Introduction to

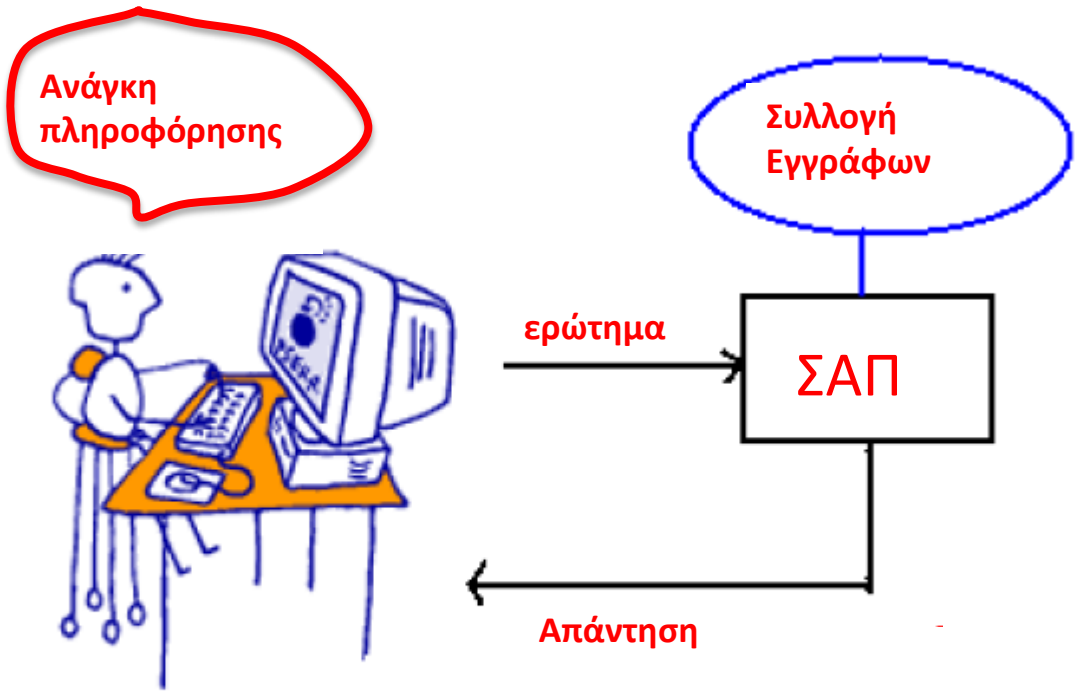
Information Retrieval

ΜΥΕ003-ΠΛΕ70: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Διάλεξη 1: Εισαγωγή. Ανάκτηση Boole

Τι είναι η «Ανάκτηση Πληροφορίας»;



Information Retrieval – Ανάκτηση Πληροφορίας

Γιατί να μας ενδιαφέρει;

Παλιότερα,

Βιβλιοθηκονόμους, βοηθούς νομικών επαγγελμάτων κλπ;

Γιατί να μας ενδιαφέρει;

Σήμερα - Μερικές εφαρμογές

Web search

Google

1 | Google

1 - eBizMBA Rank | 1,100,000,000 - Estimated Unique Monthly Visitors | 1 - Compete Rank | 1 - Quantcast Rank | 1 - Alexa Rank.
The Most Popular Search Engines | Updated 2/15/2014 | eBizMBA

bing

2 | Bing

14 - eBizMBA Rank | 285,000,000 - Estimated Unique Monthly Visitors | 5 - Compete Rank | 19 - Quantcast Rank | 19 - Alexa Rank.
The Most Popular Search Engines | Updated 2/15/2014 | eBizMBA

YAHOO!

3 | Yahoo! Search

18 - eBizMBA Rank | 250,000,000 - Estimated Unique Monthly Visitors | *8* - Compete Rank | *28* - Quantcast Rank | NA - Alexa Rank.
The Most Popular Search Engines | Updated 2/15/2014 | eBizMBA

Ask

4 | Ask

24 - eBizMBA Rank | 145,000,000 - Estimated Unique Monthly Visitors | 12 - Compete Rank | 24 - Quantcast Rank | 37 - Alexa Rank.
The Most Popular Search Engines | Updated 2/15/2014 | eBizMBA

Aol.

