# ΜΥΕ003: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

# Εισαγωγή

*Ακαδημαϊκό Έτος 2024-2025*

- Ορισμός και βασικές εφαρμογές
- Ανάκτηση πληροφορίας και ΤΝ
- Περιεχόμενο μαθήματος και διαδικαστικά

Ανάκτηση Πληροφορίας (Information Retrieval) - (IR)

Είναι η εύρεση αντικειμένων (κυρίως εγγράφων) από μεγάλες συλλογές τα οποία ικανοποιούν μια ανάγκη για πληροφόρηση

Συνήθως αναζήτηση με λέξεις κλειδιά

Εφαρμογές;

# Γιατί να μας ενδιαφέρει;

Παλιότερα,

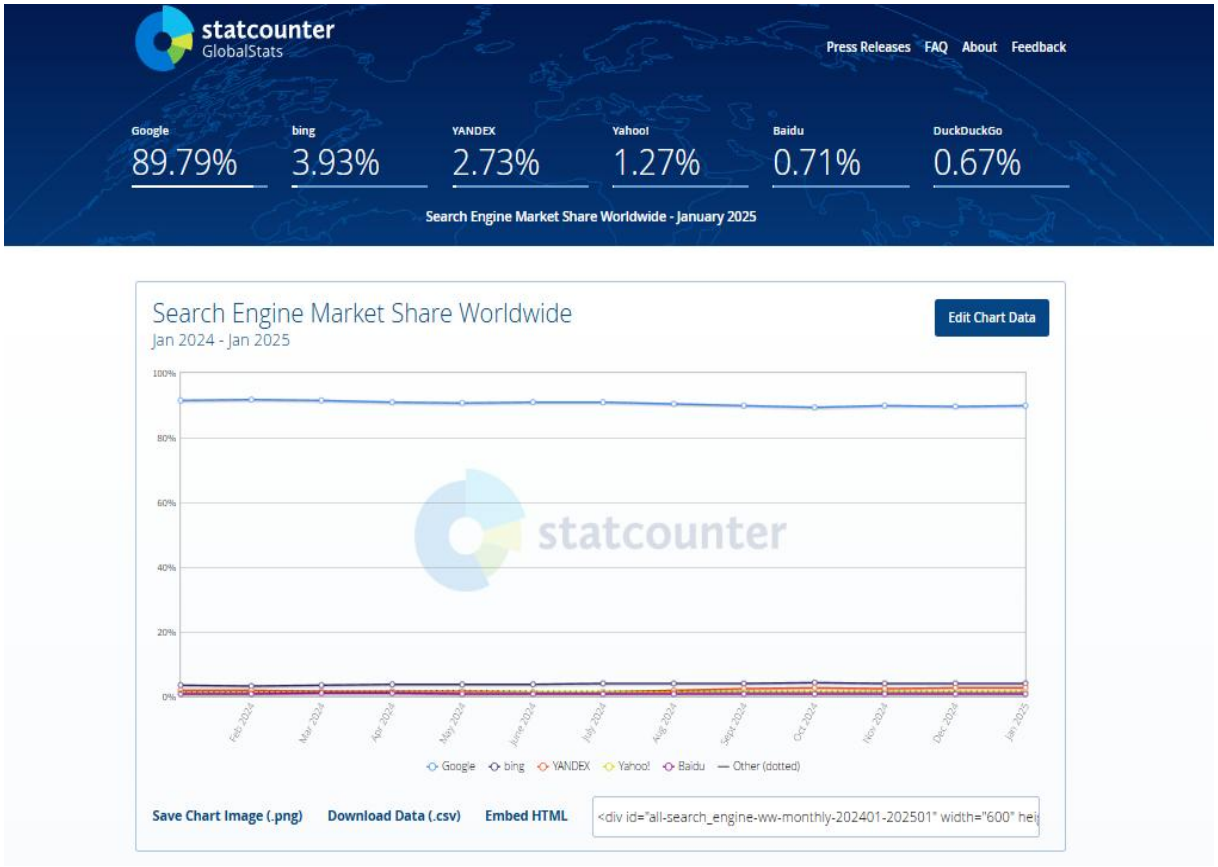
Βιβλιοθηκονόμους, βοηθούς νομικών επαγγελμάτων κλπ;

**ISBN**: 0-201-12227-8
**Author**: Salton, Gerard
**Title**: Automatic text processing: the transformation, analysis, and retrieval of information by computer
**Editor**: Addison-Wesley
**Date**: 1989
**Content**: <Text>

external attributes (metadata) and internal attribute (content)
Search by external attributes = Search in DB
IR: search by content

# Εφαρμογές

# Search Engines




Last 12 months, worldwide, all platforms

https://gs.statcounter.com/search-engine-market-share

Τα στατιστικά σε αυτή τη σελίδα είναι ενδεικτικά

## Google Search Statistics 2025 (Top Picks)

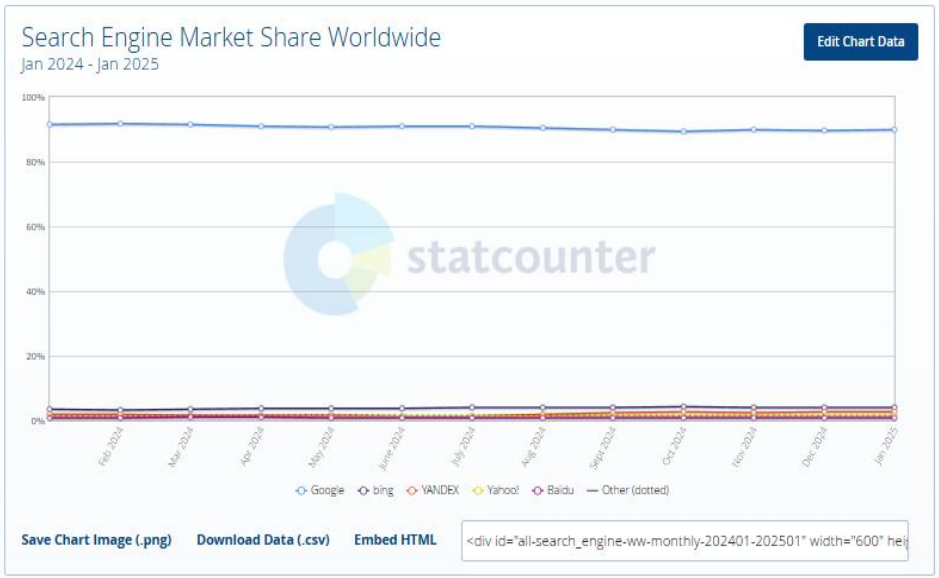
- 8.5 billion Google searches are made per day.
- Google processes approximately 99,000 searches per second.
- Approximately 4.97 billion people globally use Google.
- 84% of Google users conduct at least three searches each day.
- Google holds a dominant 90.01% share of the global search engine market.

- 1 in 5 Google searches are voice searches (rapidly increasing)
- 60% of all Google searches are from mobile devices
- 75% of people don't go past the first page of Google search results; Only 0.63% of the people click on the second page of the Google search results

| Google Search Result's Position | CTR Rate Recorded By Result |
|---|---|
| #1 | 27.6% |
| #2 | 15.8% |
| #3 | 11.0% |
| #4 | 8.4% |
| #5 | 6.3% |
| #6 | 4.9% |
| #7 | 3.9% |
| #8 | 3.3% |
| #9 | 2.7% |
| #10 | 2.4% |

- Gemini, previously known as Bard +added AI functionality

https://www.demandsage.com/google-search-statistics/

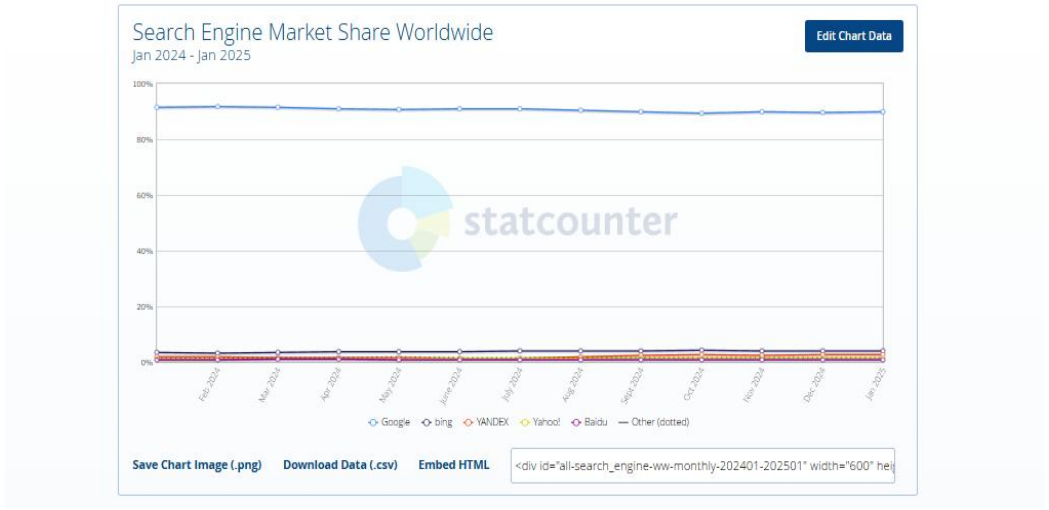

- ***Bing:*** Microsoft
  Build on previous search engines (MSN Search, Windows Live Search, Live Search),
  In 2023, faster than Google to release an AI Chatbot service (Bing Chat) based on GPT4,
  Merged with *Copilot*

- ***Yandex:*** ρωσική πολυεθνική τεχνολογική εταιρεία που ειδικεύεται σε υπηρεσίες και προϊόντα σχετικά με το Διαδίκτυο και θεωρείται η μεγαλύτερη εταιρεία τεχνολογίας της *Ρωσίας*. Λειτουργεί την μεγαλύτερη μηχανή αναζήτησης στη Ρωσία με μερίδιο αγοράς περίπου *65%*.

