

# INFORMATION RETRIEVAL-

## *INTRODUZIONE AL CORSO -*

---

Corsi di Laurea in Informatica  
Università di Roma, Tor Vergata  
(a.a. 2024-2025)

Danilo Croce

# Overview

- Information Retrieval: Motivazioni del Corso e prospettive
- Modalità di erogazione del Corso
- Forma e struttura delle prove d'esame
- Testi

# How many sites in the Web?

Year (June)	Websites	Change	Internet Users	Users per Website	Websites launched
2018	1,630,322,579	-8%			
2017	1,766,926,408	69%			
2016	1,045,534,808	21%			
2015	863,105,652	-11%	3,185,996,155*	3.7	
2014	800,000,150	11%	2,000,000,000	2.5	

*internet live stats*

live

1 second

watch

trends & more

Get our Counters!

Home > Trends and More > Total Number of Websites

## Total number of Websites

**1,987,819,880**

Websites online right now

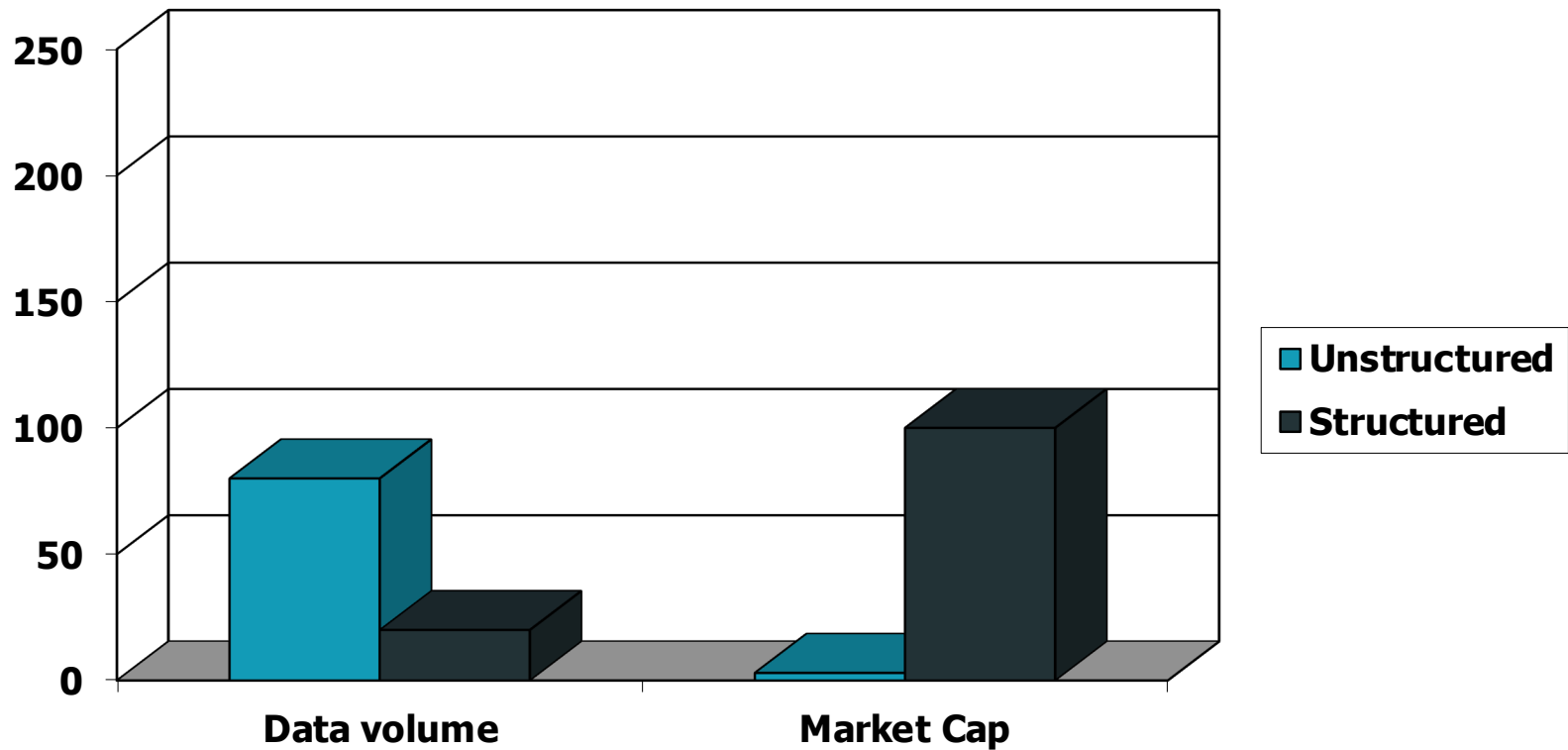
... righth now is  
01/October/2022

<https://www.internetlivestats.com/total-number-of-websites/>

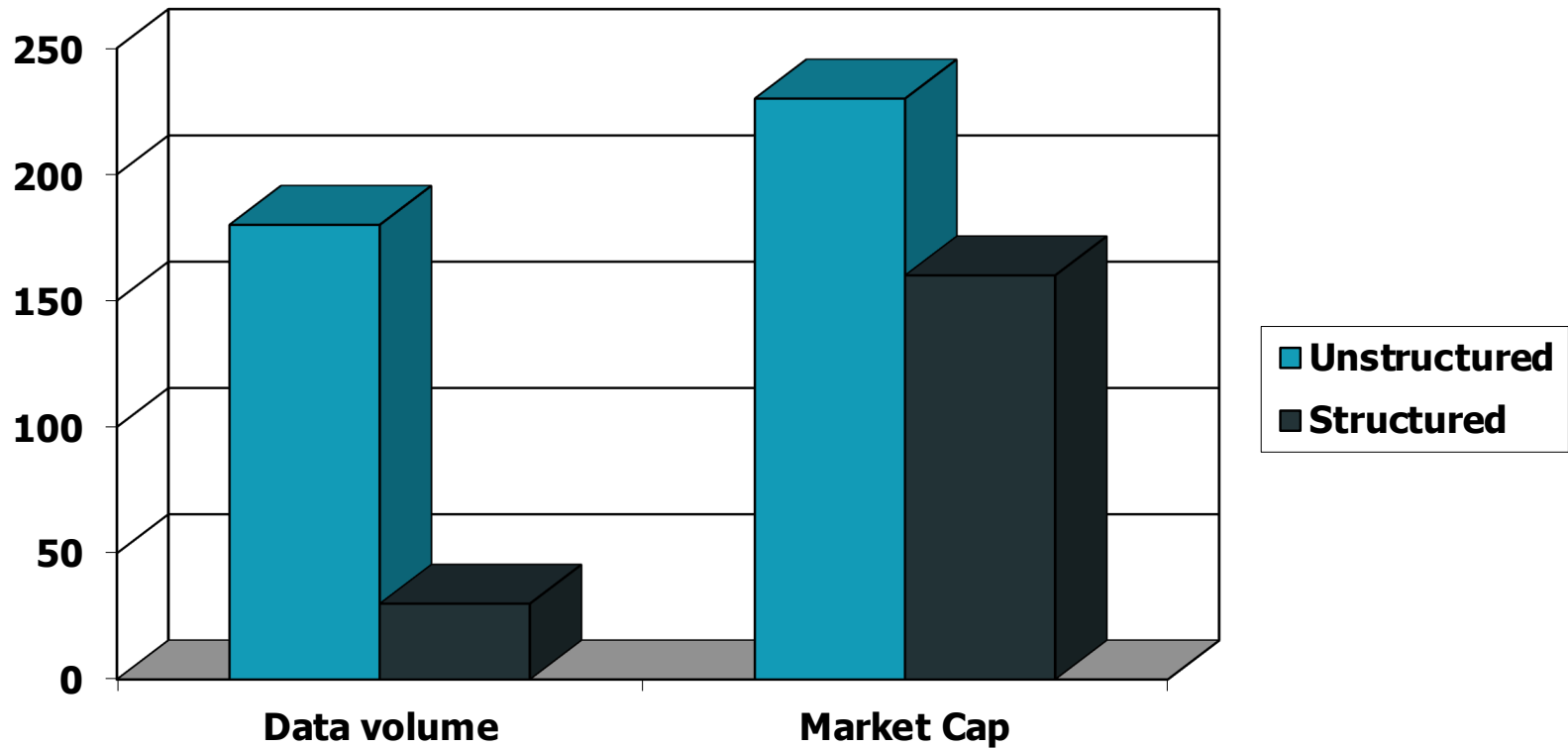
# Information Retrieval

- 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).
- These days we frequently think first of web search, but there are many other cases:
  - E-mail search
  - Searching your laptop
  - Corporate knowledge bases
  - Legal information retrieval