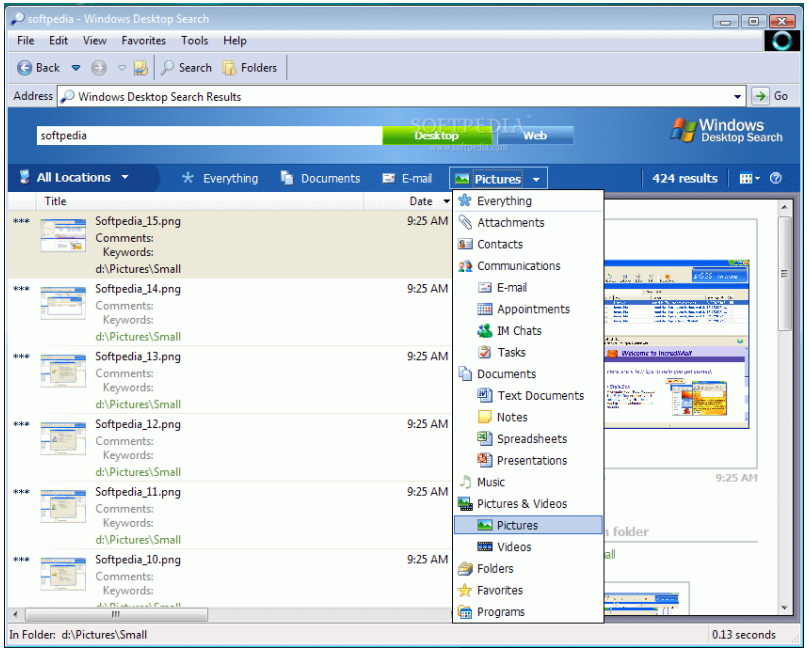
5 | Aol Search

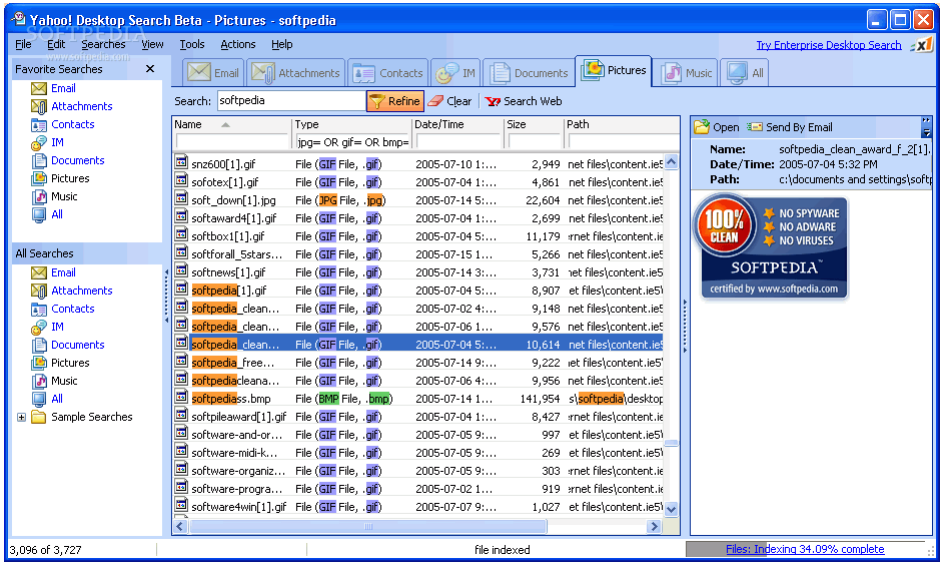
245 - eBizMBA Rank | 75,000,000 - Estimated Unique Monthly Visitors | *250* - Compete Rank | *240* - Quantcast Rank | NA - Alexa Rank.
The Most Popular Search Engines | Updated 2/15/2014 | eBizMBA

Γιατί να μας ενδιαφέρει;

Σήμερα - Μερικές εφαρμογές

Desktop search





Γιατί να μας ενδιαφέρει;

Σήμερα - Μερικές εφαρμογές

- Social search
- Digital libraries
- Enterprise search
- Domain specific search: Legal information retrieval

Διαφορετικές απαιτήσεις ανάλογα με την εφαρμογή

Σε διαφορετική κλίμακα!

- Στο web/διαδίκτυο

  Δισεκατομμύρια έγγραφα σε εκατομμύρια υπολογιστές. Θέματα?

- Προσωπική ανάκτηση πληροφορίας

  (στον προσωπικό υπολογιστή, email, κλπ) Θέματα?

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

Ανάκτηση Πληροφορίας (ορισμός)

Ανάκτηση Πληροφορίας (Information Retrieval) - (IR)

- είναι η εύρεση υλικού κυρίως εγγράφων (**documents**) αδόμητης φύσης(*) (unstructured) που συνήθως έχουν τη μορφή κειμένου (**text**)
- το οποίο ικανοποιεί μια ανάγκη πληροφόρησης (**information need**)
- από μεγάλες συλλογές (συνήθως αποθηκευμένες σε υπολογιστές)

(*) 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

Ανάκτηση Πληροφορίας vs Βάσεις Δεδομένων

Δομημένα δεδομένα

Ακολουθούν κάποιο σχήμα και είναι αποθηκευμένα με βάση κάποιο μοντέλο

Ερώτημα SQL

SELECT όνομα
FROM πλανήτες
WHERE δορυφόροι = 0
  OR δορυφόροι = 1
  OR δορυφόροι = 2

κωδικός	όνομα	διάμετρος	δορυφόροι
1	Ερμής	4880	0
2	Αφροδίτη	12103.6	0
3	Γη	12756.3	1
4	Άρης	6794	2
5	Δίας	142984	63
6	Κρόνος	120536	34
7	Ουρανός	51118	21
8	Ποσειδώνας	49532	13
9	Πλούτωνας	2274	3

Το ερώτημα είναι σαφές, προσδιορίζει επακριβώς τη συνθήκη που πρέπει να ικανοποιεί κάθε αποτέλεσμα που εμφανίζεται στην έξοδο.

Ανάκτηση Πληροφορίας vs Βάσεις Δεδομένων

Συλλογή εγγράφων

d1 : Ο κομήτης του Χάλεϋ μας επισκέπτεται περίπου κάθε εβδομήντα έξι χρόνια.
d2 : Ο κομήτης του Χάλεϋ πήρε το όνομά του από τον αστρονόμο Έντμοντ Χάλεϋ.
d3 : Ένας κομήτης διαγράφει ελλειπτική τροχιά.
d4 : Ο πλανήτης Άρης έχει δύο φυσικούς δορυφόρους, το Δείμο και το Φόβο.
d5 : Ο πλανήτης Δίας έχει 63 γνωστούς φυσικούς δορυφόρους.
d6 : Ένας κομήτης έχει μικρότερη διάμετρο από ότι ένας πλανήτης.
d7 : Ο Άρης είναι ένας πλανήτης του ηλιακού μας συστήματος.
…

Πληροφοριακή ανάγκη: πληροφορίες για τον κομήτη του Χάλεϋ
Ερώτημα: Χάλεϋ

Διαισθητικά αντιλαμβανόμαστε ότι τα έγγραφα *d*1 και *d*2 σχετίζονται περισσότερο με το ερώτημα από ότι τα υπόλοιπα έγγραφα.

Ανάκτηση Πληροφορίας vs Βάσεις Δεδομένων

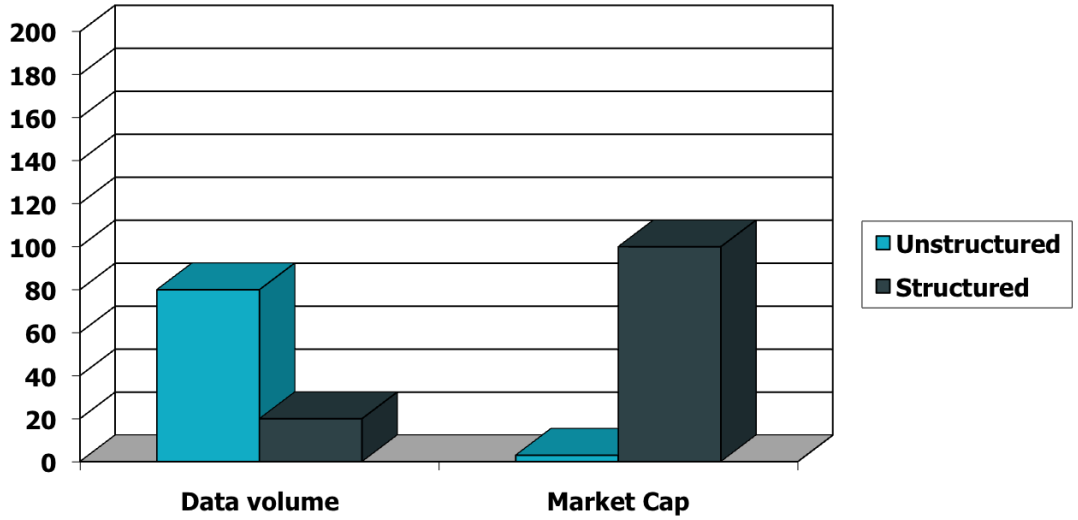
Χαρακτηριστικό	ΣΔΒΔ	ΣΑΠ
είδος δεδομένων	δομημένα	αδόμητα, ημι-δομημένα
τύπος δεδομένων	αριθμητικά, αλφαριθμητικά	έγγραφα (κειμένου)
γλώσσα ερωτημάτων	SQL	φυσική γλώσσα, λέξεις κλειδιά (keywords)
ερώτημα	σαφές	ασαφές
αποτελέσματα	χωρίς βαθμολόγηση	βαθμολογημένα