*Πηγή: Wikipedia*

https://gs.statcounter.com/search-engine-market-share

# Search Engines


- ***Baidu:*** 2η μεγαλύτερη μηχανή αναζήτησης στον κόσμο, και κατέχει το 76.05% του μεριδίου αγοράς στην αγορά μηχανών αναζήτησης της *Κίνας*.

- ***DuckDuckGo*** είναι διαδικτυακή μηχανή αναζήτησης που δίνει έμφαση στην *προστασία της ιδιωτικής ζωής* των χρηστών της και στην αποφυγή του "*φίλτρου φυσαλίδας*" των εξατομικευμένων αποτελεσμάτων αναζήτησης.
  - *Πηγή: Wikipedia*

Last 12 months, worldwide, all platforms

https://gs.statcounter.com/search-engine-market-share

# Desktop search

IR tools are designed to find information on a PC, including web browser history, e-mail archives, text documents, images, video, etc

# Εφαρμογές

Email search

Social search

Enterprise search

helps people in an organization find the information they need to perform their jobs-- data extracted from inside the business, along with external data sources like document management systems, databases, etc

Domain specific search: Legal information retrieval, Digital libraries

Διαφορετικές απαιτήσεις ανάλογα με την εφαρμογή

Domain-specific

# Παράδειγμα: WestLaw

https://legal.thomsonreuters.com/en/products/westlaw

- Μεγάλο εμπορικό (συνδρομές επί πληρωμή) σύστημα
- Αναζήτηση σε νομικά κείμενα (άρχισε το1975, η διάταξη προστέθηκε το 1992)
- Boolean ερωτήματα και ερωτήματα φυσικής γλώσσας
- 40,000 databases for case laws, 60 countries (source: Wikipedia)

# Παράδειγμα: WestLaw

- Παράδειγμα (απόσταση ανάμεσα στους όρους):
  - *Ανάγκη πληροφόρησης:* What is the statute of limitations in cases involving the federal tort claims act?
  - *Ερώτημα*:

    LIMIT! /3 STATUTE ACTION /S FEDERAL /2 TORT /3 CLAIM
    - /3 = within 3 words, /S = in same sentence

- Παράδειγμα (ανεκτική ανάκτηση):
  - *Ανάγκη πληροφόρησης:* Information on the legal theories involved in preventing the disclosure of trade secrets by employees formerly employed by a competing company
  - *Ερώτημα:*

    "trade secret" /s disclos! /s prevent /s employe!

# Κατηγορίες εφαρμογών

- Στο web/διαδίκτυο (search engines)

  Δισεκατομμύρια έγγραφα σε εκατομμύρια υπολογιστές.

  Συλλογή εγγράφων, κλίμακα, διάταξη αποτελεσμάτων, ..

- Προσωπική ανάκτηση πληροφορίας

  (στον προσωπικό υπολογιστή, email, κλπ)

  Διαφορετικά είδη αρχείων, light-weight, maintenance-free, …

- Σε επίπεδο επιχείρησης, οργανισμού (enterprise, institutional)
- Αναζήτηση ειδικού σκοπού (domain-specific search) – πχ ερευνητικά άρθρα σε βιοχημεία

# Ορισμός

Ανάκτηση Πληροφορίας (Information Retrieval) - (IR)

- είναι η εύρεση αντικειμένων κυρίως εγγράφων (**documents**) αδόμητης φύσης (*) (unstructured) που συνήθως έχουν τη μορφή κειμένου (**text**)
- από μεγάλες συλλογές (συνήθως αποθηκευμένες σε υπολογιστές)
- τα οποία ικανοποιούν μια ανάγκη πληροφόρησης (**information need**)

*(*) όχι ακριβώς!*

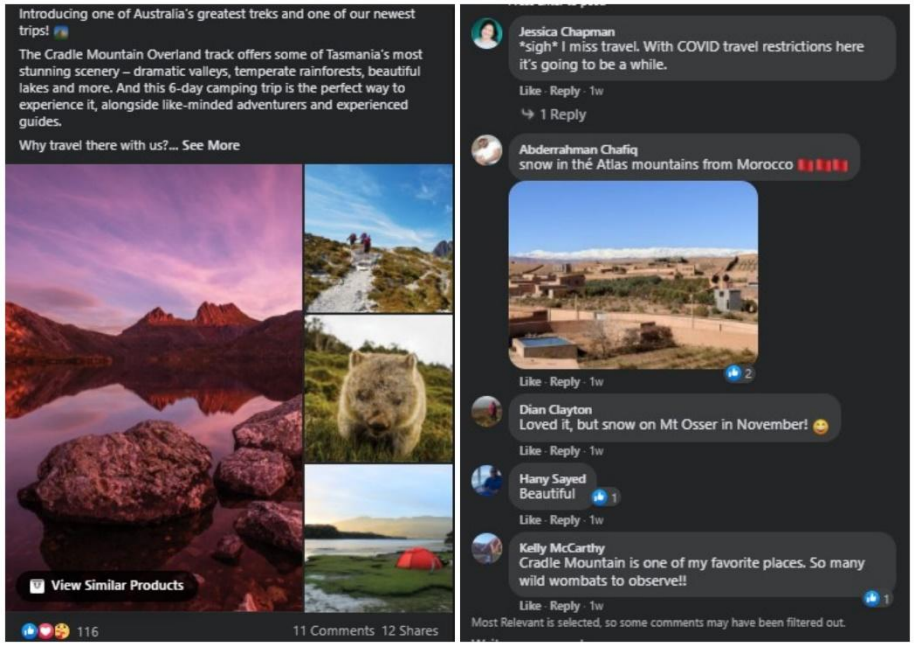
# Αδόμητα δεδομένα

- Τυπικά αναφέρεται σε *ελεύθερο κείμενο*
- Επιτρέπει
  - Ερωτήματα με λέξεις κλειδιά (keyword) με πιθανούς τελεστές
  - Ποιο περίπλοκες ερωτήσεις για έννοιες: π.χ.,
    - Βρες όλες τις web σελίδες για την απελευθέρωση των Ιωαννίνων
- Κλασσικό μοντέλο για αναζήτηση σε έγγραφα κειμένου

structured

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Purchase ID | Last name | First name | Birthday | Country | Date of purchase | Amount of purchase |
| 2 | 1 | Davidson | Michael | 04/03/1986 | United States | 10/12/2016 | 37 |
| 3 | 2 | Vito | Jim | 09/01/1994 | United Kingdom | 02/02/2016 | 85 |
| 4 | 3 | Johnson | Tom | 23/08/1972 | France | 02/11/2016 | 83 |
| 5 | 4 | Lewis | Peter | 18/10/1979 | Germany | 22/11/2016 | 27 |
| 6 | 5 | Koenig | Edward | 13/05/1983 | Argentina | 26/03/2015 | 43 |
| 7 | 6 | Preston | Jack | 16/06/1991 | United States | 06/11/2016 | 77 |
| 8 | 7 | Smith | David | 11/03/1965 | Canada | 15/11/2016 | 23 |
| 9 | 8 | Brown | Luis | 03/09/1997 | Australia | 03/07/2015 | 74 |
| 10 | 9 | Miller | Thomas | 07/01/1980 | Germany | 07/11/2016 | 13 |
| 11 | 10 | Williams | Bill | 26/07/1960 | United States | 20/11/2015 | 80 |
| 12 | 11 | Gemini | Alexia | 12/09/1995 | Canada | 11/03/2017 | 35 |
| 13 | 12 | Bond | James | 25/02/1975 | United Kingdom | 12/08/2017 | 40 |
| 14 | 13 | Burgle | Patricia | 01/12/1990 | United States | 18/01/2015 | 55 |
| 15 | 14 | Reding | Michelle | 07/04/1985 | Canada | 23/02/2017 | 28 |
| 16 | 15 | Harvey | Billy | 14/07/1971 | United Kingdom | 12/01/2016 | 41 |
| 17 | | | | | | | |

