ΜΥΕ003: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Ανάκτηση Boole

Ακαδημαϊκό Έτος 2023-2024

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);

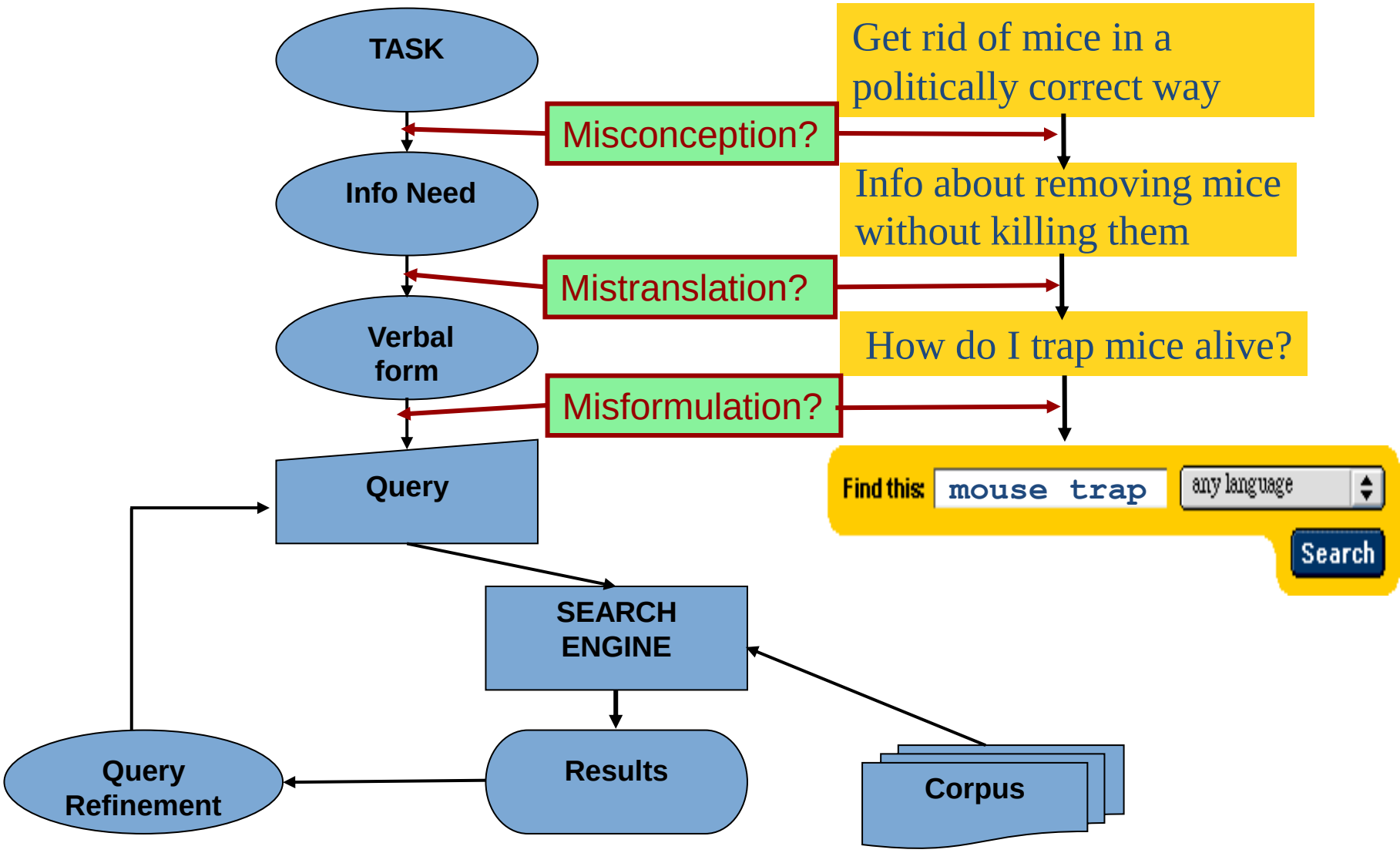
Βασικές Έννοιες

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)



Εφαρμογές

- Μηχανές Αναζήτησης (search engines):

στο web/διαδίκτυο, δισεκατομμύρια έγγραφα σε εκατομμύρια υπολογιστές.