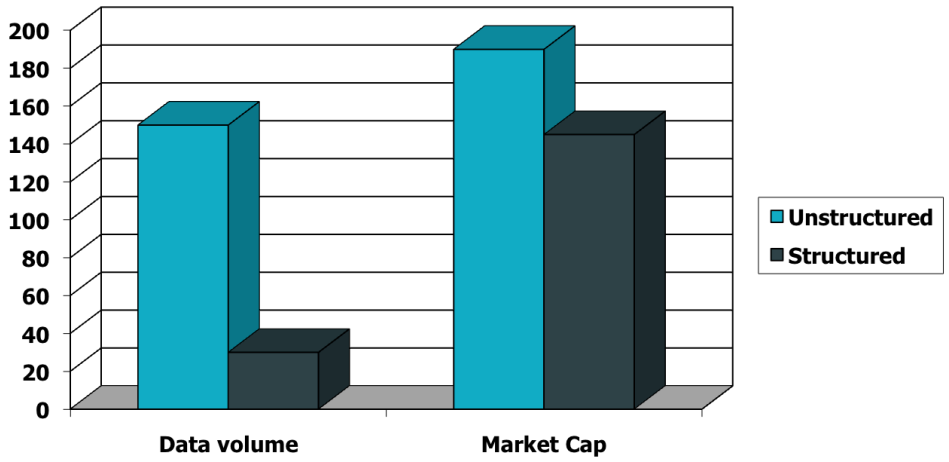
Ημιδομημένα δεδομένα

- 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 ζώνες

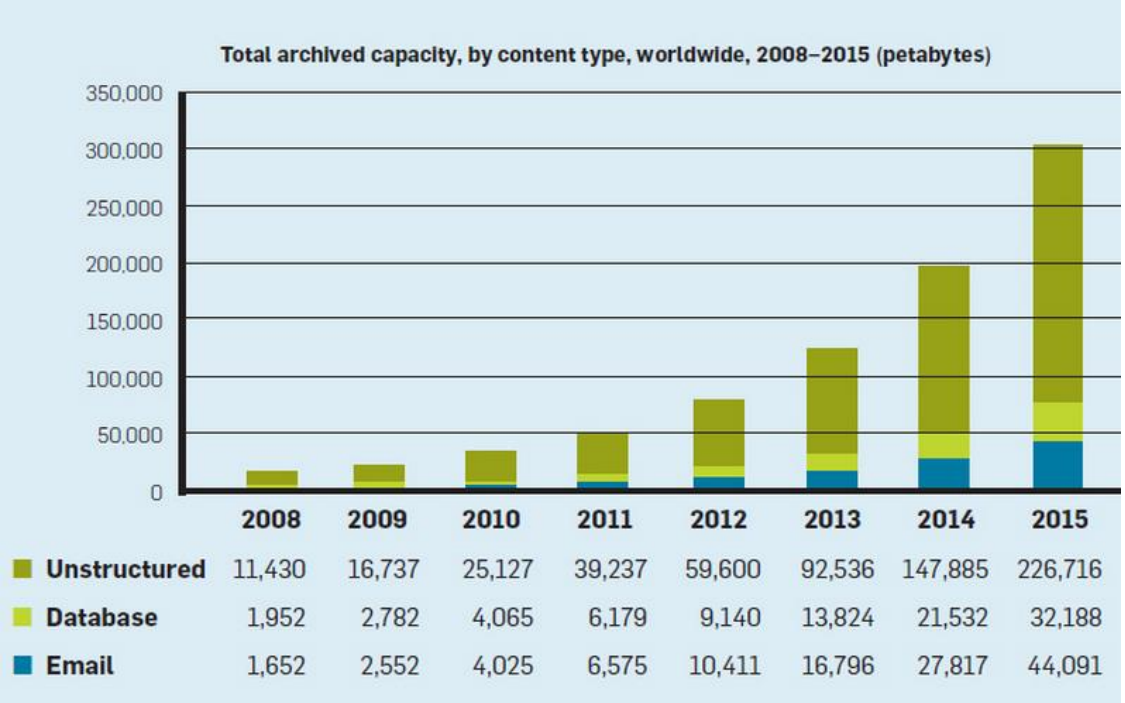
… και βέβαια υπάρχει πάντα η γλωσσική δομή

Αδόμητα (κείμενο) vs. Δομημένα (βάσεις δεδομένων) δεδομένα το 1996





Αδόμητα (κείμενο) vs. Δομημένα (βάσεις δεδομένων) δεδομένα σήμερα?



	2008	2009	2010	2011	2012	2013	2014	2015
Unstructured	11,430	16,737	25,127	39,237	59,600	92,536	147,885	226,716
Database	1,952	2,782	4,065	6,179	9,140	13,824	21,532	32,188
Email	1,652	2,552	4,025	6,575	10,411	16,796	27,817	44,091

User generated content (social networks, blogs, etc) Example: *Facebook search*

http://bhavnaober.blogspot.gr/2015/02/data-warehouse-current-scenario.html

Όχι μόνο ανάκτηση!

- Κατηγοριοποίηση (classification)

  Τοποθέτηση εγγράφων στη σωστή κατηγορία (Παράδειγμα: email spam)

- Συσταδοποίηση (clustering)

  Ομαδοποίηση σχετικών εγγράφων και περίληψη

- «Φιλτράρισμα»

  Με βάση κριτήρια σχετικότητας

- Συστάσεις (recommendations)
- Κριτικές (reviews)

Τι άλλο θα δούμε σήμερα;

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

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

2. Λίγα διαδικαστικά

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):
  - Project (έως 2 άτομα): 50%
  - Τελικό Διαγώνισμα: 50%

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*** ***(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?
  - Αργό (για μεγάλες συλλογές)
  - Grep line-oriented, ανάκτηση πληροφορίας document-oriented
  - *NOT* ***Calpurnia*** δεν είναι εύκολο
  - Επιπρόσθετες 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 …

- *Το 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 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.

