

ΜΥΕ003-ΠΛΕ70: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Κεφάλαιο 2: Κατασκευή Λεξιλογίου Όρων.

Λίστες Καταχωρήσεων.

Ορισμός

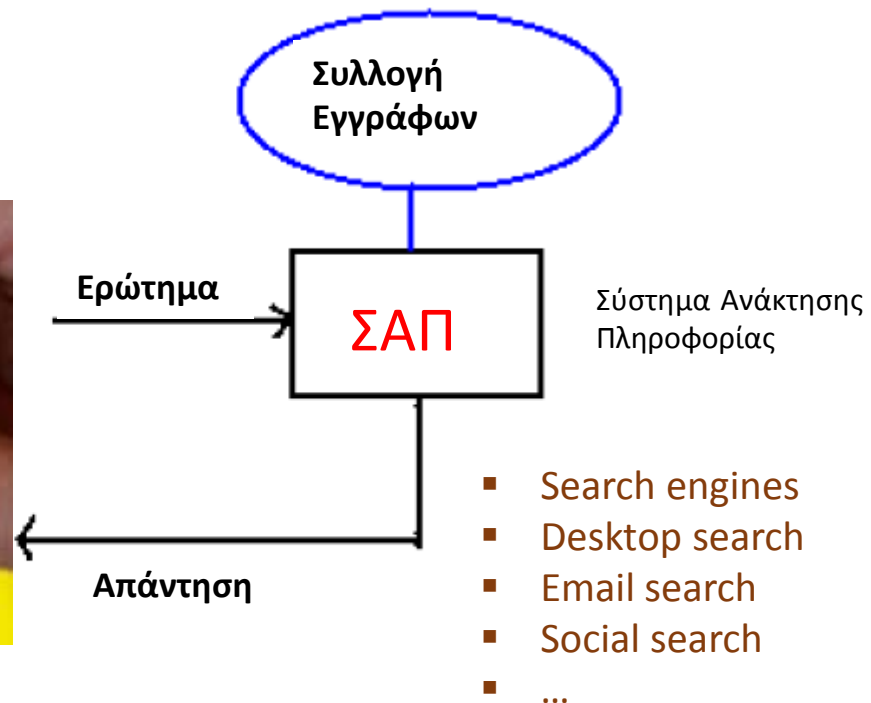
Ανάκτηση Πληροφορίας (**Information Retrieval**) - (IR)

- Μεγάλες συλλογές (**corpus**) από έγγραφα (**document**) (συνήθως κειμένου) αδόμητης φύσης (*) (**unstructured**)
- εύρεση αντικειμένων
- τα οποία ικανοποιούν μια ανάγκη πληροφόρησης (**information need**)

() όχι ακριβώς!*

Τι είναι η Ανάκτηση Πληροφορίας (Information Retrieval);

Ανάγκη
πληροφόρησης



Βασικές έννοιες

Ανάκτηση των εγγράφων που περιέχουν πληροφορία που είναι συναφής (relevant) με την ανάγκη πληροφόρησης (information need) του χρήστη

- Αποτέλεσμα σε διάταξη με βάση τη συνάφεια
- Αξιολόγηση:
 - πέρα από την απόδοση (efficiency)
 - αποτελεσματικότητα (effectiveness)
 - Ακρίβεια/Ανάκληση

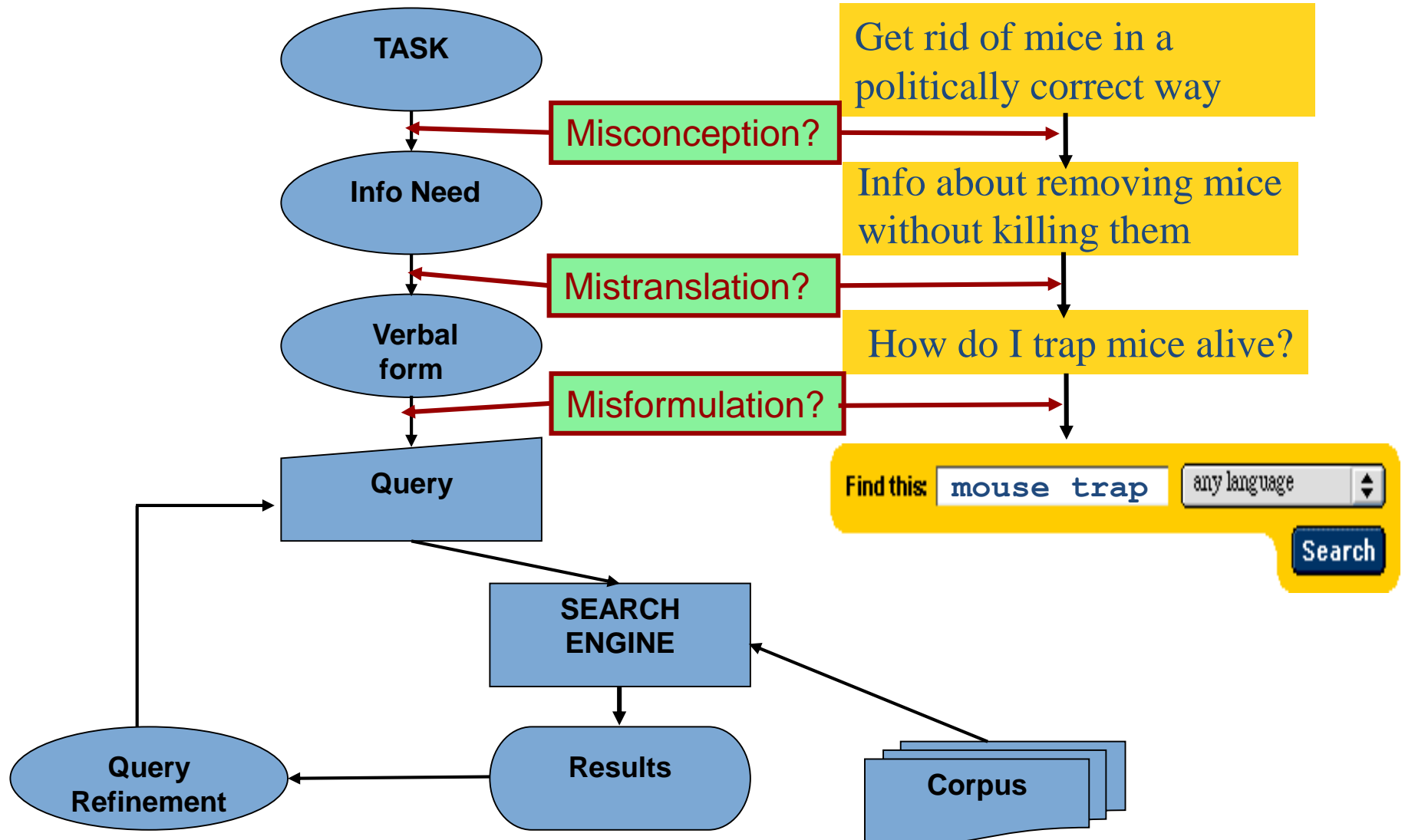
Αξιολόγηση

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

Δύο βασικές μετρικές συνάφειας (χωρίς διάταξη)