# Unstructured (text) vs. structured (database) data in the mid-nineties



# Unstructured (text) vs. structured (database) data today



# Basic assumptions of Information Retrieval

- **Collection:** A set of documents
  - Assume it is a static collection for the moment
- **Goal:** Retrieve documents with information that is **relevant** to the user's **information need** and helps the user complete a **task**

# Information Retrieval VS Databases

- A research field traditionally separate from Databases
  - Goes back to IBM, Rand and Lockheed in the 50's
  - G. Salton at Cornell in the 60's
  - Lots of research since then
- DB & IR Products traditionally separate
  - Originally, document management systems for libraries, government, law, etc.
  - Gained prominence in recent years due to web search



# IR vs. DBMS: some differences

- Seem like very different beasts:

IR	DBMS
Imprecise Semantics	Precise Semantics
Keyword search	SQL
Unstructured data format	Structured data
Read-Mostly. Add docs occasionally	Expect reasonable number of updates
top $k$ results	Generate all answers

- Both support queries over large datasets, use indexing.
  - In practice, you currently have to choose between the two. Not pleasant!

# IR's "Bag of Words" Model

- Typical IR data model:
  - Each document is just a bag (multiset) of words ("terms")
  - Bag models a doc just like a BBox models a spatial object.
- Detail 1: "Stop Words"
  - Certain words are considered irrelevant and not placed in the bag
  - e.g., "the"
  - e.g., HTML tags like <H1> [not always a good idea!]
- Detail 2: "Stemming" and other content analysis
  - Using language-specific rules, convert words to their basic form
  - e.g., "surfing", "surfed" --> "surf"

# Boolean Search in SQL

- Really only one SQL query in Boolean Search IR:
  - Single-table selects, UNION, INTERSECT, EXCEPT
- relevance () is the “secret sauce” in the search engines:
  - Combos of statistics, linguistics, and graph theory tricks!  
[computing reputation of pages, hubs and authorities on topics, etc.]
  - Unfortunately, not easy to compute this efficiently using typical DBMS implementation.

# Computing relevance 1/2

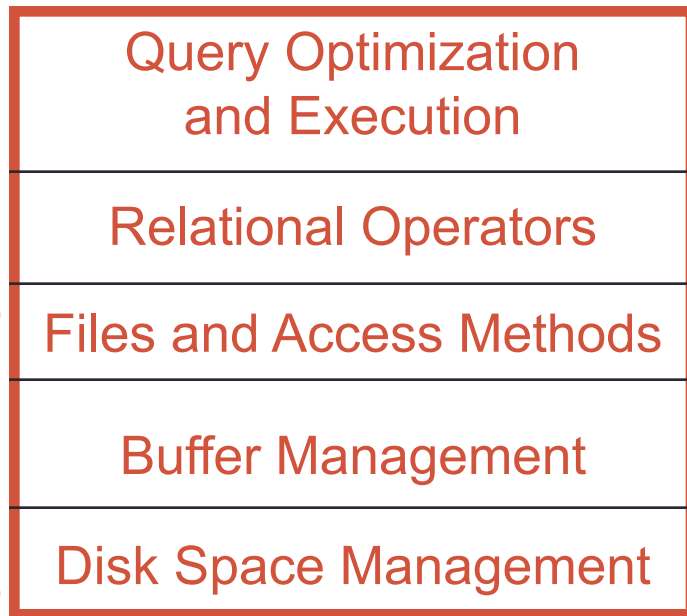
- Relevance calculation involves how often search terms appear in doc, and how often they appear in collection:
  - More search terms found in a doc is more relevant
  - Greater importance attached to finding *rare* terms (i.e., search terms, rare in the collection, but appear in this doc.).