Βασικές Έννοιες

- Συλλογή (Collection - corpus): Σταθερό σύνολο από έγγραφα
- Στόχος: Ανάκτηση των εγγράφων που περιέχουν πληροφορία που είναι σχετική/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

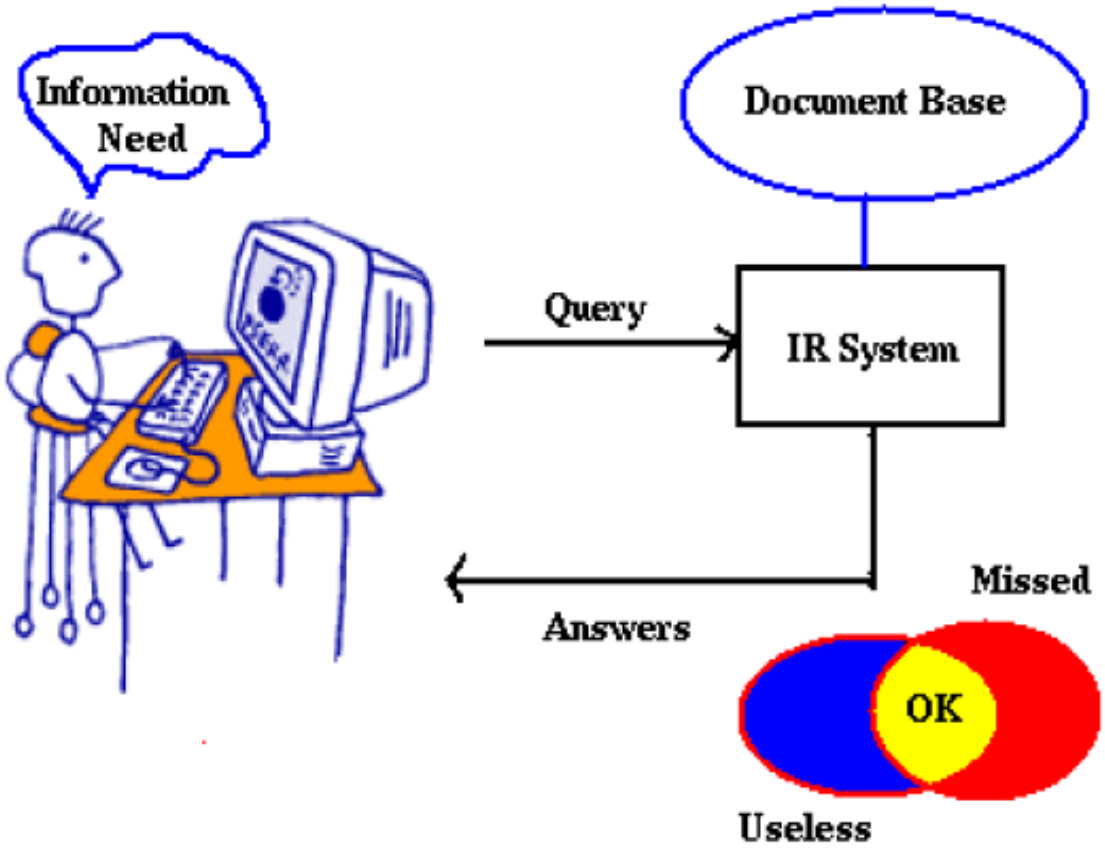
Αποτελεσματικότητα

Αποτελεσματικότητα (effectiveness): Πόσο καλά είναι τα έγγραφα που ανακτήθηκαν;

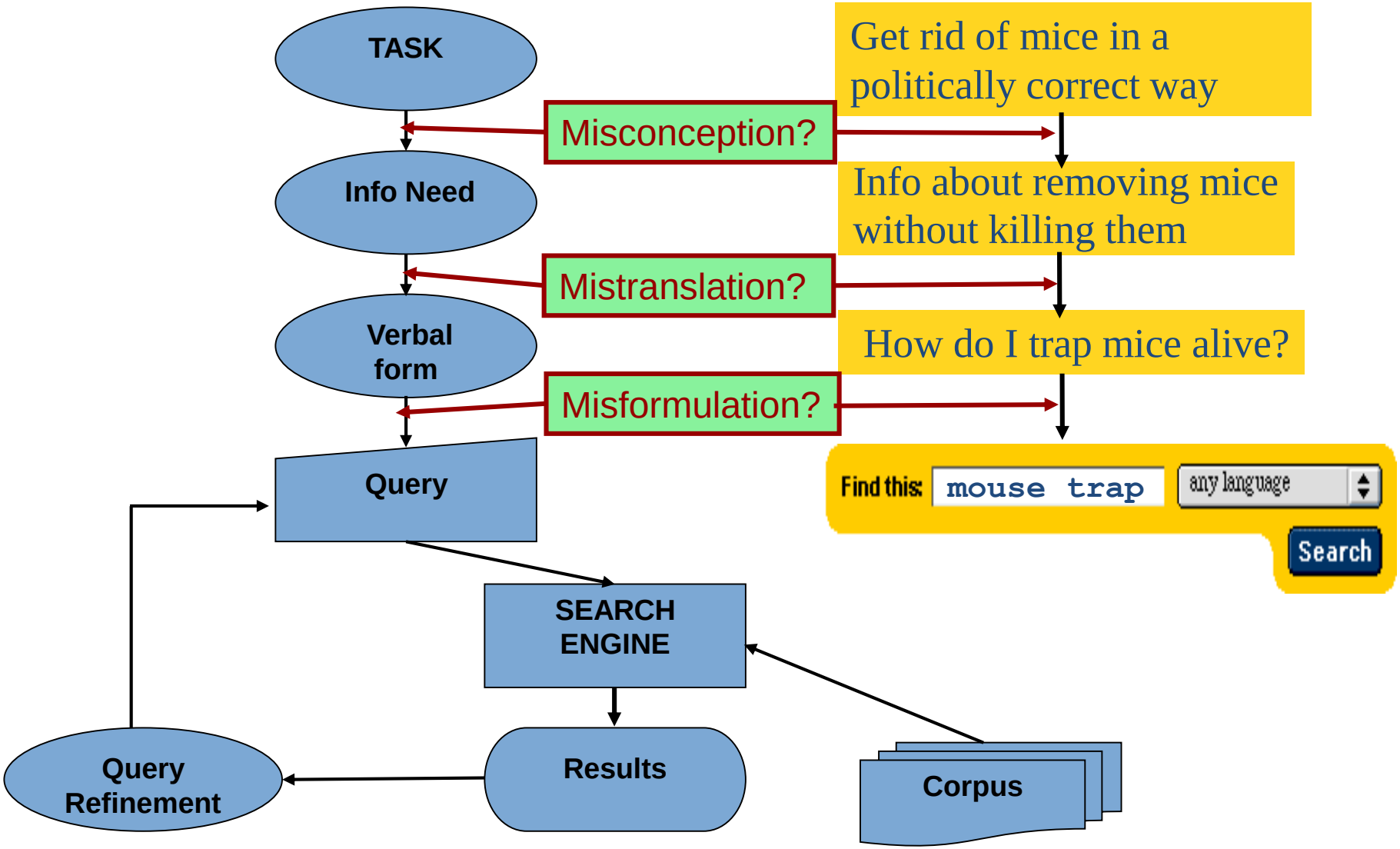
- *Ακρίβεια (Precision)*: Το ποσοστό των εγγράφων που ανακτήθηκαν που είναι συναφή με την ανάγκη πληροφόρησης του χρήστη
- *Ανάκληση (Recall)* : Το ποσοστό των συναφών με την ανάγκη πληροφόρησης του χρήστη εγγράφων της συλλογής που ανακτήθηκαν από το σύστημα
  - Περισσότερα στο μέλλον