Συλλογή εγγράφων, κλίμακα, διάταξη αποτελεσμάτων, ..

- Προσωπική ανάκτηση πληροφορίας

(στον προσωπικό υπολογιστή (desktop search, email, κλπ)

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

- Σε επίπεδο επιχείρησης, οργανισμού (enterprise, institutional)
- Αναζήτηση ειδικού σκοπού (domain-specific search) – Ψηφιακές βιβλιοθήκες, Δικηγόροι, Γιατροί, κλπ

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!

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

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

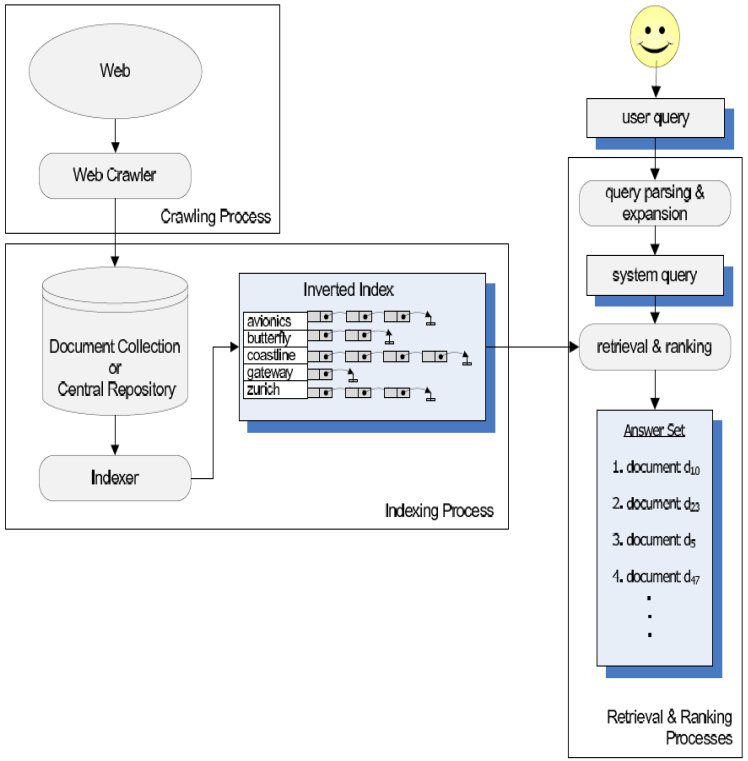
παράδειγμα Συλλογή 1000 έγγραφα
υπάρχουν 20 συναφή έγγραφα με το ερώτημα
το ΣΑΠ μας επιστρέφει 30 έγγραφα (17 συναφή
13 μη συναφή

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

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

Architecture of the IR System

- High level software architecture of an IR system



Διαδικαστικά

- Βαθμολογία (μπορεί να αλλάξει):
  - Εργασία (έως 2 άτομα) – σε φάσεις: 50%
    - Προαιρετική εργασία – ανεξάρτητη μελέτη ML μοντέλων (20%) ή σύνδεση τους με τη μηχανή αναζήτησης
  - Τελικό Διαγώνισμα: 50% (αν όχι την προαιρετική εργασία)
    30% (αν την προαιρετική εργασία)

Ανακοίνωση Εργασίας	26/3
Παράδοση 1ης Φάσης	16/4
Παράδοση Τελικής Φάσης	21/5
Εξέταση Εργασίας	εβδομάδα 27/5