unstructured




https://www.altexsoft.com/blog/structured-unstructured-data/

# Ημιδομημένα δεδομένα

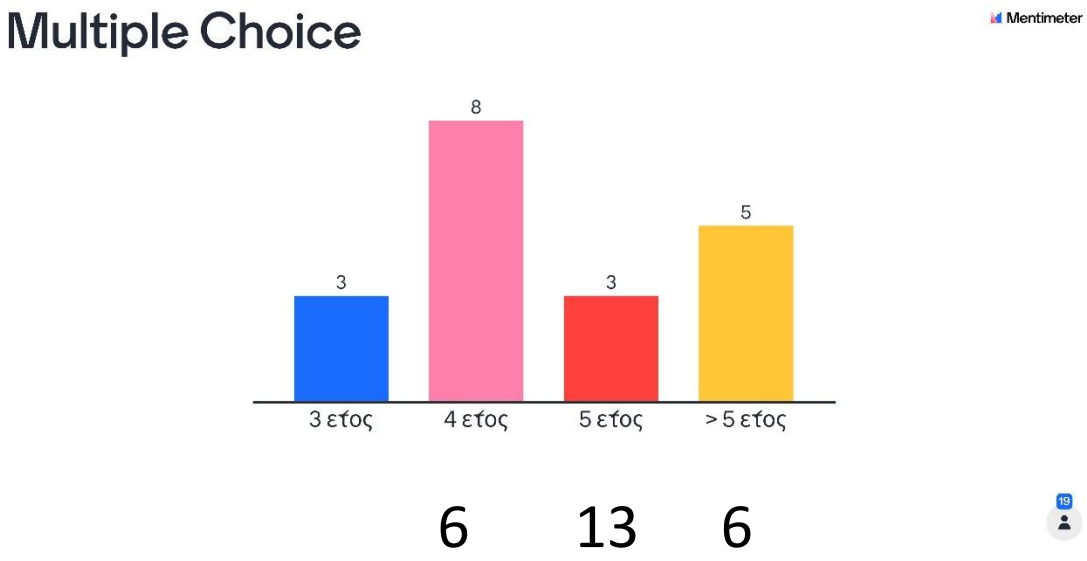
- Στην πραγματικότητα, δεν υπάρχουν αμιγώς μη δομημένα δεδομένα
  - π.χ., αυτή η διαφάνεια έχει διακριτές ζώνες όπως *Title* και *Bullets*
  - *Web pages?*
  - *Emails?*
- «Ημιδομημένη» αναζήτηση όπως:
  - *Title* contains ημιδομημένα AND *Bullets* contain αναζήτηση

… και βέβαια υπάρχει πάντα η γλωσσική δομή

# Τι είναι η Ανάκτηση Πληροφορίας (Information Retrieval);

- Ορισμός και βασικές εφαρμογές
- Ανάκτηση πληροφορίας και ΤΝ
- Περιεχόμενο μαθήματος και διαδικαστικά

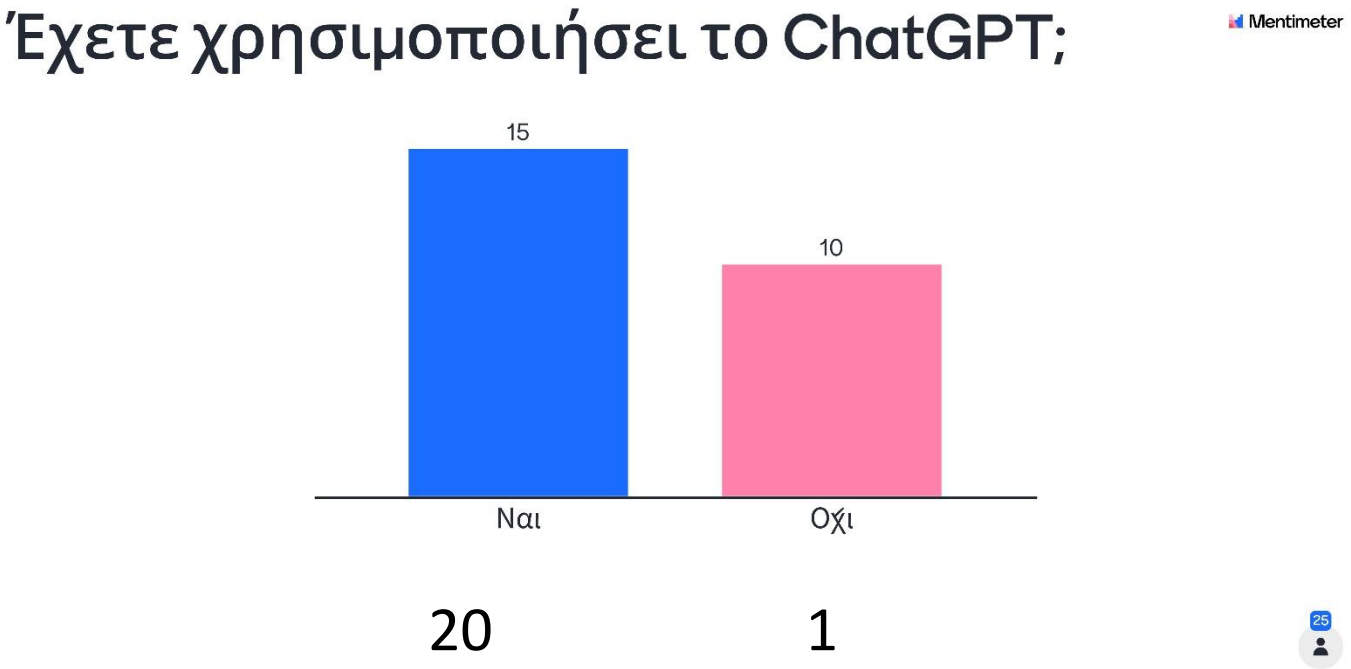
# Poll 1: Ποιοι είστε;



# ChatGPT

- Chatbot interface that provides access to OpenAI's language models, including GPT-3.5 and GPT-4 Omni (O3)
- Based on the Transformer architecture, trained using deep learning.
- Trained on vast amounts of data to generate human-like responses
- Generates the most probable next token
- O3 uses probabilistic sampling to make responses more natural

# ChatGPT


Έχετε χρησιμοποιήσει το ChatGPT;
Mentimeter
15
10
Ναι
Οχι
20
1
25

# The three models of satisfying information needs

| | Structures queries | Search Engines | Large Language Models |
|---|---|---|---|
| **Systems** | **DBMS** | **Google search** | **ChatGPT** |
| **Corpus** | **Relational tables** | **Web+** | **Web texts, books, Wikipedia** |
| **Input** | **SQL** | **Keywords** | **Natural language** |
| **Output** | **Data in tables** | **Ranked sources** | **New content** |

ChatGPT uses **dense retrieval** (embeddings) rather than keyword matching as in IR

# From ChatGPT

| Feature | Traditional IR | ChatGPT |
| --- | --- | --- |
| Retrieves documents | ✅ | ❌ |
| Keyword-based search | ✅ | ❌ |
| Understands intent | ❌ | ✅ |
| Generates summaries | ❌ | ✅ |

# ChatGPT

Poll 2
Μέχρι 3 πιο θετικά
Μέχρι 3 πιο αρνητικά

# ChatGPT

## Μέχρι 3 πιο θετικά

Mentimeter


14

# ChatGPT

## Μέχρι 3 πιο αρνητικά

Limitations of ChatGPT in IR (by ChatGPT)

- *Lack of explicit retrieval:* Does not return source documents like a search engine.
- *Hallucination problem:* May generate incorrect or misleading answers.
- *Bias & Ethics:* Reflects biases from training data.
- *Difficult to evaluate:* No citations, making accuracy hard to verify.
- *High computational cost:* Requires significant resources to run.

Λογοκλοπή (οικειοποιείται κείμενο άλλων)