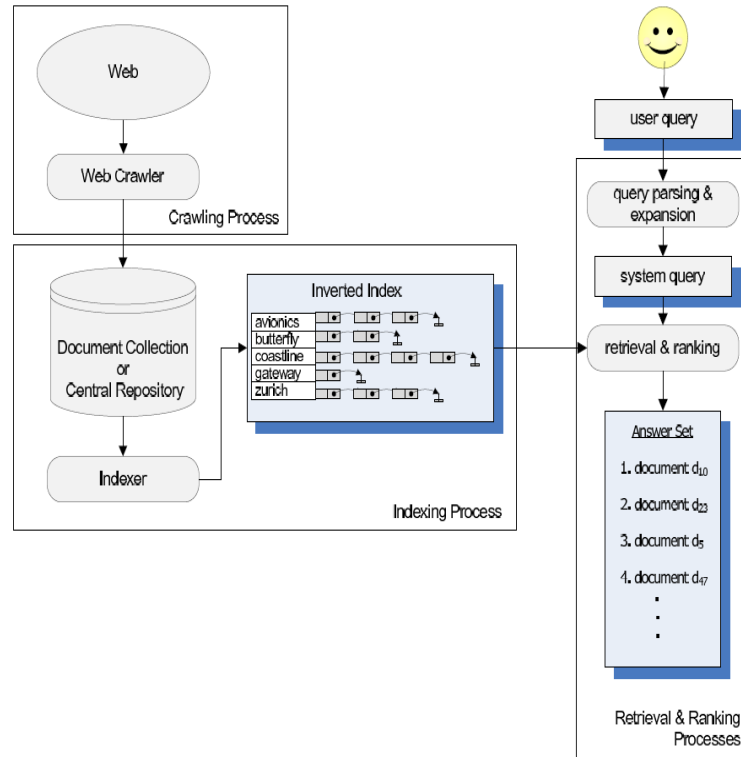
- **Ακρίβεια (Precision):** Το ποσοστό των εγγράφων που ανακτήθηκαν που είναι συναφή με την ανάγκη πληροφόρησης του χρήστη
- **Ανάκληση (Recall) :** Το ποσοστό των συναφών με την ανάγκη πληροφόρησης του χρήστη εγγράφων της συλλογής που ανακτήθηκαν από το σύστημα
 - Περισσότερα σε επόμενα μαθήματα

Το κλασικό μοντέλο αναζήτησης (search model)



Architecture of the IR System

■ High level software architecture of an IR system



Βασικά Βήματα

(προεπεξεργασία)

- Σύλλεξε τα έγγραφα
- Κατασκεύασε βοηθητικές δομές – ευρετήρια

(λειτουργία)

- Επεξεργασία ερωτήσεων

Αρχικά θα δούμε την απλούστερη μορφή:

Boolean retrieval

Boolean Ανάκτηση

Boolean μοντέλο

- Οι χρήστες διατυπώνουν ερωτήματα με τη μορφή *Boolean εκφράσεων*, δηλαδή όρων συνδυασμένων με **AND**, **OR** και **NOT**
- Κείμενο ως *σύνολο όρων*
- Επιστρέφονται ως απάντηση όλα τα κείμενα που ικανοποιούν το ερώτημα *χωρίς διάταξη*
 - *Δυαδική συνάφεια*

Παράδειγμα

- d1 a b c g e
- d2 b f a h
- d3 e d g
- d4 a c d
- d5 a b c d
- d6 a

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

Ερώτηση: a

a AND c

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

Ερώτηση: a

a AND c

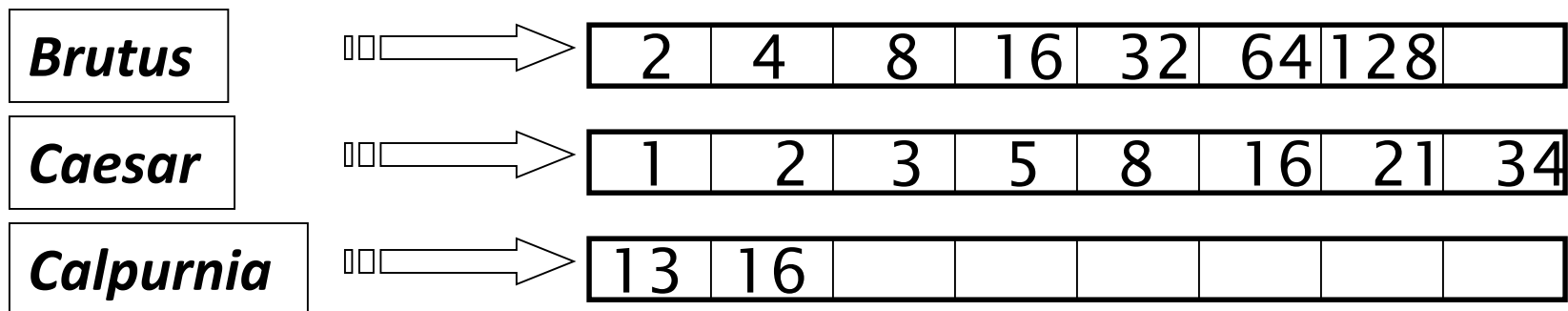
Βασική Ορολογία

- **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

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.