✓ Διαφορά μεταξύ: αποτελεσματικότητας (effectiveness) και απόδοσης (efficiency)

Βασικές Έννοιες



Το κλασικό μοντέλο αναζήτησης (search model)

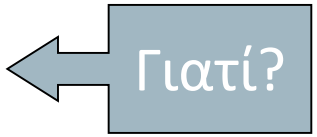


Μεγαλύτερες συλλογές

- Ας θεωρήσουμε *N* = 1 εκατομμύρια έγγραφα, το καθένα με περίπου 1000 λέξεις (~ 2-3 σελίδες βιβλίου).
- Κατά μέσο όρο 6 bytes/λέξη συμπεριλαμβανομένων κενών/συμβόλων στίξης
  - 6GB δεδομένων.
- Έστω ότι ανάμεσα τους υπάρχουν *M* = 500K διακριτοί (*distinct)* όροι.

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;
  - Καταγράφουμε μόνο τις θέσεις του 1.



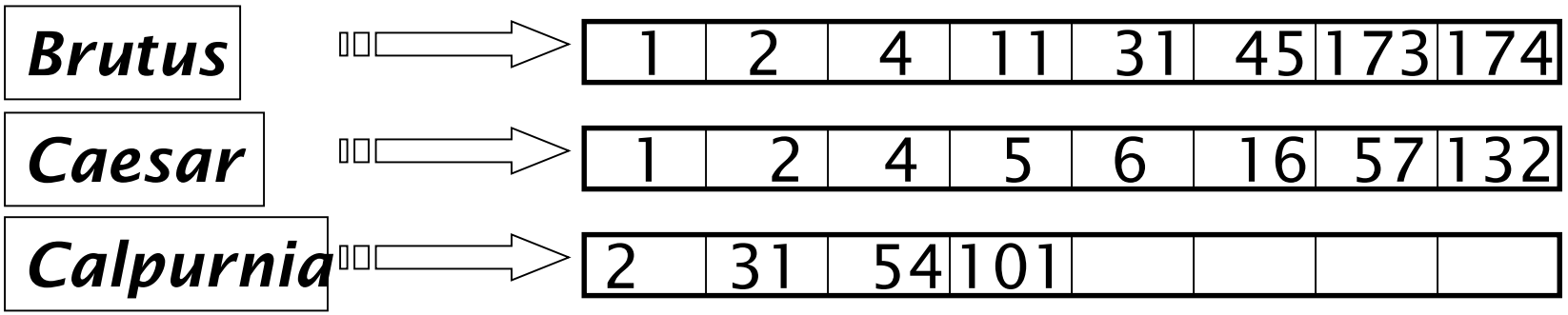
Αντεστραμμένο ευρετήριο (Inverted index)

Αντεστραμμένο ευρετήριο ή αρχείο (Inverted index/file)

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



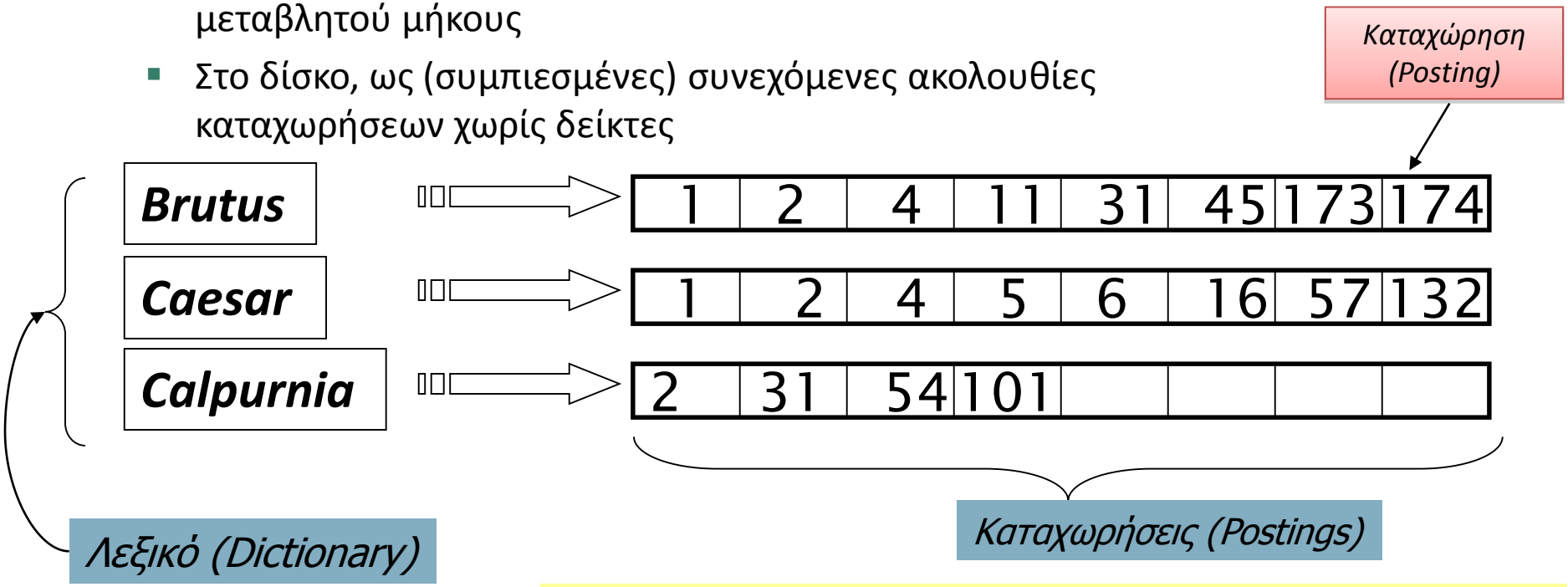
Τι γίνεται αν η λέξη ***Caesar*** προστεθεί στο έγγραφο 14?