Citations/accountability

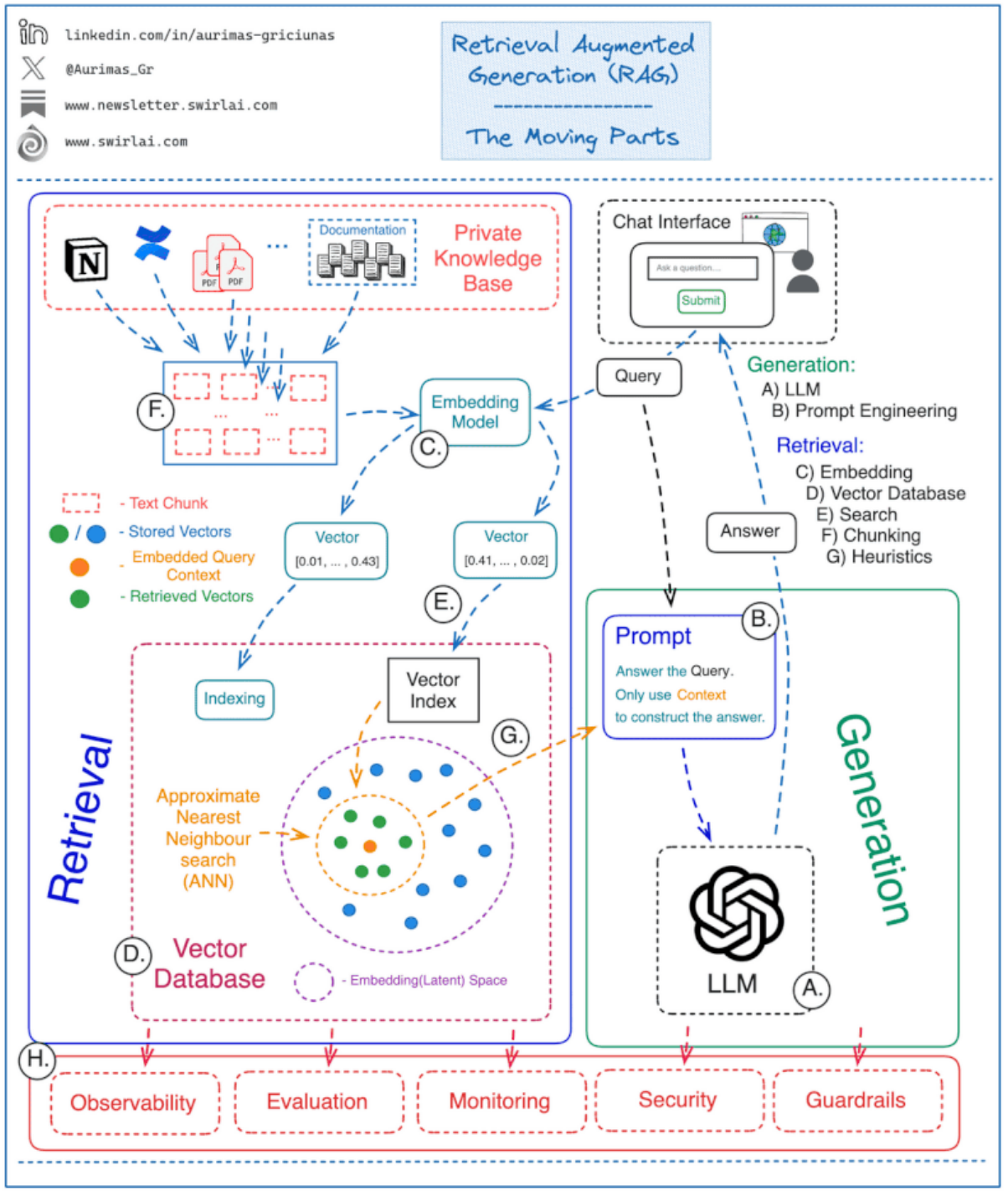
Machine generated context as input to LLM

Δύο οπτικές γωνίες/“disruptive” σε δύο άξονες

- Εκπαίδευση γενικά
- Στο αντικείμενο του μαθήματος: πως επηρεάζει την αναζήτηση

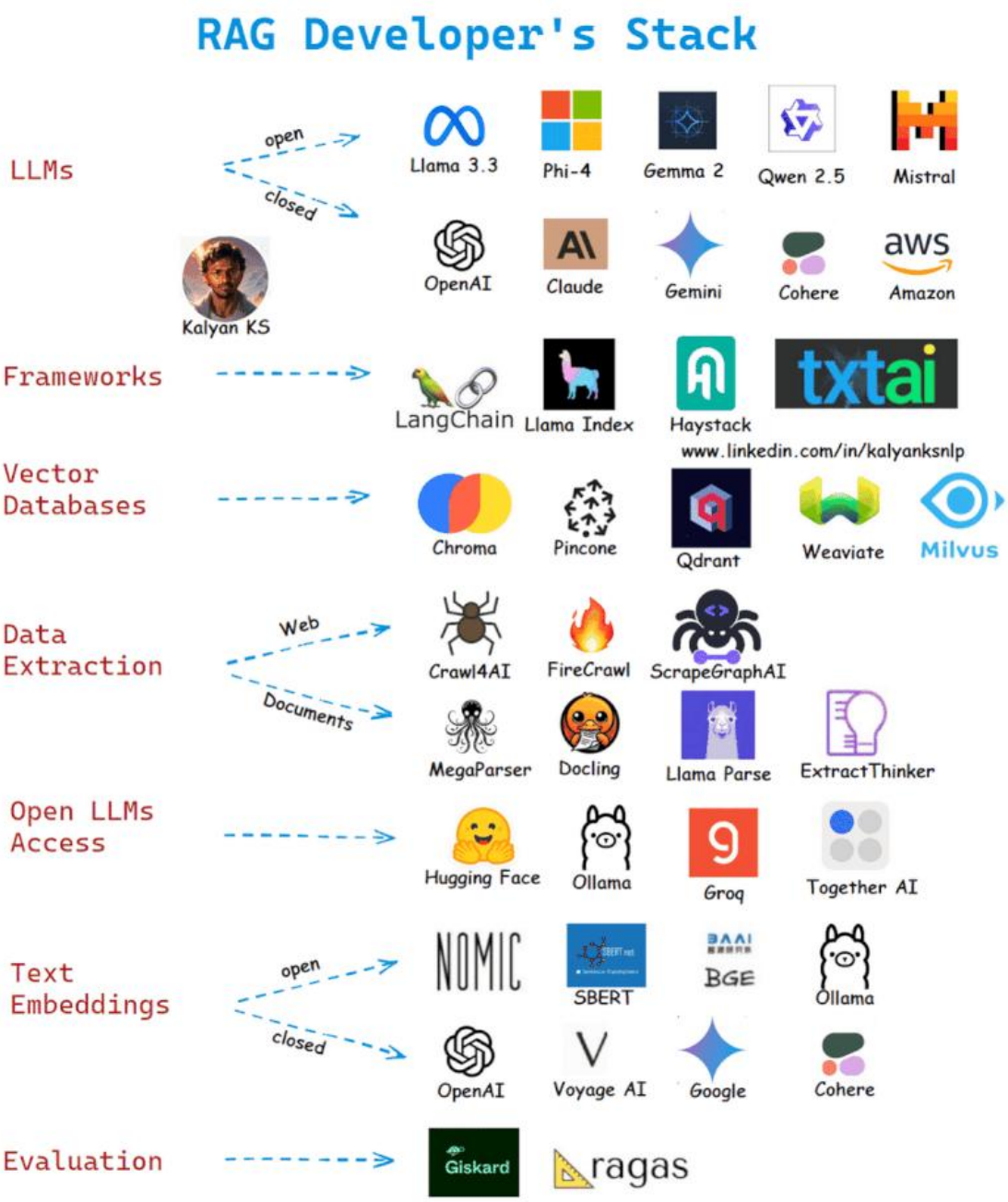
# RAG

# Retrieval-Augmented Generation



# RAG

# Retrieval-Augmented Generation



- Ορισμός και βασικές εφαρμογές
- Ανάκτηση πληροφορίας και ΤΝ
- Περιεχόμενο μαθήματος και διαδικαστικά

# Μηχανές Αναζήτησης

## Φάση 1: Δημιουργία της μηχανής αναζήτησης

- Δημιουργία συλλογής (αν δεν υπάρχει):
  - crawl, scrap, use specific APIs (social media)
- Ανάλυση του κειμένου
  - Ποιοι θα είναι οι όροι, περιλαμβάνει και γλωσσολογική επεξεργασία
- Κατασκευή ευρετηρίων
  - Η πιο απλή μορφή: για κάθε όρο σε ποια έγγραφα εμφανίζεται

(επεκτάσεις πχ με συχνότητα εμφάνισης, πληροφορία θέσης, συμπίεση, κλπ)

# Μηχανές Αναζήτησης

## Φάση 2: Λειτουργία

Ο χρήστης υποβάλει ένα ερώτημα

Επεξεργασία του ερωτήματος

Χρήση του ευρετηρίου για να βρούμε (retrieve) τα συναφή έγγραφα και να τα διατάξουμε με βάση τη συνάφεια

Πως ορίζουμε τη συνάφεια:

- Boolean model: υπάρχει, δεν υπάρχει ο όρος
- tf-idf (βασισμένη σε κείμενο)
- Source importance: Pagerank (οι συνδέσεις), clickthrough (γενικά traffic), document age, authority (wikipedia)
- Personalization, contextualization
- User intent
- Learning to rank

# Μηχανές Αναζήτησης

## Αξιολόγηση

Δε μας ενδιαφέρει η διάταξη:
Recall/Precision/AUC

Μας ενδιαφέρει η διάταξη:
MAP, NDCG

## Word embeddings + ML

Διανυσματικές αναπαραστάσεις όρων
word2vec (σε βάθος)
Transformers, LLMs (σε υψηλό επίπεδο)