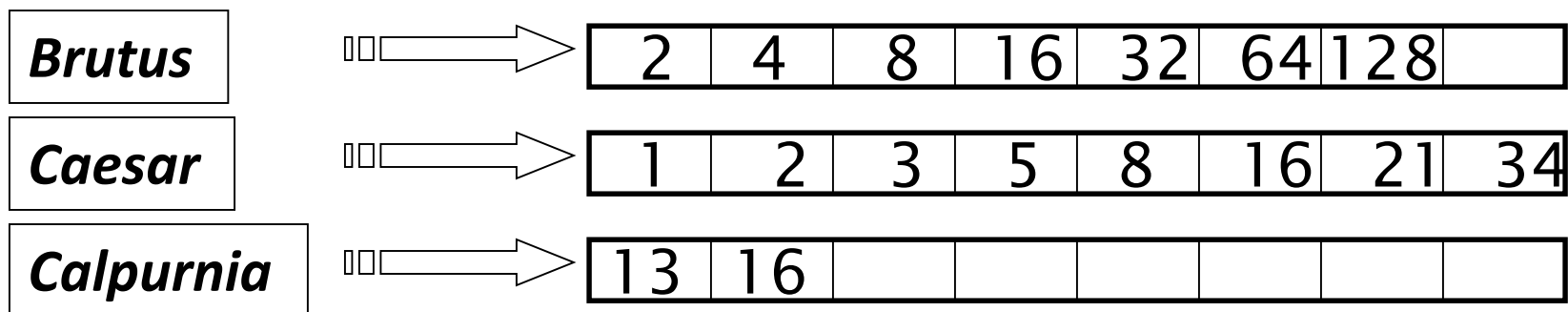


Query: Brutus AND Caesar AND Calpurnia

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

((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

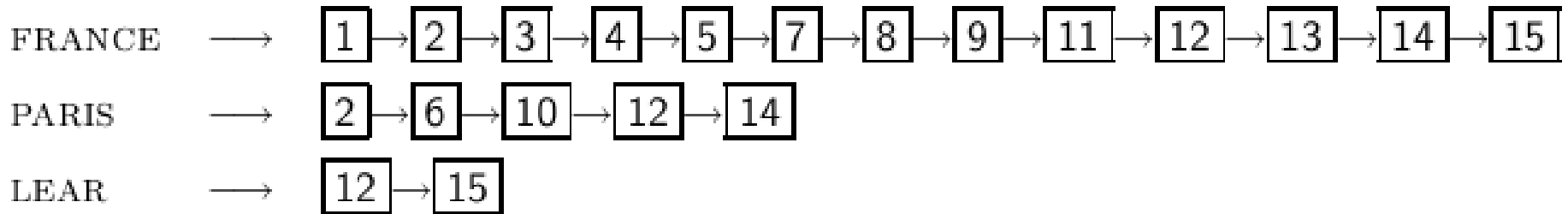
- χρησιμοποιώντας δυαδική αναζήτηση στη μεγάλη λίστα (λογαριθμικός χρόνος)
- αποθήκευση μεγάλης λίστας ως hashtable (σταθερά)

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: παράδειγμα



paris AND france AND lear

((paris AND NOT france) OR lear)

❖ 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 το 1975, 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!

Παράδειγμα: 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

Λεξιλόγιο Όρων και Λίστες Καταχωρήσεων

Τι άλλο θα δούμε σήμερα;

- Προ-επεξεργασία για τη δημιουργία του λεξιλογίου όρων

- Έγγραφα
- Tokenization – Μονάδες εγγράφου
- Ποιους όρους εισάγουμε στο ευρετήριο;

- Καταχωρήσεις

- Γρηγορότερη συγχώνευση: *Λίστες Παράβλεψης* (skip lists)
- *Λίστες καταχωρήσεων με πληροφορίες θέσεων* (Positional postings) και ερωτήματα φράσεων (phrase queries)

Ακολουθία εγγράφων

Έγγραφο 1 (d1) : Το Παν. Ιωαννίνων ιδρύθηκε το 1970.

Έγγραφο 2 (d2) : Τα Ιωάννινα είναι η μεγαλύτερη πόλη της Ηπείρου.

Έγγραφο 3 (d3) : Η πτυχιακή εξεταστική στο Τμήμα Μηχ. Η/Υ και Πληροφορικής θ' αρχίσει την 1^η Φεβρουαρίου.

Έγγραφο 4 (d4) : Οι μαθητές των Ιωαννίνων άριστευσαν στις εξετάσεις για την εισαγωγή στα Πανεπιστήμια.

Έγγραφο 5 (d5): Το 2017 ιδρύθηκε Πολυτεχνική Σχολή στο ΠΙ.



Τους όρους που θα εισάγουμε στο ευρετήριο

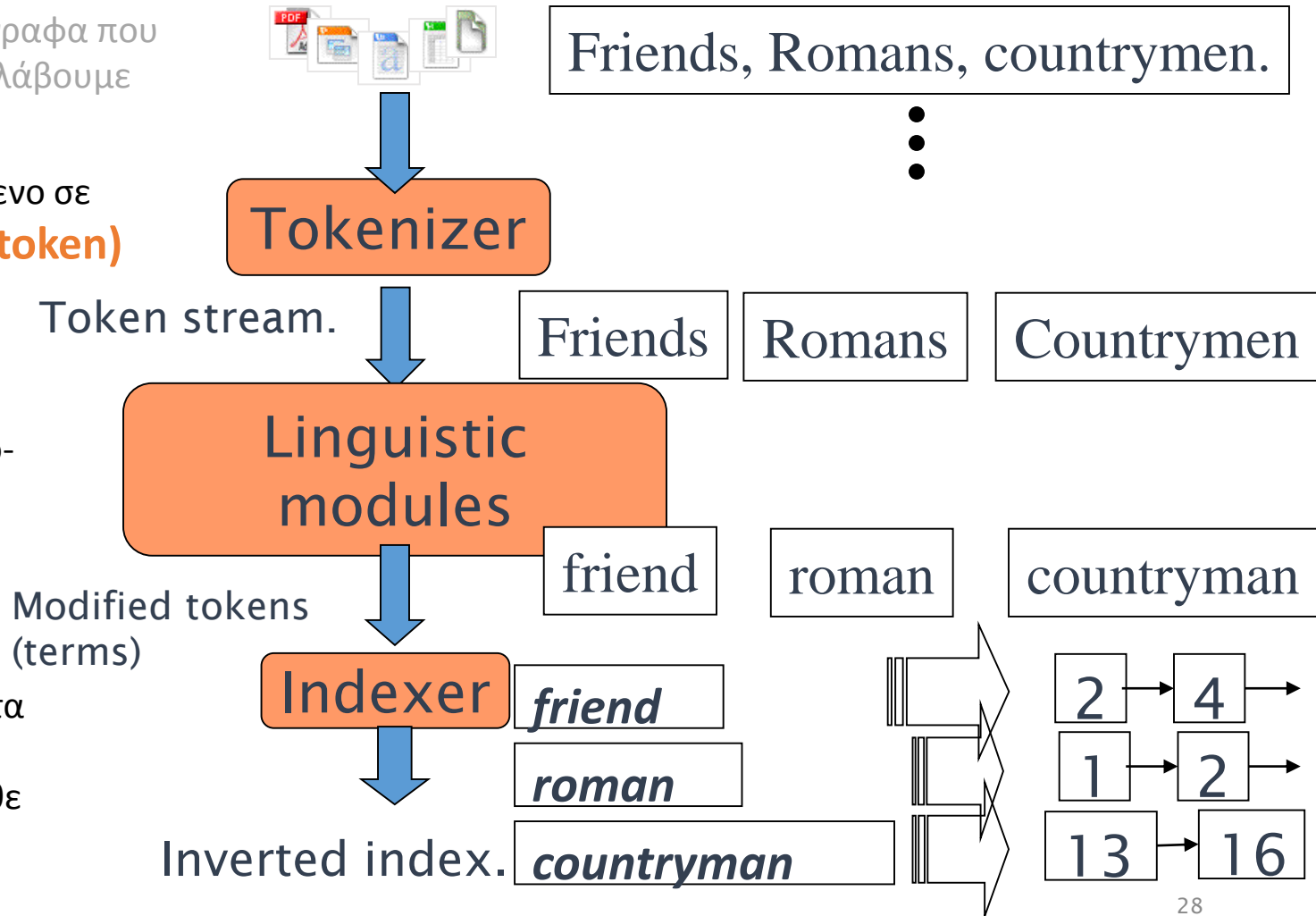
Τα βασικά βήματα για την κατασκευή του ευρετηρίου

1. Συλλέγουμε τα έγγραφα που θέλουμε να συμπεριλάβουμε στο ευρετήριο

2. Διαιρούμε το κείμενο σε γλωσσικά σύμβολα (**token**)