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



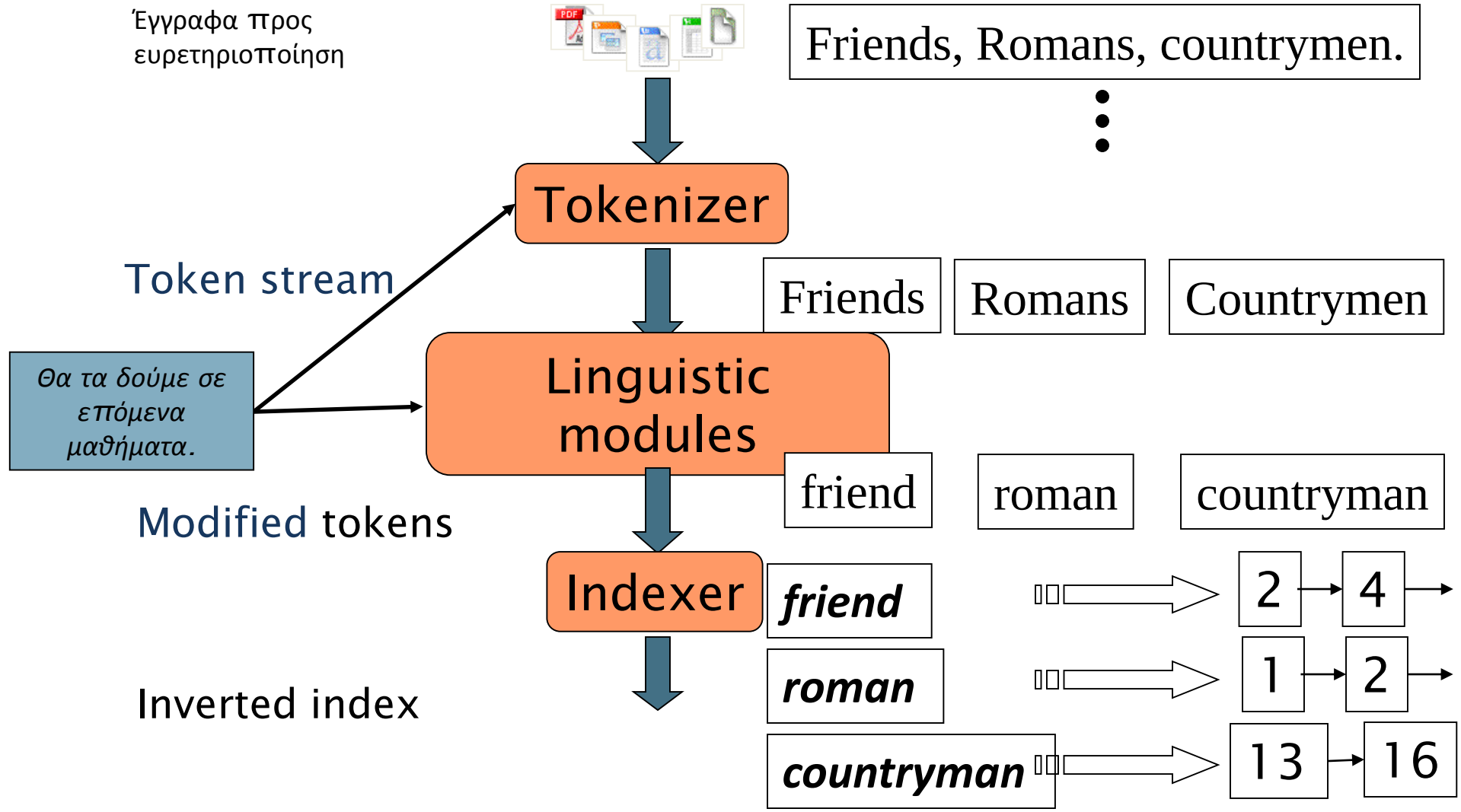
Σε διάταξη με βάση το docID (θα δούμε σε λίγο γιατί!).

Βασική Ορολογία

- Αντεστραμμένο ευρετήριο
- Λίστες καταχωρήσεων – 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

Κατασκευή του αντεστραμμένου ευρετηρίου



Βήματα του Indexer: Ακολουθία Token

- Ακολουθία από ζεύγη (Modified token, Document ID).

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Βήματα του Indexer: Ταξινόμηση (sort)

- Ταξινόμηση με βάση τους όρους
  - Και μετά το docID

Βασικό βήμα της ευρετηριοποίησης

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Βήματα του Indexer: Λεξικό & Καταχωρήσεις

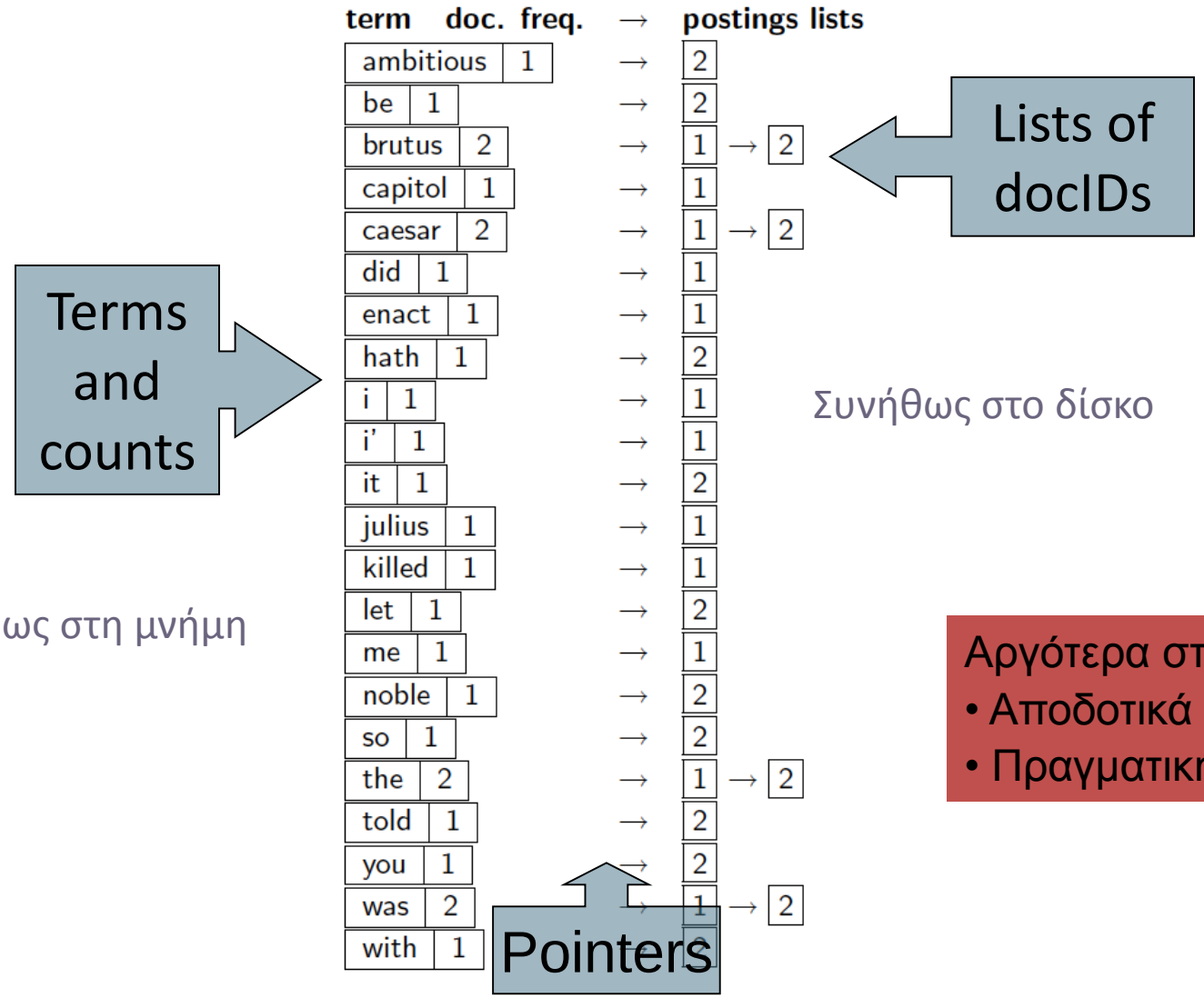
- Πολλαπλές εμφανίσεις του όρου σε ένα έγγραφο συγχωνεύονται (merged).
- Διαχωρισμός σε *λεξικό* και *καταχωρήσεις*
- Προσθέτουμε και πληροφορία για τη συχνότητα εγγράφου (doc. frequency).