## Lucene

Open-source software για τη δημιουργία μηχανών αναζήτησης

# Αναζήτηση

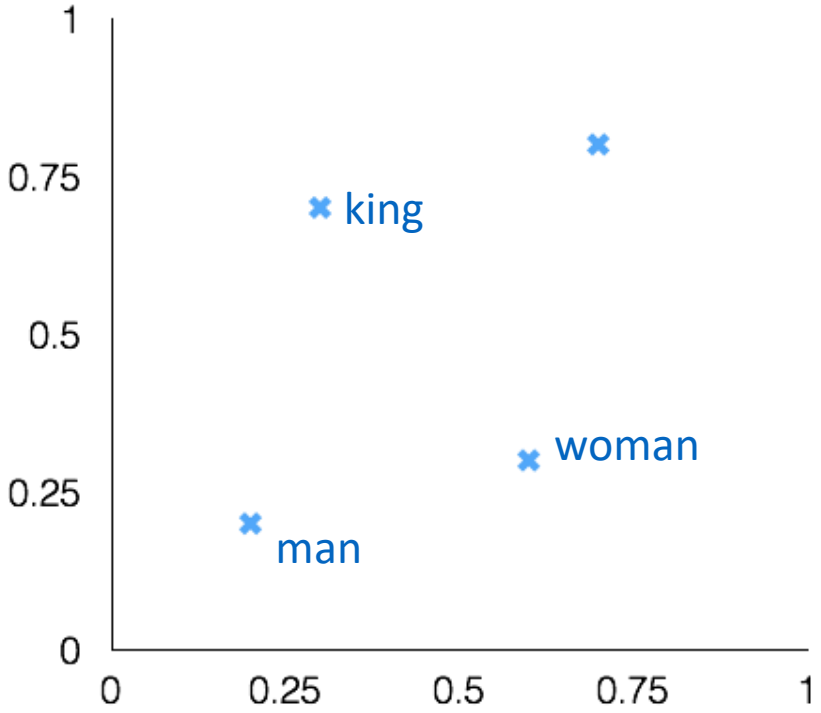
- Μοντελοποίηση
- Επεξεργασία
- Βασικές δομές
- Αξιολόγηση

+ RAG (chunking, vector search, vector databases)

# Μερικά Στοιχεία Μηχανικής Μάθησης

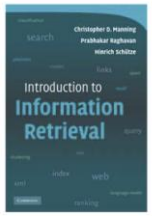
Word embeddings

| | | |
|---|---|---|
| + | king | [ 0.30 0.70 ] |
| − | man | [ 0.20 0.20 ] |
| + | woman | [ 0.60 0.30 ] |
| | queen | [ 0.70 0.80 ] |



# Διαδικαστικά

- Ιστοσελίδα
- Βιβλίο
  - Cristopher D. Manning, Prabhakar Raghavan and Hinrich Schutze. *Εισαγωγή στην Ανάκτηση Πληροφοριών*, Εκδόσεις Κλειδάριθμος
  - Η αγγλική έκδοση διαθέσιμη δωρεάν




- Ricardo Baeza-Yates and Berthier Ribeiro-Neto, *Ανάκτηση Πληροφορίας*, 2η Έκδοση, Εκδόσεις Τζιόλα

# Διαδικαστικά

- Βαθμολογία (μπορεί να αλλάξει):
  - Εργασία (έως 2 άτομα) – σε φάσεις: 50% ή 70% (προαιρετικό κομμάτι)
    γραπτή εξέταση 50% ή 30%
  - Προκαταρκτικός σχεδιασμός

| | |
|---|---|
| Ανάθεση | 27/3/2025 |
| Παράδοση 1ης Φάσης | 11/4/2025 |
| Παράδοση 2ης Φάσης | 23/5/2025 |
| Προφορική Εξέταση | Εβδομάδα 26/5/205 |

- Η εργασία δεν «κρατιέται»
- Για να περάσετε το μάθημα, βαθμός διαγωνίσματος $\geq 4$

# Λίγα λόγια για την εργασία

Μηχανή αναζήτησης

Lucene (solar) +

Φέτος + RAG

Θεματικό (παλαιότερα, ταινίες, αναζήτηση τραγουδιών (στίχοι), επιστημονικά άρθρα)

# ΤΕΛΟΣ Εισαγωγής

Ερωτήσεις;

# ΜΥΕ003: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

# Ανάκτηση Boole

*Ακαδημαϊκό Έτος 2024-2025*

# Περιεχόμενο

1. Μερικές βασικές έννοιες

2. Ένα απλό ΣΑΠ

   Μια μικρή εισαγωγή στο απλούστερο μοντέλο αναζήτησης (Boolean) (Κεφάλαιο 1 του βιβλίου)

   *Ένα απλό σύστημα ΑΠ (βασικές δομές δεδομένων και παραδείγματα ερωτημάτων)*

# Βασικές ‘Εννοιες

# Τι είναι η Ανάκτηση Πληροφορίας (Information Retrieval);

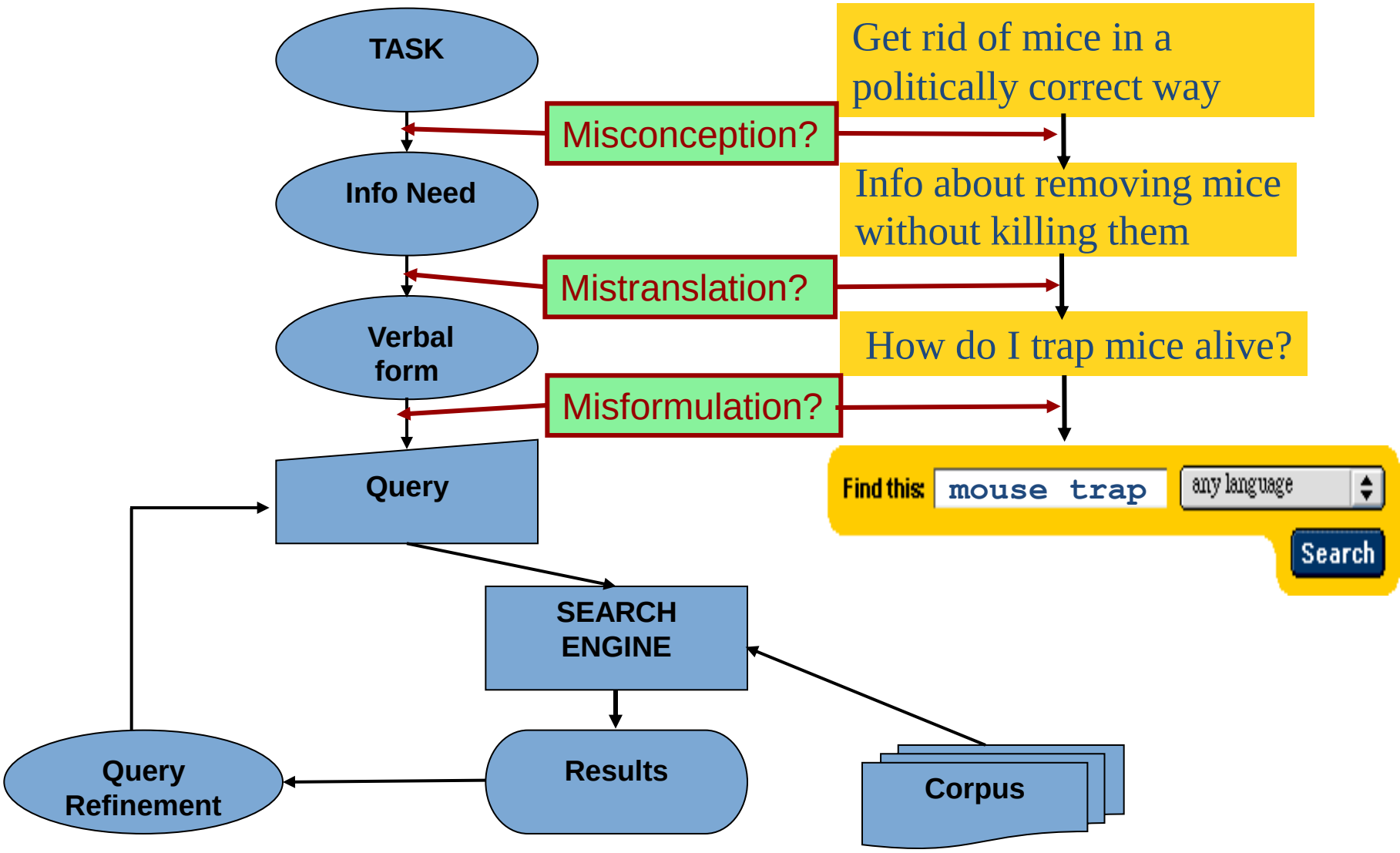
# Βασικές Έννοιες

Συλλογή (Collection - corpus): Σύνολο από έγγραφα

Στόχος: Ανάκτηση των εγγράφων που περιέχουν πληροφορία που είναι συναφής (relevant) με την ανάγκη πληροφόρησης (information need) του χρήστη και τον βοηθά να ολοκληρώσει κάποιο έργο (task)

- ✓ Διαφορά μεταξύ: information need και ερωτήματος (query)
- ✓ Ad hoc retrieval

# Το κλασικό μοντέλο αναζήτησης (search model)



# Βασικά Βήματα

(προεπεξεργασία)

- Σύλλεξε τα έγγραφα
- Ανάλυση
- Κατασκεύασε βοηθητικές δομές – ευρετήρια

(λειτουργία)

- Επεξεργασία ερωτήσεων

Αρχικά θα δούμε την απλούστερη μορφή:

Boolean retrieval

# Προεπεξεργασία (Μηχανές Αναζήτησης)

## Δημιουργία της μηχανής αναζήτησης

- Δημιουργία συλλογής (αν δεν υπάρχει):
  - crawl, scrap, use specific APIs (social media)
- Ανάλυση του κειμένου
  - Ποιοι θα είναι οι όροι, περιλαμβάνει και γλωσσολογική επεξεργασία
- Κατασκευή ευρετηρίων
  - Η πιο απλή μορφή: για κάθε όρο σε ποια έγγραφα εμφανίζεται

(επεκτάσεις πχ με συχνότητα εμφάνισης, πληροφορία θέσης, συμπίεση, κλπ)

# Λειτουργία (Μηχανές Αναζήτησης)

Ο χρήστης υποβάλει ένα ερώτημα

Επεξεργασία του ερωτήματος

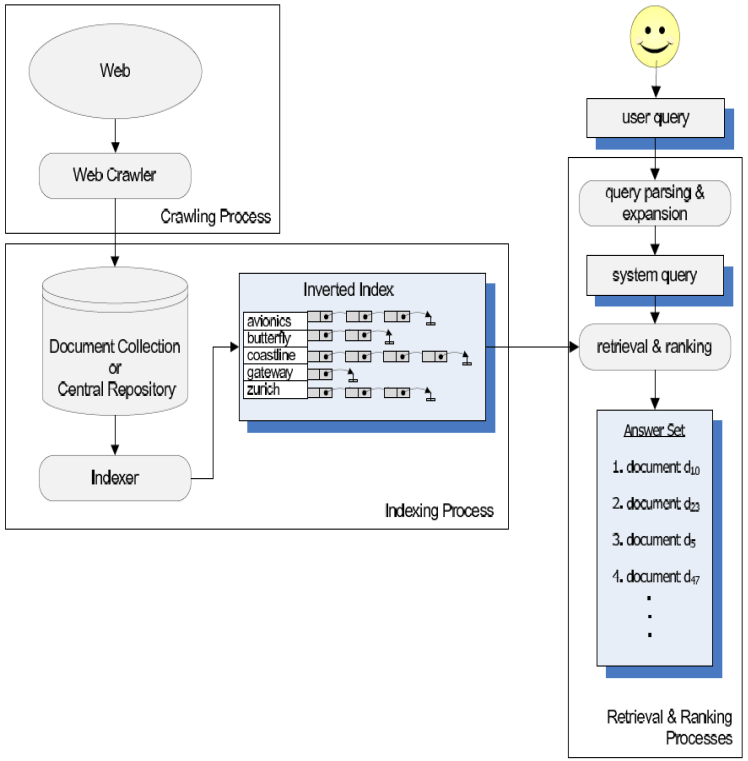
Χρήση του ευρετηρίου για να βρούμε (retrieve) τα συναφή έγγραφα και να τα διατάξουμε με βάση τη συνάφεια

Πως ορίζουμε τη συνάφεια:

- Boolean model: υπάρχει, δεν υπάρχει ο όρος
- tf-idf (βασισμένη σε κείμενο)
- Source importance: Pagerank (οι συνδέσεις), clickthrough (γενικά traffic), document age, authority (wikipedia)
- Personalization, contextualization
- User intent
- Learning to rank

# Architecture of the IR System

- High level software architecture of an IR system



# Βασικές έννοιες

Αποτέλεσμα σε διάταξη με βάση τη συνάφεια

Αξιολόγηση:

- πέρα από την απόδοση (efficiency)
  αποτελεσματικότητα (effectiveness)

Αποτελεσματικότητα (effectiveness): Πόσο *καλά (χρήσιμα, συναφή)* είναι τα έγγραφα που ανακτήθηκαν;

# Αποτελεσματικότητα

- *Ακρίβεια (Precision)*: Το ποσοστό των εγγράφων που ανακτήθηκαν που είναι συναφή με την ανάγκη πληροφόρησης του χρήστη
- *Ανάκληση (Recall)* : Το ποσοστό των συναφών με την ανάγκη πληροφόρησης του χρήστη εγγράφων της συλλογής που ανακτήθηκαν από το σύστημα
  - Περισσότερα στο μέλλον

παράδειγμα Συλλογή 1000 έγγραφα
υπάρχουν 20 συναφή έγγραφα με το ερώτημα
το ΣΑΠ μας επιστρέφει 30 έγγραφα (17 συναφή, 13 μη συναφή)

$$precision = \frac{17}{30}$$

$$recall = \frac{17}{20}$$

# Boolean Ανάκτηση

# Boolean μοντέλο

- Επιστρέφονται ως απάντηση όλα τα κείμενα που ικανοποιούν το ερώτημα *χωρίς διάταξη*
  - *Δυαδική συνάφεια (συναφές, μη συναφές)*
- Κείμενο ως σύνολο όρων
- Οι χρήστες διατυπώνουν ερωτήματα με τη μορφή *Boolean εκφράσεων*, δηλαδή όρων συνδυασμένων με AND, OR και NOT
- Συναφές αν περιέχει τους όρους

# Αδόμητα δεδομένα το 1680

Shakespeare's Collected Works

# Αδόμητα δεδομένα το 1680

- Ποια θεατρικά έργα του Shakespeare περιέχουν τις λέξεις ***Brutus*** *και* ***Caesar*** αλλά όχι τη λέξη ***Calpurnia***
  - Ερώτημα*: **Brutus** AND **Caesar** AND NOT **Calpurnia***

# Οι απαντήσεις:

- Antony and Cleopatra, Act III, Scene ii

*Agrippa* [Aside to DOMITIUS ENOBARBUS]: Why, Enobarbus,
When Antony found Julius ***Caesar*** dead,
He cried almost to roaring; and he wept
When at Philippi he found ***Brutus*** slain.

- Hamlet, Act III, Scene ii

*Lord Polonius:* I did enact Julius ***Caesar*** I was killed i' the Capitol; ***Brutus*** killed me.

# Αδόμητα δεδομένα το 1680

- Ποια θεατρικά έργα του Shakespeare περιέχουν τις λέξεις ***Brutus*** *και* ***Caesar*** αλλά όχι τη λέξη ***Calpurnia***
  - Ερώτημα*: **Brutus** AND **Caesar** AND NOT **Calpurnia***
- Να διαβάσουμε όλα τα έργα σειριακά από την αρχή σημειώνοντας …
- Θα μπορούσαμε να κάνουμε `grep` σε όλα τα έργα για ***Brutus*** και ***Caesar,*** και να σβήσουμε τις γραμμές που περιέχουν τη λέξη ***Calpurnia***

# Αδόμητα δεδομένα το 1680

- Γιατί όχι grep (ή κάποιο άλλο filter)?
  - Αργό (για μεγάλες συλλογές)
  - Grep line-oriented, η ανάκτηση πληροφορίας document-oriented
  - *NOT* ***Calpurnia*** δεν είναι εύκολο
  - Επιπρόσθετη λειτουργικότητα (π.χ., βρες τη λέξη ***Romans*** κοντά στο ***countrymen***)
  - Διάταξη! Ranked retrieval (τα «καλύτερα» έγγραφα ανάμεσα σε αυτά που ικανοποιούν την ερώτηση)
    - Σε επόμενα μαθήματα ….

*Θα* ***προ-επεξεργαστούμε*** *τα έγγραφα και θα δημιουργήσουμε ευρετήρια*

Υπόθεση: έχει γίνει ανάλυση και κάθε κείμενο είναι μια συλλογή από όρους

# Ανάλυση

Έγγραφα από τη συλλογή

Friends, Romans, countrymen.

*Θα τα δούμε στο επόμενο μάθημα*

Tokenizer

Token stream

Friends | Romans | Countrymen

Linguistic modules

Γλωσσολογική επεξεργασία

Modified tokens

friend | roman | countryman

Όροι (terms)

# Για να δούμε τα βασικά ...

## Boolean μοντέλο

Δυαδική μήτρα (πίνακας) σύμπτωσης $M$

Γραμμές: Term (όροι, λέξεις)

Στήλες: Document (έγγραφα, έργα)

$M[i, j] = 1$, αν ο όρος $i$ εμφανίζεται στο έγγραφο $j$

0, αλλιώς

# Term-document incidence matrix (μήτρα σύμπτωσης)

| | 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 |

1 αν το έργο περιέχει τη λέξη, 0 αλλιώς

***Brutus*** *AND* ***Caesar*** *BUT NOT* ***Calpurnia***

# Οι όροι και τα έγγραφα ως διανύσματα

Έχουμε ένα *δυαδικό διάνυσμα* για κάθε *όρο* και κάθε *έγγραφο*

- Για να απαντήσουμε στην ερώτηση: παίρνουμε τα διανύσματα για το ***Brutus, Caesar*** και το *συμπλήρωμα* του διανύσματος για το ***Calpurnia*** → bitwise *AND*.

  110100 *AND* 110111 *AND* 101111 = 100100.