3. Γλωσσολογική προεπεξεργασία των συμβόλων

4. Ευρετηριάζουμε τα έγγραφα στα οποία περιλαμβάνεται κάθε όρος



Parsing

Λήψη της ακολουθίας χαρακτήρων ενός εγγράφου

Ποια είναι τα θέματα;

- Σε τι format?
 - pdf/word/excel/html ή και zip
 - Αν σε δυαδική μορφή - χρήση αποκωδικοποιητή (decoder) ώστε ακολουθία χαρακτήρων
- Σε ποια φυσική γλώσσα?
- Σε διαφορετικές κωδικοποιήσεις (σύνολο χαρακτήρων/character set)
 - Π.χ., UTF-8

Parsing

- Να αγνοήσουμε τα ειδικά σύμβολα (mark up)
 - JSON, XML
 - `&` -> `&` (XML)

Complications: Format/language

- Τα έγγραφα για τα οποία κατασκευάζουμε το ευρετήριο μπορεί να είναι γραμμένα σε διαφορετικές γλώσσες το καθένα
 - Στο ίδιο ευρετήριο μπορεί να υπάρχουν όροι από πολλές γλώσσες
- Πολλαπλές γλώσσες/format μπορεί να εμφανίζονται και σε ένα έγγραφο ή στα τμήματά του
 - *French email στα Γαλλικά με pdf attachment στα Γερμανικά.*

❖ Πως θα το καταλάβουμε;

Πρόβλημα ταξινόμησης (classification) αλλά στην πράξη συνήθως επιλογή από το χρήστη, χρήση μετα-δεδομένων αρχείου κλπ

Όχι πάντα γραμμική ακολουθία χαρακτήρων

ك ت ا ب ← كِتَابٌ
 un b ā t i k
 /kitābun/ 'a book'

Αραβικά: δισδιάστατη ακολουθία χαρακτήρων και χαρακτήρες σε μεικτή σειρά (φωνήεντα διακριτά σημεία πάνω και κάτω από τα γράμματα)

Από δεξιά στα αριστερά

Πιθανή απουσία φωνηέντων

Η αντίστοιχη **ακουστική** γραμμική ακολουθία

Μονάδα εγγράφου

Ποια θεωρείται η *μονάδα εγγράφου* που βάζουμε στο ευρετήριο;

- Ένα αρχείο;
- Ένα email; (από τα πολλά στο ένα αρχείο του mbox)
- Ένα email με 5 συνημμένα έγγραφα (attachments); Αν το 1 συνημμένο σε μορφή zip;
- Ανάποδα: εργαλεία χωρίζουν ένα αρχείο σε πολλά, (PPT ή LaTeX σε πολλαπλές HTML σελίδες) ίσως ένωση τους

Αναλυτικότητα ευρετηρίασης (indexing granularity)

Π.χ., ποια πληροφορία για ένα βιβλίο έχουμε στο ευρετήριο (σε επίπεδο κεφαλαίου, παραγράφου, πρότασης;)

Ακρίβεια/ανάκληση

Προβλήματα με μεγάλα έγγραφα -> θα δούμε πληροφορία εγγύτητας

ΣΥΜΒΟΛΑ (Tokens) και ΟΡΟΙ (TERMS)

Tokenization – Διαίρεση σε Σύμβολα

- Είσοδος: “*Friends, Romans, Countrymen*”
- Έξοδος: Tokens
 - *Friends*
 - *Romans*
 - *Countrymen*
- Ένα σύμβολο (**token**) είναι μια ακολουθία από χαρακτήρες σε ένα κείμενο (που είναι ομαδοποιημένοι ως μια χρήσιμη σημασιολογικά μονάδα)
- Κάθε τέτοιο token είναι υποψήφιο για να εισαχθεί στο ευρετήριο μετά από περαιτέρω επεξεργασία

Tokenization – Διαίρεση σε Σύμβολα

- **Token** (λεκτική μονάδα)
- **Type** (τύπος) μία ομάδα (κλάση) από tokens που αποτελείται από την ίδια ακολουθία χαρακτήρων
- **Term** (όρος) συχνά κανονικοποιημένος τύπος που εισάγεται στο ευρετήριο του συστήματος

Παράδειγμα: to sleep perchance to dream

Tokenization – Διαίρεση σε Σύμβολα

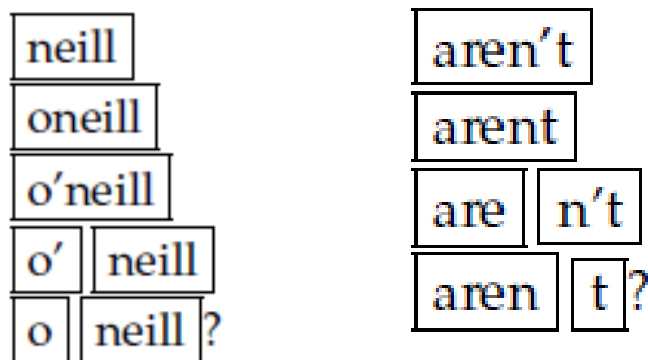
Αλλά ποια είναι τα κατάλληλα tokens;

Αρκεί να χωρίσουμε το κείμενο στα κενά και στα σημεία στίξης;

Εξαρτάται από τη γλώσσα

Tokenization: Θέματα

- Αγγλικά: απόστροφος (σύντμηση και γενική κτητική)
 - *Finland's capital* → *Finland? Finlands? Finland's?*
 - *Mr. O' Neill thinks that the boys' stories about Chile's capital aren't amusing*



- καθορίζουν ποιες Boolean ερωτήσεις θα απαντούν
πχ *neill AND capital*, *o'neill AND capital*

Την ίδια πολιτική και στην ερώτηση και στο κείμενο

Tokenization: Θέματα

- **Ενωτικό (hyphen):**

- (συνένωση λέξεων ως επωνυμίες) **Hewlett-Packard** → **Hewlett** και **Packard** ως δύο tokens ?
- (ομαδοποίηση λέξεων) **state-of-the-art** ή **the-hold-him-back-and-drag-him-away maneuver** (να διασπάσουμε την ακολουθία;)
- (χωρισμός φωνηέντων) **co-education** (ως μια λέξη;)
- **lowercase, lower-case, lower case ?**

(ως πρόβλημα ταξινόμησης, απλοί ευριστικοί κανόνες, κλπ)

- **Διάσπαση στο κενό σύμβολο**

- **San Francisco, Los Angeles York University vs New York University** (διάσπαση ονομάτων) αλλά πως μπορούμε να το καταλάβουμε;
- Συχνά και τα δύο **white space, white-space** και **whitespace**
- Επίσης, ημερομηνίες και αριθμοί τηλεφώνου

- Έη και συνδυασμός

- **San Francisco-Los Angeles**

- ✓ Την ίδια πολιτική και στην ερώτηση και στο κείμενο
- ✓ Μερικά συστήματα ζητούν οι χρήστες πάντα το ‘-’ όταν θέλουν να εξεταστούν όλες οι περιπτώσεις
- ✓ Αντιμετώπιση ως φράσεις (πχ lower, case, lowercase)