Γιατί τη συχνότητα;
Επίσης, συχνότητα όρου (term frequency)

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Πόσο χώρο χρειαζόμαστε?



Συνήθως στο δίσκο

Συνήθως στη μνήμη

Αργότερα στο μάθημα:
- Αποδοτικά ευρετήρια
- Πραγματική αποθήκευση

Φτιάξαμε το ευρετήριο, τώρα;

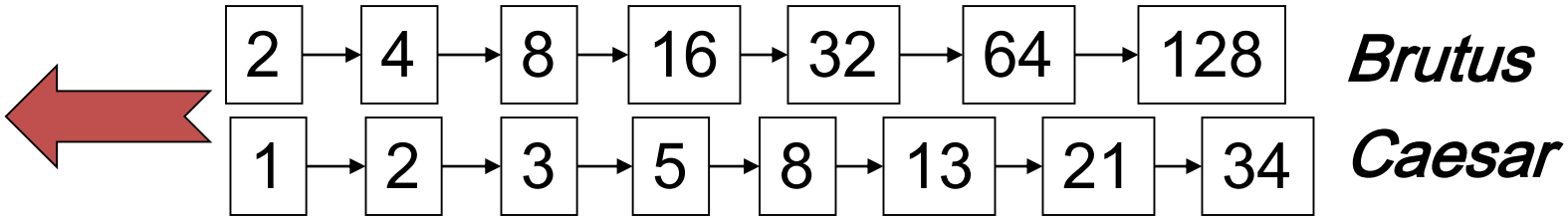
- Πως επεξεργαζόμαστε μια ερώτηση;
  - Αργότερα – τι άλλου είδους ερωτήσεις

Επεξεργασία ερωτήσεων: AND

- Έστω η ερώτηση:

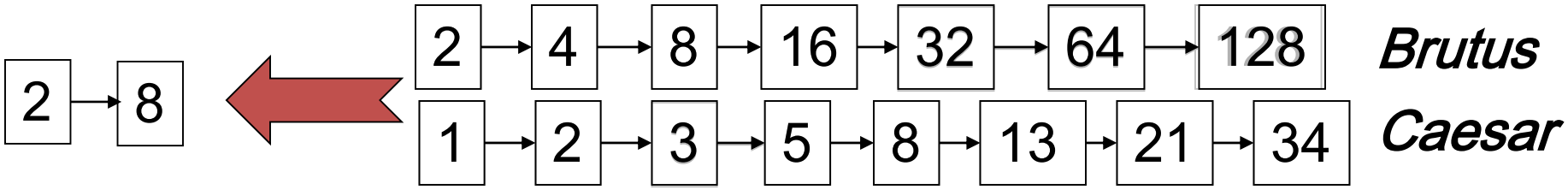
  Brutus *AND* ***Caesar***

  - Βρες το ***Brutus*** στο Λεξικό
    - Ανέκτησε τις καταχωρήσεις.
  - Βρες το ***Caesar*** στο Λεξικό
    - Ανέκτησε τις καταχωρήσεις.
  - "Merge" τις δυο καταχωρήσεις (για τον υπολογισμό της τομής):



Η 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.

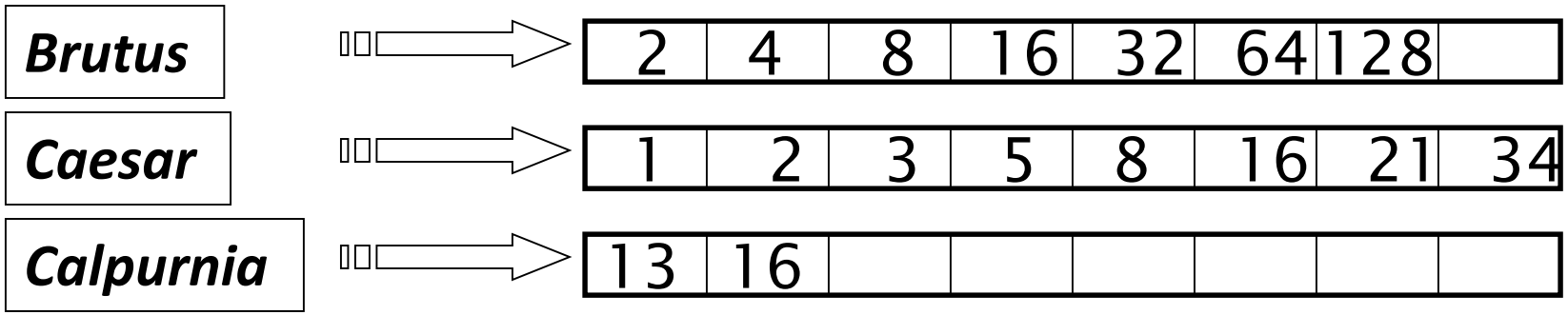
Σημαντικό: οι καταχωρήσεις πρέπει να είναι διατεταγμένες με βάση το docID.

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.