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

Boolean Ανάκτηση

Boolean μοντέλο

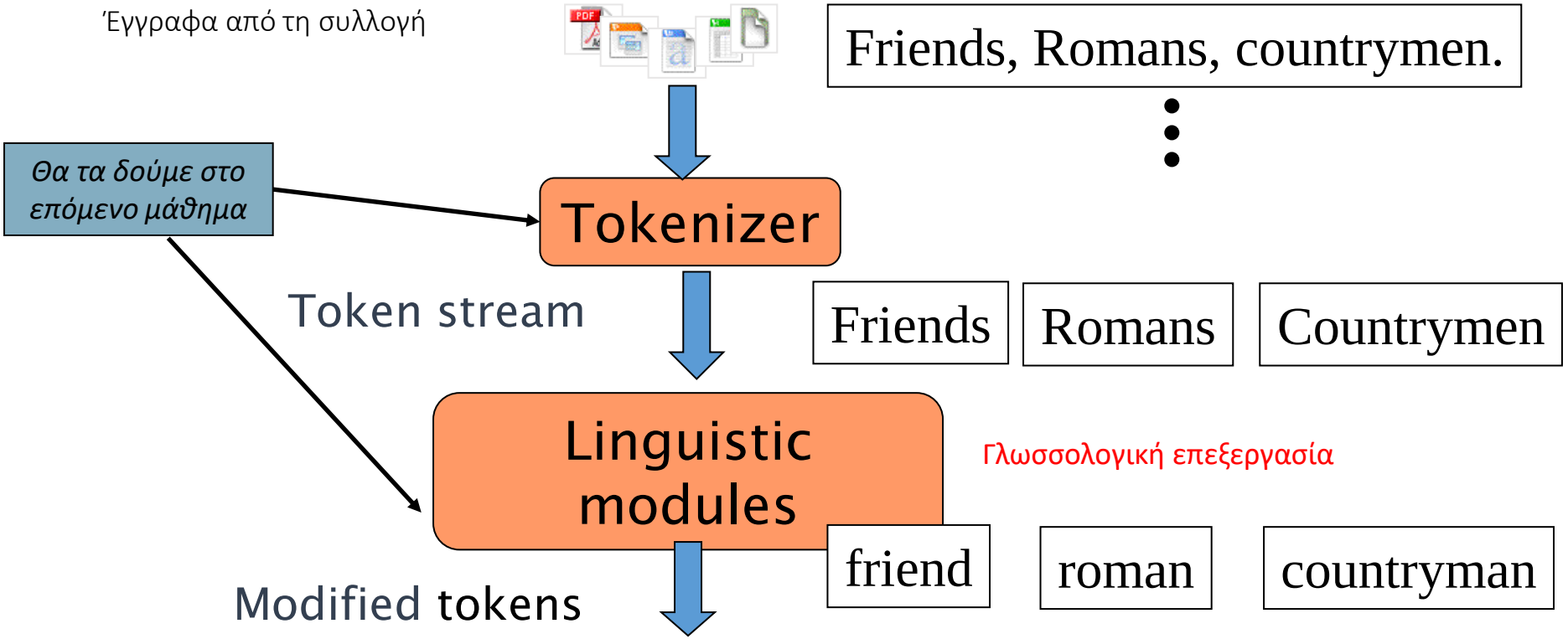
- Επιστρέφονται ως απάντηση όλα τα κείμενα που ικανοποιούν το ερώτημα *χωρίς διάταξη*
  - *Δυαδική συνάφεια (συναφές, μη συναφές)*
- Κείμενο ως σύνολο όρων
- Οι χρήστες διατυπώνουν ερωτήματα με τη μορφή *Boolean εκφράσεων*, δηλαδή όρων συνδυασμένων με AND, OR και NOT
- Συναφές αν περιέχει τους όρους

Αδόμητα δεδομένα το 1680

Shakespeare's Collected Works

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

Ανάλυση



Όροι (terms)

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

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.

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.

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

a → d_1 | d_2 | d_3 | d_4

b → d_1 | d_2 | d_3

c → d_1

d → d_2 | d_5

e → d_4

f → d_3 | d_5

	d_1	d_2	d_3	d_4	d_5
a	1	1	1	1	0
b	1	1	1	0	0
c	1	0	0	0	0
d	0	1	0	0	1
e	0	0	0	1	0
f	0	0	1	0	1

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

στηλών 1.000.000 # εγ.
γραμμών 500.000 # όρων

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

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

1.000.000 × 1.000

Ποσοστό των 1: 0.2%

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;