Tokenization: Αριθμοί

- **3/12/91** **Mar. 12, 1991** **12/3/91**
- **55 B.C.**
- **B-52**
- **My PGP key is 324a3df234cb23e**
- **(800) 234-2333**
 - Συχνά περιέχουν ενδιάμεσα κενά
 - Τα παλιότερα συστήματα μπορεί να μη έβαζαν στο ευρετήριο τους αριθμούς
 - Συχνά όμως είναι χρήσιμοι, πχ αναζήτηση για κώδικες λάθους error codes/stacktraces στο web, IP διευθύνσεις, package tracking numbers
 - (Χρήση n-grams)
 - Ευρετηριοποίηση των μεταδεδομένων ξεχωριστά
 - Ημερομηνία δημιουργίας, format, κλπ

Tokenization

- Επίσης ειδικές λέξεις
 - M*A*S*H
 - C++
 - C#
- Αλλά και email και web, IP διευθύνσεις, κλπ δε θέλουμε να τις «σπάσουμε»

Tokenization: άλλες γλώσσες

- Γαλλικά
 - *L'ensemble* → (σύντμηση άρθρου)
 - *L ? L' ? Le ?*
 - Θα θέλαμε τα *l'ensemble* να ταιριάζει με το *un ensemble*
 - Έως το 2003, δεν το υποστήριζε το Google
 - *Internationalization!*
- Γερμανικά (οι σύνθετες λέξεις δεν διαχωρίζονται)
 - *Lebensversicherungsgesellschaftsangestellter* (life insurance company employee)
 - Τα Γερμανικά συστήματα ανάκτησης πληροφορίας χρησιμοποιούν μια μονάδα **compound splitter**
 - Βελτίωση της απόδοσης κατά 15%

Tokenization: άλλες γλώσσες

莎拉波娃现在居住在美国东南部的佛罗里达。今年4月9日，莎拉波娃在美国第一大城市纽约度过了18岁生日。生日派对上，莎拉波娃露出了甜美的微笑。

Κινέζικα: δεν υπάρχουν κενά

Tokenization: άλλες γλώσσες

- Τα Κινέζικα και τα Ιαπωνικά δεν έχουν κενούς χαρακτήρες ανάμεσα στις λέξεις:
 - 莎拉波娃现在居住在美国东南部的佛罗里达。

Χωρισμός σε λέξεις (word segmentation)

- Διάφορες τεχνικές: χρήση λεξικού και ταίριασμα της μεγαλύτερης ακολουθίας, μηχανική μάθηση

Αλλά δεν υπάρχει πάντα μια μοναδική tokenization

Tokenization: άλλες γλώσσες

和尚

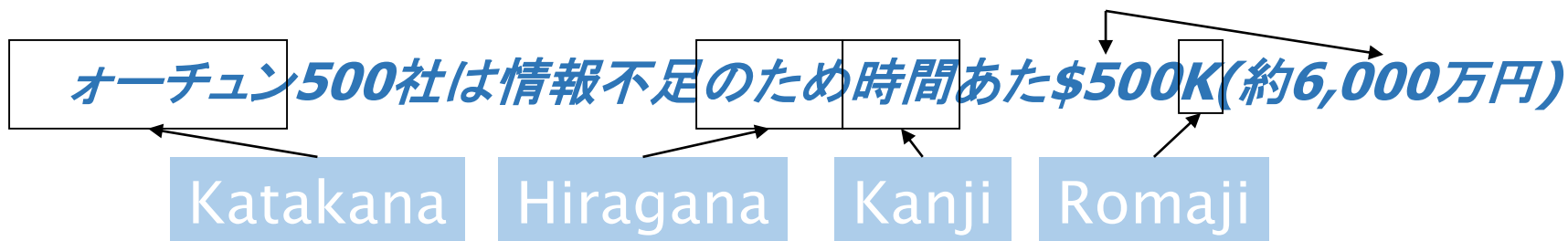
Κινέζικα: είτε ως ακολουθία δύο λέξεων “and” και “still” ή ως μια λέξη “monk”

Tokenization: άλλες γλώσσες

Αντί για ευρετηριοποίηση σε επίπεδο λέξεων
ευρετηριοποίηση όλων των ακολουθιών k -χαρακτήρων
(k -grams)

Tokenization: άλλες γλώσσες

- Ακόμα πιο δύσκολο στα Ιαπωνικά, ανάμιξη πολλαπλών αλφάβητων
 - Ημερομηνίες/ποσά σε πολλά formats



Ο χρήστης μπορεί να διατυπώσει την ερώτηση μόνο σε hiragana!

Tokenization: άλλες γλώσσες

- Τα Αραβικά και στα Εβραϊκά γράφονται από τα δεξιά προς τα αριστερά, αλλά με συγκεκριμένα τμήματα (πχ αριθμοί) να γράφονται από τα αριστερά στα δεξιά
- Οι λέξεις διαχωρίζονται αλλά τα γράμματα μέσα στις λέξεις περίπλοκοι χαρακτήρες

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.
 ← → ← → ← start

- ‘Algeria achieved its independence in 1962 after 132 years of French occupation.’
- Με χρήση Unicode, η αποθηκευμένη μορφή είναι απλοποιημένη

Stop words (Διακόπτουσες λέξεις)

- Χρήση **stop list**, αποκλείουμε από το λεξικό τις πιο κοινές λέξεις (με βάση τη συχνότητα συλλογής (collection frequency)). Γιατί;
 - Έχουν μικρό σημασιολογικό περιεχόμενο: *a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with*
 - Είναι πάρα πολλές: ~30% των καταχωρήσεων αφορούν τις πιο συχνές 30 λέξεις

Stop words (Διακόπτουσες λέξεις)

- Ωστόσο η τάση είναι *να μη χρησιμοποιούνται* λίστες:
 - Καλές τεχνικές *συμπίεσης* οδηγούν στο να ελαχιστοποιούν το χώρο που χρειάζεται για την αποθήκευση τους
 - Καλές τεχνικές για την *επεξεργασία ερωτημάτων* (βάρη όρων και διάταξη όρων στα ευρετήρια βάση σημαντικότητας) μειώνουν το κόστος στην εκτέλεση μιας ερώτησης εξαιτίας των stop words.

Είναι χρήσιμα για:

- Φράσεις: “King **of** Denmark”
- Τίτλους τραγουδιών, κλπ.: “Let it be”, “To be or not to be”
- “Σχεσιακά” ερωτήματα: “flights to London”

Κανονικοποίηση (Token normalization)

- Token normalization: κανονικοποίηση των token ώστε να εντοπίζονται αντιστοιχίες παρά κάποιες μικρές διαφορές
- Συχνά ορίζουμε έμμεσα κλάσεις ισοδυναμίας (equivalence classes) για τους όρους, δηλαδή, τις απεικονίζουμε στον ίδιο όρο π.χ.,
 - Σβήνουμε τις τελείες από έναν όρο
 - ***U.S.A., USA*** (***USA***)
 - Σβήνουμε τα ενωτικά από έναν όρο ***anti-discriminatory, antidiscriminatory***

Απλοί κανόνες αντιστοίχισης (mapping rules)

- Δε χρειάζεται πλήρης προσδιορισμός
- Μερικές φορές δεν είναι εύκολο να εντοπιστεί πότε χρειάζεται προσθήκη χαρακτήρων