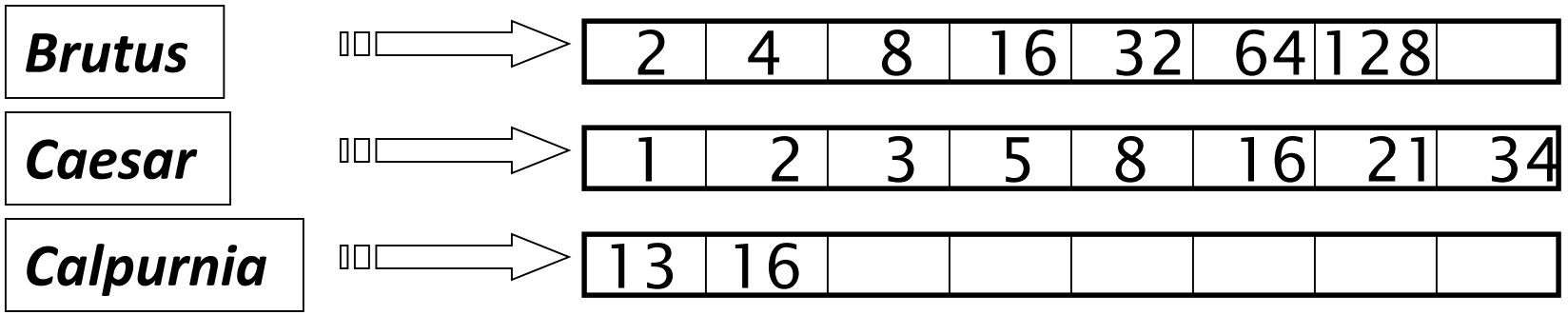


Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

Βελτιστοποίηση ερωτήματος

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

Χρήση της συχνότητας εγγράφου στο λεξικό



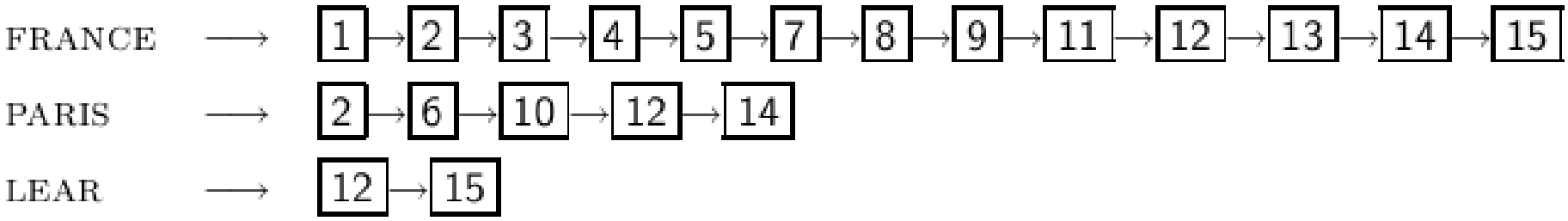
Εκτέλεση του ερωτήματος ως (***Calpurnia*** *AND* ***Brutus)*** *AND* ***Caesar***.

Βελτιστοποίηση ερωτήματος

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

- Βρες τη συχνότητας 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: παράδειγμα



((paris AND NOT france) OR lear)

((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
- Εναλλακτικά, όταν μεγάλη διαφορά στα μεγέθη των λιστών – δυαδικές αναζητήσεις στις μεγαλύτερες ή δημιουργία hashtable για τις μεγάλες λίστες

Παρατήρηση

- Δοκιμάστε το http://www.rhymezone.com/shakespeare/

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
    - Είναι ακριβές (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 ?

Παράδειγμα: WestLaw http://www.westlaw.com/

- Μεγάλο εμπορικό (συνδρομές επί πληρωμή) σύστημα
- Αναζήτηση σε νομικά κείμενα (άρχισε το1975, η διάταξη προστέθηκε το 1992)
- Δεκάδες terabytes δεδομένων, 700.000 χρήστες
- Η πλειοψηφία των χρηστών *ακόμα* χρησιμοποιεί Boolean ερωτήματα

Παράδειγμα: WestLaw http://www.westlaw.com/

- Παράδειγμα:
  - *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:
  - *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!

Example: WestLaw http://www.westlaw.com/

- Ακόμα ένα παράδειγμα:
  - Requirements for disabled people to be able to access a workplace
  - disabl! /p access! /s work-site work-place (employment /3 place)
- SPACE σημαίνει διάζευξη (disjunction)
- ***Μακροσκελή, επακριβή ερωτήματα, τελεστές εγγύτητας (proximity operators), διατυπωμένα σταδιακά (διαφορά από web search)***
- *Boolean αναζήτηση χρησιμοποιείται ακόμα από πολλούς επαγγελματίες*
  - Ξέρεις ακριβώς τι παίρνεις ως απάντηση
- Αυτό δε σημαίνει ότι δουλεύει καλύτερα

Evidence accumulation

- 1 vs. 0 εμφάνιση ενός όρου αναζήτησης
  - 2 vs. 1 εμφανίσεις
  - 3 vs. 2 εμφανίσεις, κλπ.
  - Συχνά φαίνεται καλύτερο
- Χρειαζόμαστε και τη συχνότητα εμφάνισης του όρου στα έγγραφα

Τι άλλο πέρα της αναζήτησης όρων

- «Λάθη», wildcards, κλπ
- Φράσεις
  - ***Stanford University, Πανεπιστήμιο Ιωαννίνων***
- Γειτονικότητα (Proximity): Find ***Gates*** *NEAR* ***Microsoft***.
  - Χρειαζόμαστε ευρετήρια που να διατηρούν πληροφορία για τη θέση των όρων σε ένα έγγραφο
- Ζώνες σε έγγραφα: Find documents with (*author =* ***Ullman***) *AND* (text contains ***automata***).

Καταταγμένη (Ranked) αναζήτηση

- Συχνά θέλουμε να κατατάξουμε/ομαδοποιήσουμε τα αποτελέσματα
  - Την ομοιότητα (γειτονικότητα) ενός ερωτήματος με ένα έγγραφο
  - Χρειάζεται να αποφασίσουμε αν τα έγγραφα που παρουσιάζουμε στους χρήστες είναι μονοσύνολα ή αν ένα σύνολο από έγγραφα *καλύπτει διαφορετικές απόψεις* ενός ερωτήματος.

Ποιο περίπλοκη ημιδομημένη αναζήτηση

- *Title* is about Object Oriented Programming AND *Author* something like stro*rup
  - όπου * είναι ο wild-card τελεστής
- Θέματα:
  - Πως αντιμετωπίζουμε το “about”?
  - Πως γίνεται η κατάταξη?

web

- Πέρα από τους όρους
  - συνδέσεις
- Διαφορετικοί χρήστες, ανάγκες, ερωτήματα, κείμενα
- Ιδέες από κοινωνικά δίκτυα
  - Ανάλυση συνδέσμων, clickstreams ...

- Πως δουλεύουν οι μηχανές αναζήτησης; Μπορούμε να τις βελτιώσουμε;

Ακόμα

- Διαφορετικές γλώσσες (πολύγλωσσα κείμενα)
- Απαντήσεις ερωτήσεων (Question answering)
- Περιλήψεις
- Εξόρυξη κειμένου
- …

ΤΕΛΟΣ 1ου Μαθήματος

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)*