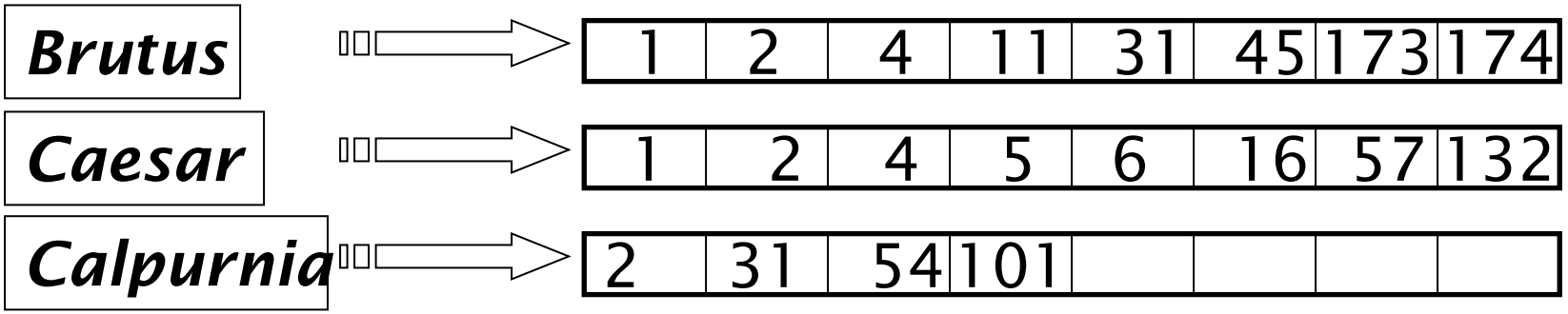
Αντεστραμμένο ευρετήριο ή αρχείο (Inverted index/file)

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

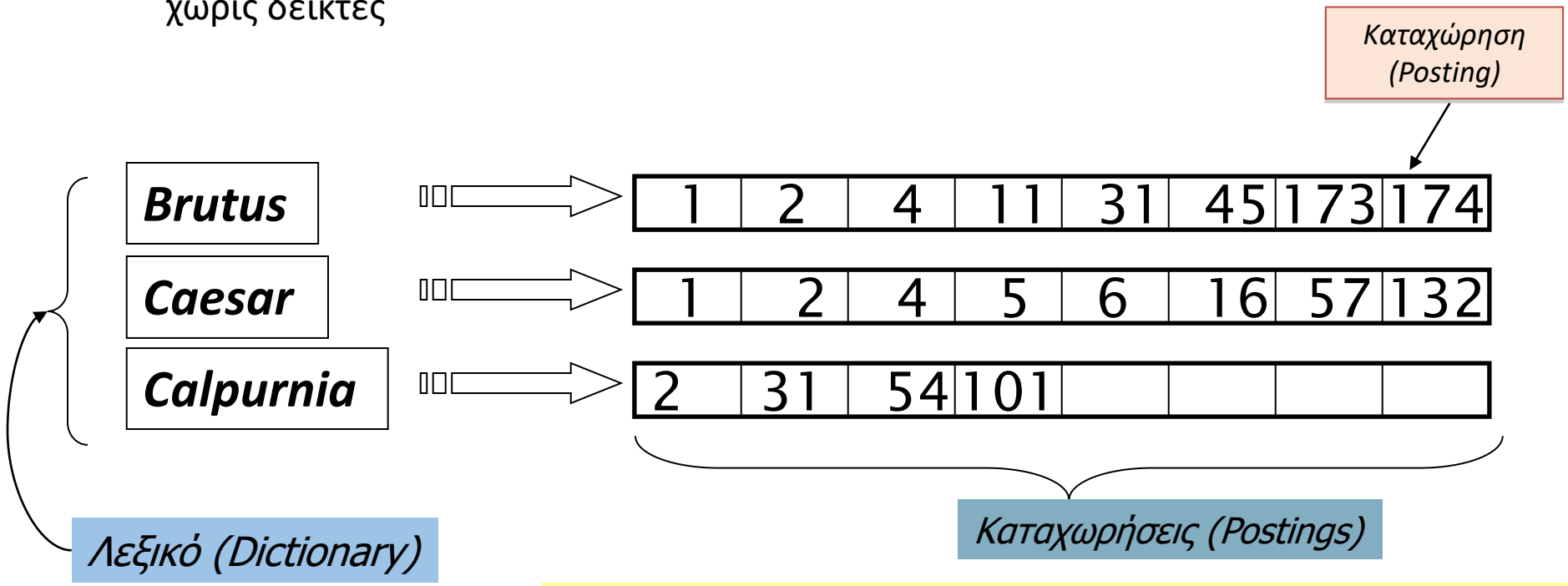
- Μπορούμε να χρησιμοποιήσουμε σταθερού μεγέθους arrays για αυτό?