Κανονικοποίηση

Μια εναλλακτική προσέγγιση στις κλάσεις ισοδυναμίας είναι να *κρατάμε όλα* τα μη κανονικοποιημένα *token* (να ορίσουμε αντιστοιχία μεταξύ τους)

- (μπορεί να χρησιμοποιηθεί και για συνώνυμα, «χειροποίητες» λίστες συνωνύμων)

1. (α) Ευρετηριοποίηση του μη κανονικοποιημένου όρου και (β) διεύρυνση κατά την ερώτηση (διάζευξη)

Enter: **windows** Search: **Windows, windows, window**

(συνώνυμα) Enter: **car** Search: **car automobile**

2. Εναλλακτικά, καταχωρούμε το έγγραφο στις λίστες καταχώρησης κάθε συνώνυμου (πχ έγγραφο που περιέχει το car καταχωρείται και στο automobile)

Το 1 ή το 2 είναι καλύτερο;

Κανονικοποίηση σε όρους

Μη συμμετρική διεύρυνση

- Ένα παράδειγμα όπου αυτό μπορεί να φανεί χρήσιμο
 - Enter: **window** Search: **window, windows**
 - Enter: **windows** Search: **Windows, windows, window**
 - Enter: **Windows** Search: **Windows**
- Θεωρητικά πιο ισχυρό από τις λίστες αλλά λιγότερο αποδοτικό
- Είναι η κανονικοποίηση πάντα καλή, U.S.A, C.A.T?

Κανονικοποίηση: άλλες γλώσσες, τόνοι, διακριτικά

60% ιστοσελίδων στα Αγγλικά (2007) – 1/3 των χρηστών του διαδικτύου - 10% του παγκόσμιου πληθυσμού μιλούν Αγγλικά

- Accents: π.χ., Γαλλικά *résumé* vs. *resume*.
- Umlauts: π.χ., Γερμανικά: *Tuebingen* vs. *Tübingen*
 - Πρέπει να είναι ισοδύναμα
- Πιο βασικό κριτήριο:
 - Πως προτιμούν οι χρήστες να γράφουν αυτές τις λέξεις στα ερωτήματά τους
- Ακόμα και σε γλώσσες που έχουν accents, οι χρήστες δεν τα πληκτρολογούν *(σκεφτείτε τους τόνους στα Ελληνικά)*
 - Οπότε συχνά είναι καλύτερο να κανονικοποιούμε ή να αφαιρούμε το accent από ένα όρο
 - *Tuebingen, Tübingen, Tubingen \ Tubingen*

Κανονικοποίηση: άλλες γλώσσες

- Κανονικοποίηση σε περιπτώσεις όπως οι ημερομηνίες
 - *7月30日 vs. 7/30*
 - *Japanese use of kana vs. Chinese characters*
- Tokenization και οι κανονικοποίηση μπορεί να εξαρτάται από τη γλώσσα
όποτε μαζί με αναγνώριση γλώσσας
- Βασικό: Πρέπει το κείμενο που θα ευρετηριοποιηθεί και οι
όροι στο ερώτημα να κανονικοποιούνται με τον ίδιο τρόπο

Morgen will ich in MIT ...

Is this
German "mit

Μετατροπή σε κεφαλαία/μικρά

- Μετατροπή όλων των γραμμάτων σε μικρά (case folding)
 - εξαίρεση: κεφαλαία στη μέση της πρότασης; (truefolding)
 - e.g., **General Motors**
 - **Fed** vs. **fed**
 - **Bush** vs. **bush**
 - Πρακτικά μετατροπή όλων σε μικρά, αφού συχνά οι χρήστες χρησιμοποιούν μικρά ανεξάρτητα της «σωστής» χρήσης των κεφαλαίων
 - Παράδειγμα από τη Google:
 - Δοκιμάστε την ερώτηση C.A.T.
 - #1 αποτέλεσμα για "cat"



Θησαυροί (Thesauri) και soundex

- Πως χειριζόμαστε τα συνώνυμα και τα ομώνυμα;
 - Π.χ., κατασκευάζοντας **λίστες ισοδυναμίας** με το χέρι
 - *car = automobile* *color = colour*
 - Μπορούμε να το ξαναγράψουμε (rewrite) για να δημιουργήσουμε κλάσεις ισοδυναμίας όρων
 - Καταχωρούμε το έγγραφο στις λίστες καταχώρησης κάθε συνώνυμου (πχ έγγραφο που περιέχει το *car* καταχωρείται και στο *automobile* και το ανάποδο)
 - Ή να διευρύνουμε το ερώτημα
 - Όταν το ερώτημα περιέχει *automobile*, ψάξε και για το *car*
- Τι γίνεται με τα ορθογραφικά λάθη (spelling mistakes)?
 - Μια προσέγγιση είναι το soundex, που σχηματίζει κλάσεις ισοδυναμίας από λέξεις βασιζόμενες σε ακουστικούς ευριστικούς κανόνες (phonetic heuristics)

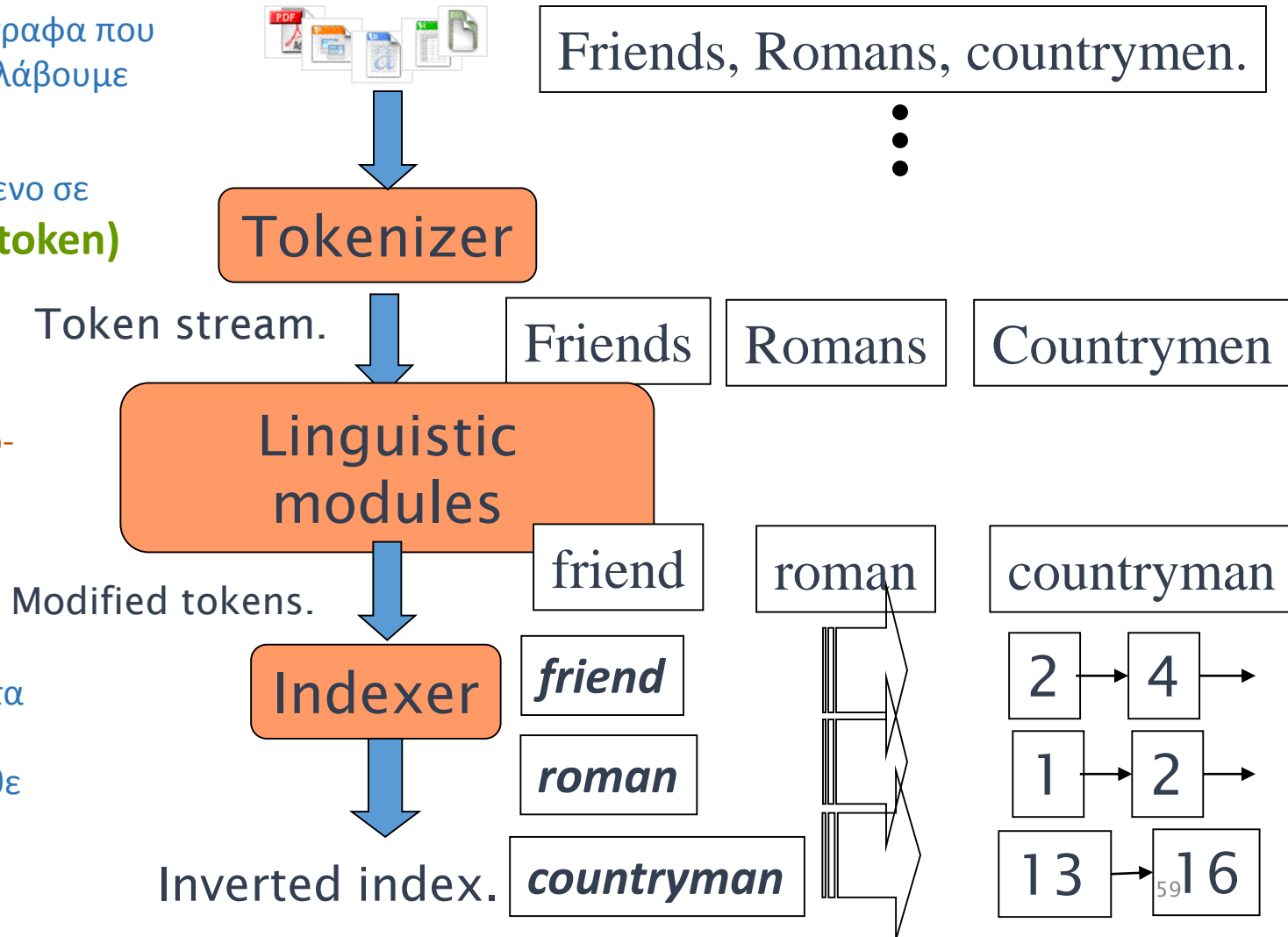
Τα βασικά βήματα για την κατασκευή του ευρετηρίου