# Computing relevance 2/2

- Doing this efficiently in current SQL engines is not easy:
  - “Relevance of a doc wrt a search term” is a function that is called once per doc the term appears in (docs found via inv. index):
    - For efficient computation, for each term, we can store the # times it appears in each doc, as well as the # docs it appears in.
    - Must also sort retrieved docs by their relevance value.
    - Also, think about Boolean operators (if the search has multiple terms) and how they affect the relevance computation!

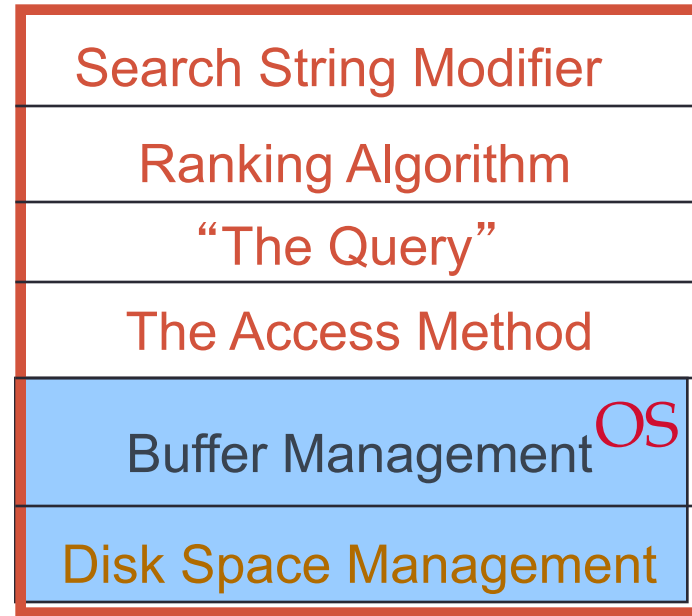
# DBMS vs. Search Engine Architecture

## DBMS



Concurrency  
and  
Recovery  
Needed

## Search Engine



} Simple  
DBMS

# IR vs. DBMS Revisited

- Semantic Guarantees
  - DBMS guarantees transactional semantics
    - If inserting Xact commits, a later query *will* see the update
    - Handles multiple concurrent updates correctly
  - IR systems do not do this; nobody notices!
    - Postpone insertions until convenient
    - No model of correct concurrency
- Data Modeling & Query Complexity
  - DBMS supports any schema & queries
    - Requires you to define schema
    - Complex query language hard to learn
  - IR supports only one schema & query
    - No schema design required (unstructured text)
    - Trivial-to-learn query language

# IR vs. DBMS, Contd.

- Performance goals
  - DBMS supports general SELECT plus arbitrarily complex queries
    - Plus mix of INSERT, UPDATE, DELETE
    - General purpose engine must always perform “well”
  - IR systems expect only one stylized SELECT
    - Plus delayed INSERT, unusual DELETE, no UPDATE.
    - Special purpose, must run super-fast on “The Query”
    - Users rarely look at the full answer in Boolean Search



# Lots More in IR ...

- How to “rank” the output? I.e., how to compute relevance of each result item w.r.t. the query?
  - Doing this well / efficiently is hard!
- Other ways to help users paw through the output?
  - Document “clustering”, document visualization
- How to take advantage of hyperlinks?
  - Really cute tricks here! (visibility, authority, page rank, etc.)
- How to use compression for better I/O performance?
  - E.g., making RID lists smaller
  - Try to make things fit in RAM!
- How to deal with synonyms, misspelling, abbreviations?
- How to write a good web crawler?

# Obbiettivi del Corso

Il corso si propone di introdurre lo studente agli scopi, alle principali problematiche e ai principali modelli dell'Information Retrieval

## Argomenti

- Introduzione al problema dell'Information Retrieval
- Definizione della nozione di Inverted Indices
- Costruzione di Indici per l'Information Retrieval
- Algoritmi per la codifica e compressione dell'Informazione
- Funzione di Ranking documentale
- Introduzione al Vector Space Model
- Modello Probabilistici per l'Information Retrieval
- Valutazione dei Sistemi di IR
- Sviluppo efficiente e su larga scala di sistemi di IR
- Crawling e Detection di risorse duplicate
- Introduzione a IR Engines
- Introduzione a Map Reduce