παράδειγμα

$d_1$ a b c a

$d_2$ d a b

$d_3$ b a f

$d_4$ a e

$d_5$ d f

# Μεγαλύτερες συλλογές

- Ας θεωρήσουμε $N$ = 1 εκατομμύρια έγγραφα, το καθένα έχει περίπου 1000 (διακριτούς) όρους (~2-3 σελίδες βιβλίου).
- Έστω ότι ανάμεσα τους υπάρχουν $M$ = 500K διακριτοί (*distinct*) όροι.

Πόσα κελιά έχει ο πίνακας σύμπτωσης;

A. 1 δισεκατομμύριο
B. 500 δισεκατομμύρια
C. 500 εκατομμύρια
D. 5 δισεκατομμύρια

Πόσα μη μηδενικά κελιά (δηλαδή 1) έχει ο πίνακας σύμπτωσης;

A. 1 δισεκατομμύριο
B. 1 εκατομμύριο
C. 50 δισεκατομμύρια
D. 50 εκατομμύρια

Ποσοστό των 1:

# Πόσο είναι το μέγεθος του πίνακα;

- Ο 500K x 1M πίνακας έχει μισό τρισεκατομμύριο 0's και 1.
- Αλλά δεν έχει περισσότερα από ένα δισεκατομμύριο 1.
  - Ο πίνακας είναι εξαιρετικά αραιός (sparse) – τουλάχιστον το 99.8% είναι 0.
- Ποια είναι μια καλύτερη αναπαράσταση;

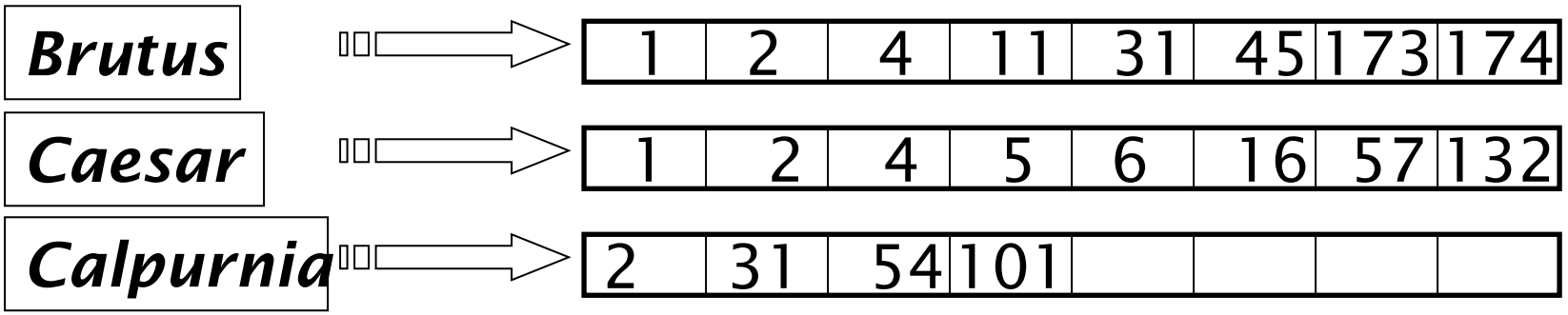
# Αντεστραμμένο ευρετήριο ή αρχείο (Inverted index/file)

Για κάθε *όρο (term) t*, διατηρούμε μια λίστα με όλα τα έγγραφα που περιέχουν τον όρο.

- Κάθε έγγραφο χαρακτηρίζεται από ένα αναγνωριστικό εγγράφου (**docID**), πχ αριθμό που ανατίθεται σειριακά στα έγγραφα κατά τη δημιουργία τους

# Αντεστραμμένο ευρετήριο

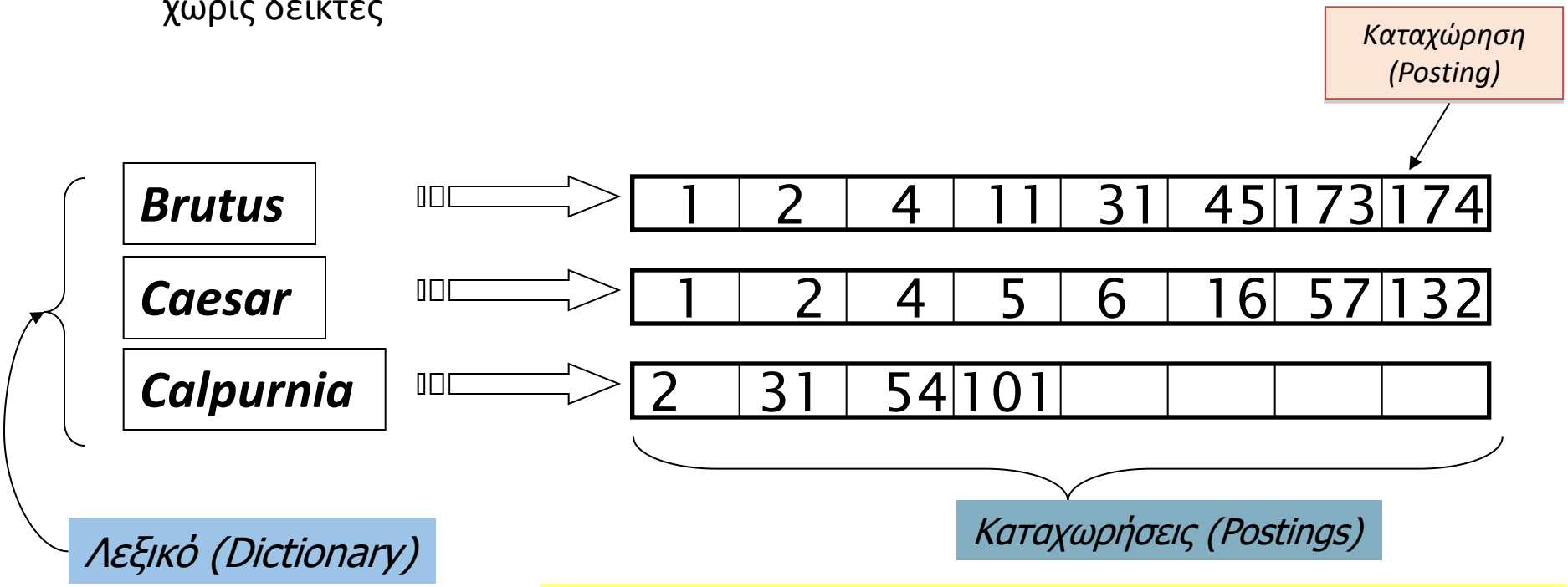
- Μπορούμε να χρησιμοποιήσουμε σταθερού μεγέθους arrays για αυτό?




Τι γίνεται αν η λέξη ***Caesar*** προστεθεί στο έγγραφο 14?

# Αντεστραμμένο ευρετήριο

- Χρειαζόμαστε μεταβλητού μεγέθους **λίστες καταχωρήσεων (postings lists)**
  - *Ποια δομή δεδομένων είναι κατάλληλη;*
    - Στη μνήμη, απλά-διασυνδεδεμένες λίστες (skip lists) ή πίνακες μεταβλητού μήκους
    - Στο δίσκο, ως (συμπιεσμένες) συνεχόμενες ακολουθίες καταχωρήσεων χωρίς δείκτες




Σε διάταξη με βάση το docID (θα δούμε σε λίγο γιατί!).

παράδειγμα

$d_1$ a c b c a

$d_2$ d a b

$d_3$ b a f

$d_4$ a e

$d_5$ d f

# Βασική Ορολογία

- Αντεστραμμένο ευρετήριο (Inverted index)
- Λίστες καταχωρήσεων (posting lists) – μία για κάθε όρο
  - Καταχώρηση – ένα στοιχείο της λίστας
  - ✓ Κάθε λίστα είναι διατεταγμένη με το DocID
- Λεξιλόγιο (Vocabulary): το σύνολο των όρων
- Λεξικό (Dictionary) δομή δεδομένων για τους όρους
  - ✓ Αρχικά ας θεωρήσουμε αλφαβητική διάταξη

*Το δημιουργούμε από πριν, θα δούμε πως*

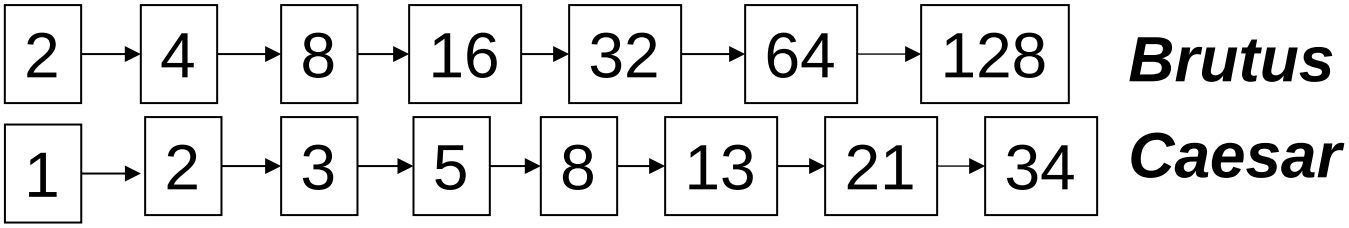
# Φτιάξαμε το ευρετήριο, τώρα;