1. Συλλέγουμε τα έγγραφα που θέλουμε να συμπεριλάβουμε στο ευρετήριο

2. Διαιρούμε το κείμενο σε γλωσσικά σύμβολα (**token**)

3. Γλωσσολογική προεπεξεργασία των συμβόλων

4. Ευρετηριάζουμε τα έγγραφα στα οποία περιλαμβάνεται κάθε όρος



Λημματοποίηση και Stemming

Δύο διαφορετικές προσεγγίσεις: λημματοποίηση και stemming

Πριν την εισαγωγή στο ευρετήριο

- Αναγωγή των όρων *στις ρίζες του* (λημματοποίηση)
- *Περικοπή κλιτικών καταλήξεων* και αναγωγή παράγωγων μορφών μιας λέξης σε κοινή βασική μορφή (stemming)

Λημματοποίηση (Lemmatization)

- Π.χ.,
 - *am, are, is* → *be*
 - *car, cars, car's, cars'* → *car*
- *the boy's cars are different colors* → *the boy car be different color*
- Η **λημματοποίηση** προϋποθέτει «ορθή» αναγωγή που χρησιμοποιεί λεξιλόγιο και μορφολογική ανάλυση των λέξεων και επιστρέφει τη βασική μορφή της λέξης, το λήμμα
- POS (part of speech)

Lemmatizer

Stemming (Περιστολή)

- “Stemming” υπονοεί ωμό κόψιμο των καταλήξεων
 - εξαρτάται από τη γλώσσα
 - π.χ., ***automate(s), automatic, automation*** όλα ανάγονται στο ***automat***.

for example compressed and compression are both accepted as equivalent to compress.



for exampl compress and compress ar both accept as equival to compress

Ο αλγόριθμος του Porter

- Ο πιο διαδεδομένος αλγόριθμος stemming για τα Αγγλικά
 - Τα αποτελέσματα δείχνουν ότι είναι τουλάχιστον τόσο καλός όσο οι άλλες επιλογές
- Συμβάσεις + 5 φάσεις περικοπών
 - Οι φάσεις εφαρμόζονται διαδοχικά
 - Κάθε φάση αποτελείται από ένα σύνολο κανόνων
 - Παράδειγμα σύμβασης: Επιλογή εκείνου του κανόνα από κάθε ομάδα που μπορεί να εφαρμοστεί στο μεγαλύτερο επίθεμα.

www.tartarus.org/~martin/PorterStemmer

Χαρακτηριστικοί κανόνες του Porter

Ομάδα κανόνων της πρώτης φάσης:

- *sses* → *ss*
- *ies* → *i*
- *ss* → *ss*
- *s* →

Παράδειγμα

caresses → *caress*

ponies → *poni*

caress → *caress*

cats → *cat*

Άλλοι κανόνες

- *ational* → *ate*
- *tional* → *tion*
- Οι κανόνες χρησιμοποιούν ένα είδους μέτρου (*measure*) που ελέγχει το πλήθος των συλλαβών
- $(m > 1)$ *EMENT* →
 - *replacement* → *replac*
 - *cement* → *cement*

Άλλοι stemmers

- Υπάρχουν και άλλοι π.χ., Lovins stemmer
 - <http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm>
 - Ένα πέρασμα, αφαίρεση της μεγαλύτερης κατάληξης (περίπου 250 κανόνες)
- Πλήρη μορφολογική ανάλυση – περιορισμένα οφέλη
- Βοηθά το stemming και οι άλλοι κανονικοποιητές;
 - English: ανάμικτα αποτελέσματα. Βοηθά την ανάκληση αλλά βλάπτει την ακρίβεια
 - operative (dentistry) ⇒ oper
 - operational (research) ⇒ oper
 - operating (systems) ⇒ oper
- Οπωσδήποτε χρήσιμο για Ισπανικά, Γερμανικά, Φινλανδικά
 - 30% βελτίωση για τα Φινλανδικά

Άλλοι stemmers: σύγκριση

- Sample text:* Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Porter stemmer:* such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Lovins stemmer:* such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Paice stemmer:* such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Εξάρτηση από τη γλώσσα

- Πολλά από τα παραπάνω περιλαμβάνουν μετασχηματισμούς που
 - Εξαρτώνται από τη γλώσσα και
 - Συχνά από την εφαρμογή
- Με τη μορφή “plug-in” πριν τη διαδικασία δεικτοδότησης
- Ελεύθερου λογισμικού και εμπορικά

Ακολουθία εγγράφων

Περίληψη

Έγγραφο 1 (d1) : Το Παν. Ιωαννίνων ιδρύθηκε το 1970.
Έγγραφο 2 (d2) : Τα Ιωάννινα είναι η μεγαλύτερη πόλη της Ηπείρου.
Έγγραφο 3 (d3) : Η πτυχιακή εξεταστική στο Τμήμα Μηχ. Η/Υ και Πληροφορικής θ' αρχίσει την 1^η Φεβρουαρίου.
Έγγραφο 4 (d4) : Οι μαθητές των Ιωαννίνων αρίστευσαν στις εξετάσεις για την εισαγωγή στα Πανεπιστήμια.
Έγγραφο 5 (d5): Το 2017 ιδρύθηκε Πολυτεχνική Σχολή στο ΠΙ.