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 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

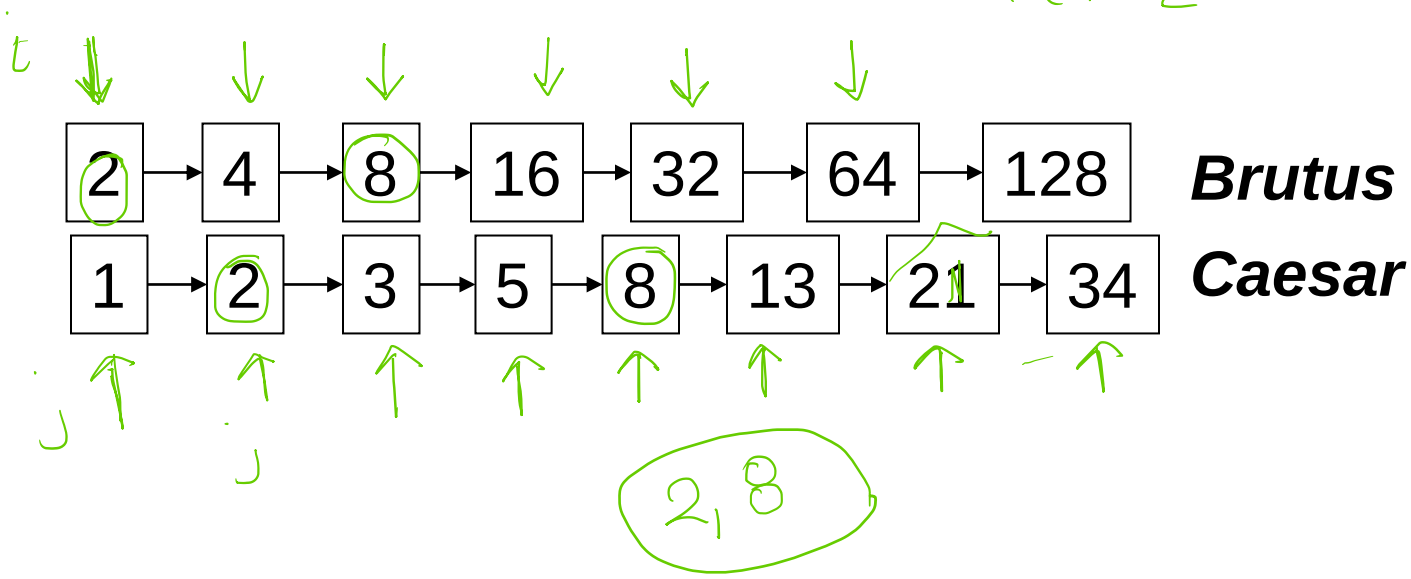
Φτιάξαμε το ευρετήριο, τώρα;

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

Επεξεργασία ερωτήσεων: AND

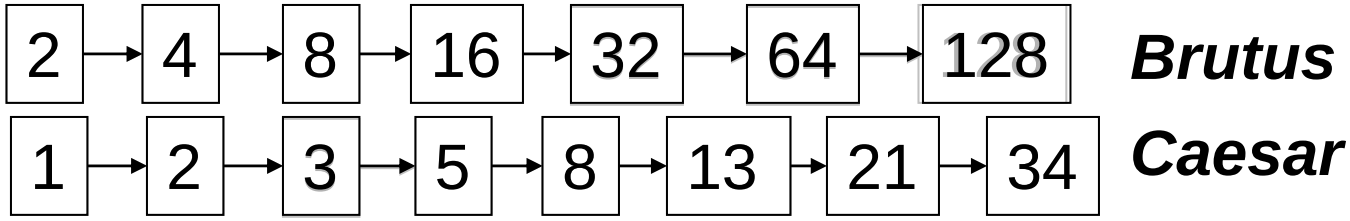
Έστω η ερώτηση: ***Brutus*** *AND* ***Caesar***

- 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.

