answer a query about the address of a person in response to a keyword query

text retrieval vs data retrieval

the **queries** that can be supported by the two are also **different**

A **DB query** clearly specifies the constraints on the fields of the data table, and thus the expected results are very **well specified** with **no ambiguity**



In a **search engine**, the queries are generally **keyword queries**, which are only a **vague specification** of what documents should be returned.

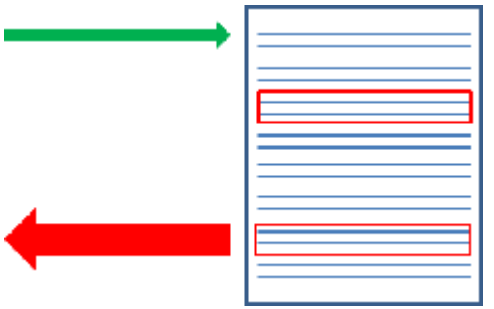
For example, in the case of searching for relevant literature to a research problem, the user is unlikely able to clearly and completely specify which documents should be returned

text retrieval vs data retrieval

DB search: we can retrieve very **specific data elements** (e.g., specific columns)

TR: we are generally only able to retrieve a set of relevant **documents**.

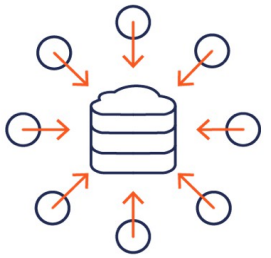
A SE can also retrieve **passages**, but it is difficult to retrieve specific entities or attribute values as we can in a database.



text retrieval vs data retrieval

In **DBs** there is no challenge in determining which data elements satisfy the user's query

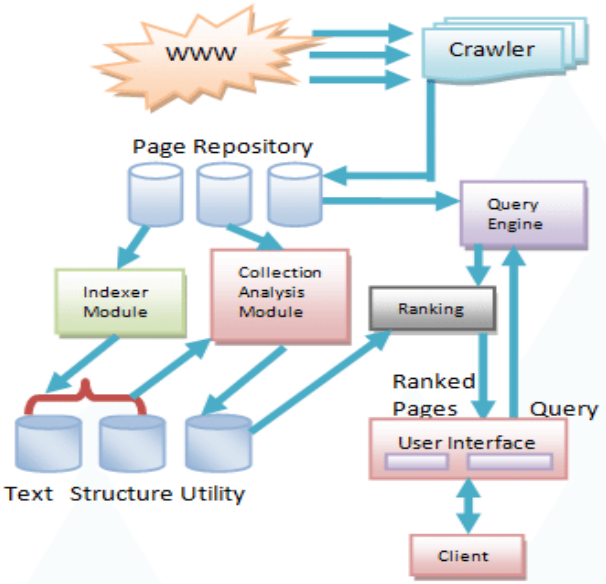
a major remaining challenge is how to find the answers **as quickly as possible**



many **queries** being **issued at the same time**

text retrieval vs data retrieval

the **efficiency** challenge also exists in a **SE**



but a more important challenge is to figure out **which documents should be returned**

text retrieval vs data retrieval

In **DBs**, it is also crucial to maintain the **integrity** of the data; e.g., to ensure no inconsistency occurs due to failures

text retrieval vs data retrieval

In TR, modeling a user's **information need** and search **tasks** is important, again due to the difficulty for a user to clearly specify information needs and the difficulty in NLP

text retrieval vs data retrieval

In TR, there is no mathematical way to prove that one answer/method is better than another

TR relies rely on **empirical evaluation**

using some **test collections** and **users**

In DB, the main issue is **efficiency**, and one can prove one algorithm is better than another by analyzing the **computational complexity** or do some simulation study.

O (Big Oh), Ω (Omega), and Θ (Theta)

text retrieval vs data retrieval

DBs. Traditional field. Widespread applications in virtually every domain with a well-established strong industry

IR community. **interdisciplinary** community involving **library** and **information science** and **computer science**

strong industry base since the early 1990s

as more and more online information is available, the search engine technologies (including components such as machine learning and natural language processing) will continue to grow

the task of text retrieval

Given a **document collection** (a set of unordered text documents), the task of TR retrieval can be defined as using a **user query** (i.e., a description of the user's **information need**) to identify a subset of documents that can **satisfy the user's information need**

the task of text retrieval

V = {w1, . . . , wN} be a **vocabulary** (set of all the words in a particular natural language)

user's query q = q1, q2, . . . , qm, a sequence of words, qi ∈ V .

doc di= di1, . . . , dim, a sequence of words, dij∈ V .

In general, a **query** is much **shorter** than a document but some retrieval tasks have large queries (e.g. patent retrieval)

the task of text retrieval

text collection C = {d1, . . . , dM} is a set of textual docs.

R(q), subset of docs, i.e., R(q) ⊂ C, which are relevant to the user's query q

the user's query is only a **"hint"**

different users may use the same query to intend to retrieve somewhat different sets of relevant documents

the task of text retrieval

it is **unrealistic** to expect a computer to return **exactly** the set R(q) (unlike the case in database search)

return an **approximation** of R(q), denoted as R'(q)

how can a computer compute R'(q)?

document selection vs. document ranking

document selection

a **binary classifier** to classify a document as either **relevant or non-relevant** with respect to a particular query.

indicator function $f(q, d) \in \{0, 1\}$

$R'(q) = \{d \mid f(q, d) = 1, d \in C\}$.

the system estimates the "**absolute relevance**"

document ranking

rank documents and **let the user decide a cutoff**

a **ranking function f (q, d)** $\in$ **R**

rank all the documents in descending values of this ranking function

R'(q) is defined **partly by the system and partly by the user**

the user implicitly chooses a score threshold θ based on the rank position where he stopped

$R'(q) = \{d \mid f(q, d) \geq \theta\}$

the system only needs to estimate the "**relative relevance**" of documents

document ranking

estimation of **relative relevance** is intuitively **easier** than that of absolute relevance

ranking is generally preferred to document selection

difficulty for a user to prescribe the exact criteria for selecting relevant documents, **the binary classifier is unlikely accurate**

query over-constrained, there may be **no relevant documents** matching all the query words, so forcing a binary decision may result in **no delivery of any search result**

query under-constrained (too general),
too many documents matching the query,
resulting in **over-delivery**

even if the classifier can be accurate, a user would still benefit from **prioritization** of the matched relevant documents for examination.

some relevant docs are more useful than others (**relevance degrees**)

probability ranking principle (PRP)

(Robertson 1997) the strategy of **ranking** shown to be **optimal theoretically** under two assumptions

a ranked list of documents in descending order of predicted relevance is the optimal strategy under the following two assumptions:

1. The utility of a document to a user is independent of the utility of any other document.

2. A user will browse the results sequentially.