# IR Laboratories

**Obiettivo:** studio e implementazione di alcuni dei paradigmi di Information Retrieval visti a lezione

- Vector Space Model
- Modelli Probabilistici
- Architetture di Retrieval distribuite

Verranno assegnati degli esercizi da completare prima della verbalizzazione

# Orari

- **Martedì, h. 14:30-16:30**
  - **(Aula 18 – Sogene)**
  
- **Mercoledì, h. 11:30-13:30**
  - **(Aula 18 – Sogene)**

**Ricevimento: termine della lezione o su appuntamento**

# Materiale a disposizione dello studente

Slides delle lezioni messe a disposizione dal docente

Slides dei laboratori e progetti software sviluppati a lezione

Testi consigliati

# Sito del Corso

I materiali pubblicati sono sul sito:

[http://sag.art.uniroma2.it/didattica/croce/IR\\_24\\_25](http://sag.art.uniroma2.it/didattica/croce/IR_24_25)

**Information Retrieval (a.a. 2019/20)**



Elenco dei File nel deposito



## Sommario Contenuti

1. [Novità](#)
2. [Programma del Corso](#)
3. [Testi di Riferimento](#)
4. [Link Utili](#)
5. [Diapositive delle lezioni](#)
6. [Progetti ed Esercizi Proposti](#)

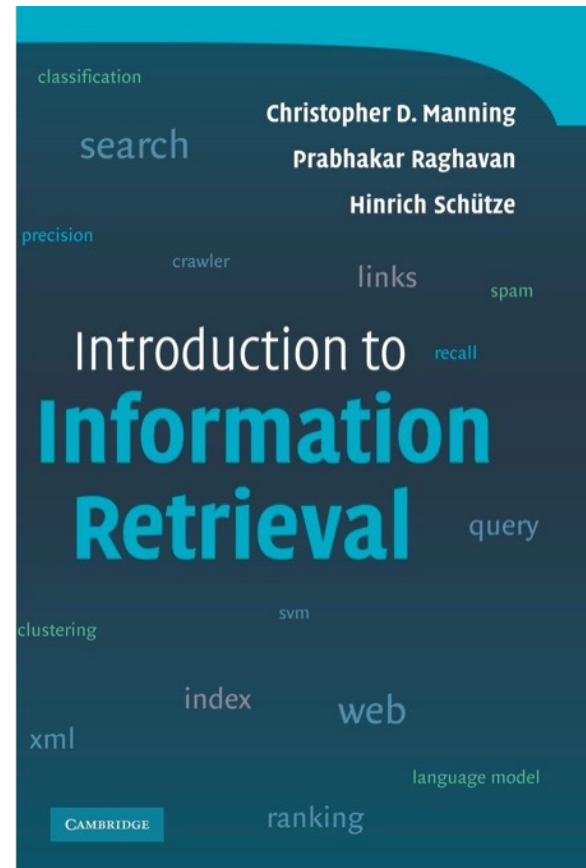
# Testi Consigliati

## *Introduction to Information Retrieval*

Christopher D. Manning,  
Prabhakar Raghavan and  
Hinrich Schütze

Cambridge University Press. 2008.

<http://nlp.stanford.edu/IR-book/>



# Organizzazione: Esami

- La prova scritta è composta da un **Test a Risposte Multiple** e **Domanda Aperta (Homework)**.
  - Essi verranno articolati in due Test In Itinere oppure in un'unica prova finale.
- La valutazione sarà mediata tra il punteggio ottenuto durante le prove scritte con un punteggio assegnato a valle di una **prova orale**.
- Progetto:
  - Lo studente **potrà** svolgere un progetto che completerà il voto finale per gli esami da **6 CFU**.
  - Il progetto è **obbligatorio** per gli studenti che dovranno sostenere l'esame da **9 CFU**.
  - *La complessità del progetto è legata al numero di CFU*



Domande?

# Action List

- Registrarsi al Corso presso Delphi presso :
  - URL: <https://delphi.uniroma2.it/totem/jsp/>
- Definire i propri estremi e tipo di Corso (ad es. i CFU e o i Corsi già sostenuti) tramite il campo “Note”
- Verranno pubblicati:
  - Elenchi dei gruppi registrati
  - Progetti
  - Orari ricevimento per gli studenti che non seguono
  - Slide e materiali complementari (*in progress*)