- Πως επεξεργαζόμαστε μια ερώτηση;
  - Αργότερα – τι άλλου είδους ερωτήσεις

# Επεξεργασία ερωτήσεων: AND

Έστω η ερώτηση: ***Brutus*** *AND* ***Caesar***

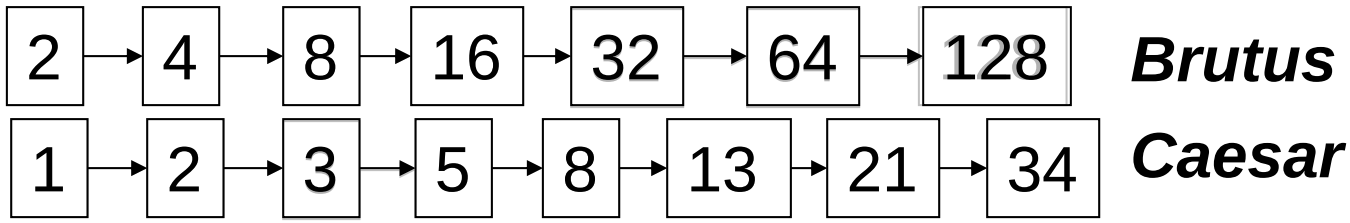
- Βρες το ***Brutus*** στο Λεξικό
  - Ανέκτησε τις καταχωρήσεις.
- Βρες το ***Caesar*** στο Λεξικό
  - Ανέκτησε τις καταχωρήσεις.
- Υπολογισμό της τομής του – ΠΩΣ;



# Συγχώνευση (merge)



- Διέσχισε τις δύο λίστες ταυτόχρονα, σε χρόνο γραμμικό (linear) στο συνολικό αριθμό των καταχωρήσεων




Αν τα μήκη των λιστών είναι *x* και *y*, η συγχώνευση παίρνει O(*x*+*y*) λειτουργίες.
Σημαντικό: οι καταχωρήσεις πρέπει να είναι διατεταγμένες με βάση το docID.

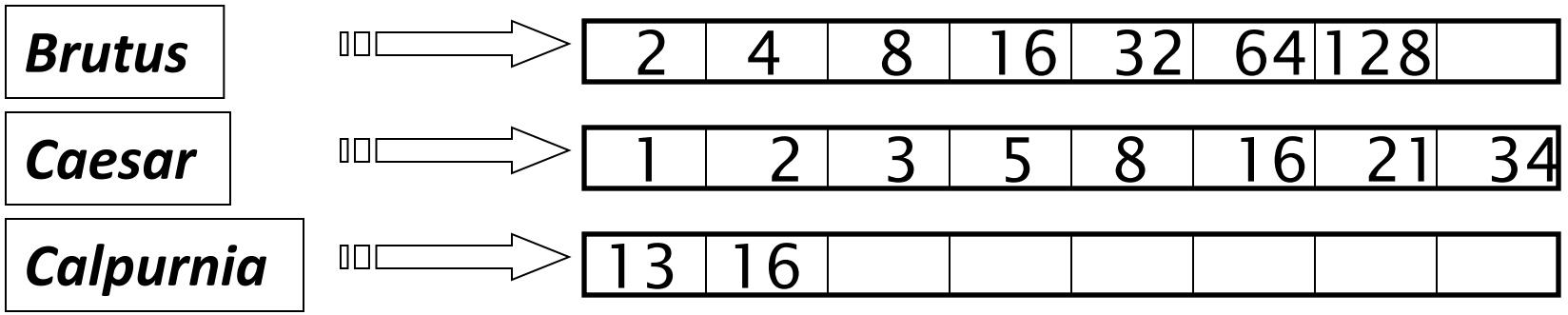
# Ο αλγόριθμος συγχώνευσης

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

# Βελτιστοποίηση ερωτήματος

- Ποια είναι *βέλτιστη σειρά* για την επεξεργασία ενός ερωτήματος;

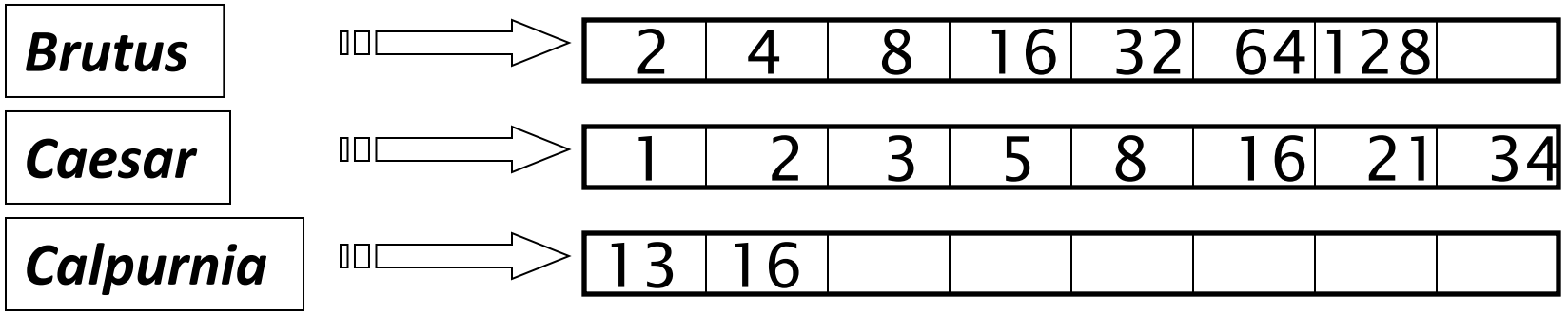
- Έστω μια ερώτηση που είναι το *AND* *n* όρων.
- Για καθέναν από τους *n* όρους, βρες τις καταχωρήσεις του και εκτέλεσε το *AND* σε όλες.




Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

# Βελτιστοποίηση ερωτήματος

- Επεξεργασία με αύξουσα συχνότητα:
  - *Ξεκίνησε με το μικρότερο σύνολο και συνέχισε μειώνοντας και άλλο το αποτέλεσμα*


Εκτέλεση του ερωτήματος ως (***Calpurnia*** *AND* ***Brutus)*** *AND* ***Caesar***.

# Βελτιστοποίηση ερωτήματος

Π.χ., *(**madding** OR **crowd**) AND (**ignoble** OR **strife**)*

- Βρες τη συχνότητα εγγράφου για όλους τους όρους.
- Εκτίμησε το μέγεθος κάθε *OR* (συντηρητικά: ως το άθροισμα των συχνοτήτων εγγράφου).
- Επεξεργασία του ερωτήματος κατά αύξουσα σειρά κάθε όρου.

# Βελτιστοποίηση ερωτήματος

((A and B) and C) and D

- Κρατάμε *το ενδιάμεσο αποτέλεσμα στη μνήμη* και διαβάζουμε τη άλλη λίστα από το δίσκο
- Αρχικά, ενδιάμεσο αποτέλεσμα = A

Όταν πολλοί μεγάλες λίστες, εναλλακτικές για τον υπολογισμό τομής

- χρησιμοποιώντας δυαδική αναζήτηση στη μεγάλη λίστα (λογαριθμικός χρόνος)
- αποθήκευση μεγάλης λίστας ως hashtable (σταθερά)

# Boolean ερωτήματα: Ακριβές ταίριασμα (Exact match)

- Το Boolean μοντέλο ανάκτησης απαντά ερωτήματα που είναι Boolean εκφράσεις:
  - Χρήση *AND, OR* και *NOT* για το συνδυασμό όρων
    - Θεωρούν κάθε έγγραφο ως ένα **σύνολο** όρων
    - Η αναζήτηση είναι ακριβής (precise): *ένα έγγραφο είτε ικανοποιεί τη συνθήκη είτε όχι.*
  - Ίσως, το απλούστερο μοντέλο
- Το βασικό μοντέλο σε εμπορικά συστήματα για 3 δεκαετίες (πριν τον web).
- Πολλά συστήματα ακόμα Boolean:
  - Email, library catalog, Mac OS X Spotlight

Η Google χρησιμοποιεί το Boolean μοντέλο ?

# Τι είδαμε σήμερα

1- Βασικές έννοιες

2- Είδαμε ένα απλό σύστημα ανάκτησης πληροφορίας βασισμένο στο Boolean μοντέλο

α- ανάλυση εγγράφου
β- κατασκευάζουμε το ανεστραμμένο ευρετήριο
γ- το χρησιμοποιούμε για να απαντάμε σε ερωτήματα

# ΤΕΛΟΣ 1ου Κεφαλαίου

Ερωτήσεις?

*Χρησιμοποιήθηκε κάποιο υλικό των:*

- ✓ *Pandu Nayak and Prabhakar Raghavan, CS276:Information Retrieval and Web Search (Stanford)*
- ✓ *Απόστολου Ν. Παπαδόπουλου , Ανάκτηση Πληροφορίας (Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο)*