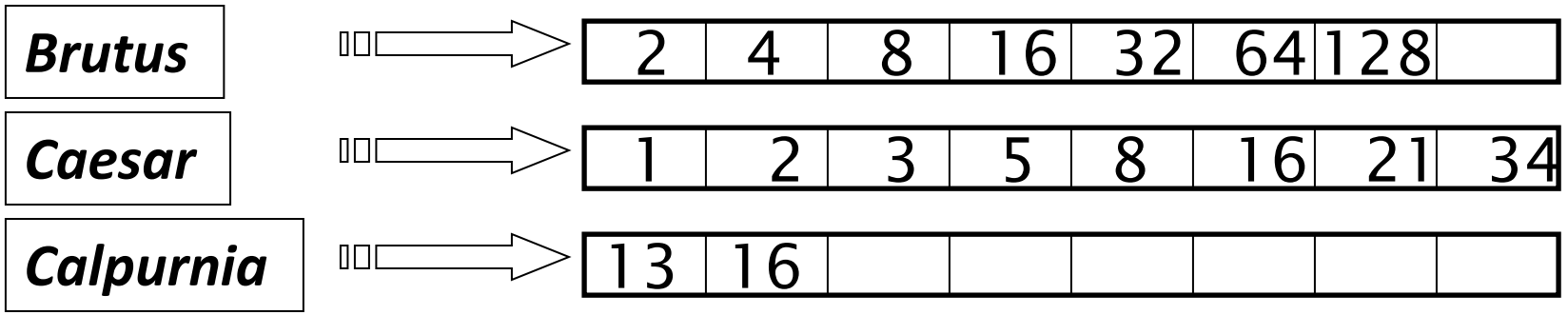
Ο αλγόριθμος συγχώνευσης

```
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.

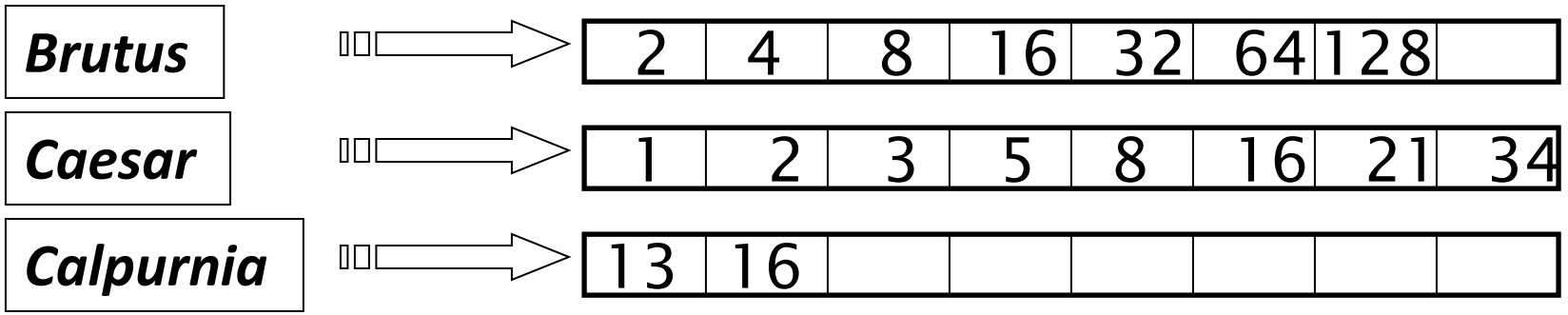


Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

Βελτιστοποίηση ερωτήματος

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

Χρήση της συχνότητας εγγράφου στο λεξικό



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)*