Granularity: Μονάδα εγγράφου



Token (λεκτικές μονάδες)

Θέματα

- Που σταματάμε: κενό/σημείο στίξης αλλά και απόστροφοι/όχι κενό/παύλα, κλπ
- Stop words (το, και?)
- Κανονικοποίηση
 - Κεφαλαία/μικρά
 - Τόνοι
 - Κανόνες vs Λίστες ισοδυναμίας



Όροι (terms) που θα εισαχθούν στο ευρετήριο

- **Περιστολή (stemming)** περικοπή καταλήξεων
- **Λημματοποίηση (lemmatization)** γλωσσική/μορφολογική επεξεργασία και αναγωγή της λέξης στη ρίζα της

✓ Ίδια πολιτική και στο κείμενο και στην ερώτηση

Επανάληψη (ερωτήσεις)

Άσκηση 2.1

Are the following statements true or false?

- a. In a Boolean retrieval system, stemming never lowers precision.
- b. In a Boolean retrieval system, stemming never lowers recall.
- c. Stemming increases the size of the vocabulary.
- d. Stemming should be invoked at indexing time but not while processing a query.

Boolean Μοντέλο

- Οι χρήστες διατυπώνουν ερωτήματα με τη μορφή *Boolean* εκφράσεων, δηλαδή όρων συνδυασμένων με *AND*, *OR* και *NOT*
- Κείμενο ως σύνολο όρων
- Επιστρέφονται ως απάντηση *όλα* τα κείμενα που ικανοποιούν το ερώτημα *χωρίς διάταξη*

Άσκηση

Έγγραφο 1 (d1) : Summer holidays in Crete.
Έγγραφο 2 (d2) : Crete is an island in Greece.
Έγγραφο 3 (d3) : Summer is the best season of the year.
Έγγραφο 4 (d4) : Holidays in the island of Crete.
Έγγραφο 5 (d5): Many islands in Greece.

Ποιοι είναι οι όροι (terms)

Κατασκευή

(α) Πίνακας σύμπτωσης (incident matrix)

(β) Ανεστραμμένο ευρετήριο (inverted index) -Λεξικό (λεξιλόγιο) – λίστα καταχωρήσεων

(γ) Απάντηση στις ερωτήσεις “island”, “island AND Crete” με πίνακα και λίστες

Λίστες Καταχωρήσεων

Αντεστραμμένο ευρετήριο

Όροι (term) -> Λεξιλόγιο (Vocabulary)

Καταχώρηση
(Posting)

Brutus

1 | 2 | 4 | 11 | 31 | 45 | 173 | 174

Caesar

1 | 2 | 4 | 5 | 6 | 16 | 57 | 132

Calpurnia

2 | 31 | 54 | 101

Λεξικό (Dictionary)

Λίστα Καταχωρήσεων (Postings list)

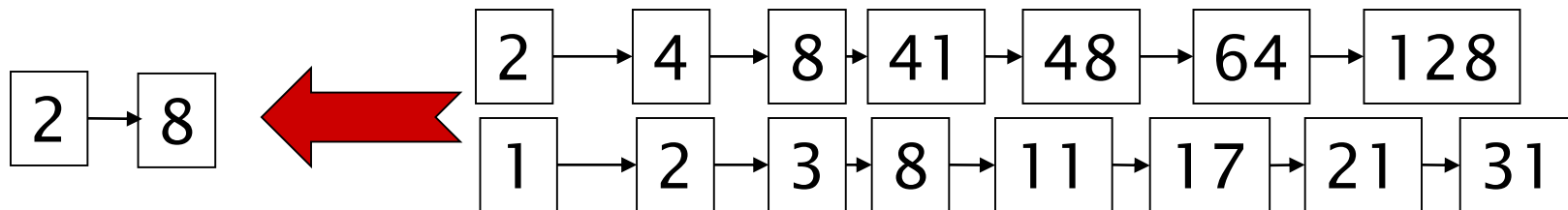
Σε διάταξη με βάση το docID

Τι θα δούμε σήμερα;

Καταχωρήσεις

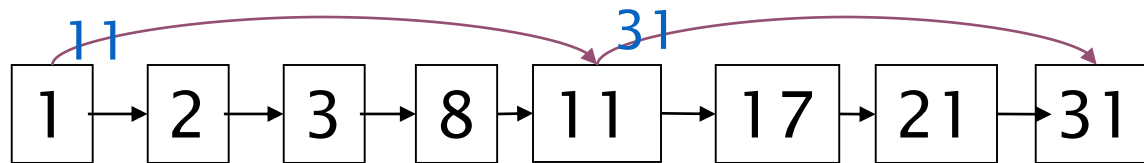
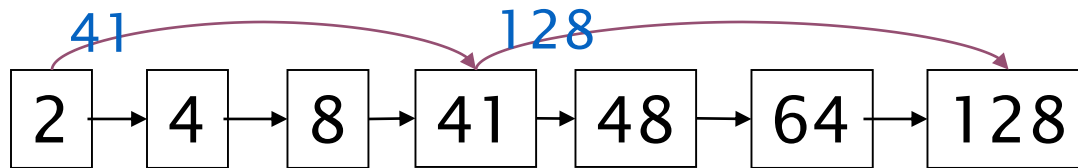
- Ασκήσεις για συγχώνευση
- Γρηγορότερη συγχώνευση: *Λίστες Παράβλεψης (skip lists)*
- *Λίστες καταχωρήσεων με πληροφορίες θέσεων (positional postings)* και ερωτήματα φράσεων (*phrase queries*)

Βασική συγχώνευση



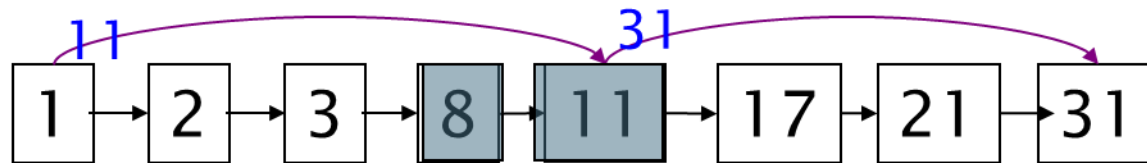
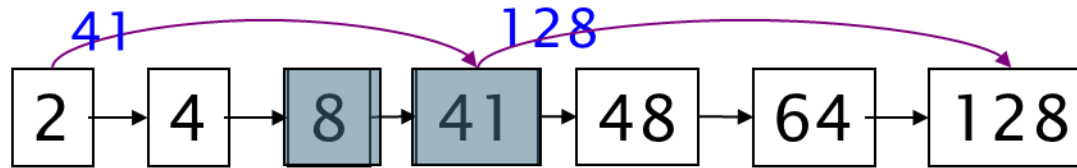
Αν τα μήκη των λιστών είναι m και n , $O(m+n)$

Επέκταση των λιστών με δείκτες παράβλεψης skip pointers (κατά την κατασκευή του ευρετηρίου)



- Γιατί?
 - Για να αποφύγουμε (skip) καταχωρήσεις που δεν θα εμφανιστούν στο αποτέλεσμα της αναζήτησης.
- Πως?
- Που να τοποθετήσουμε αυτούς τους δείκτες?

Επεξεργασία ερωτήματος με skip pointers

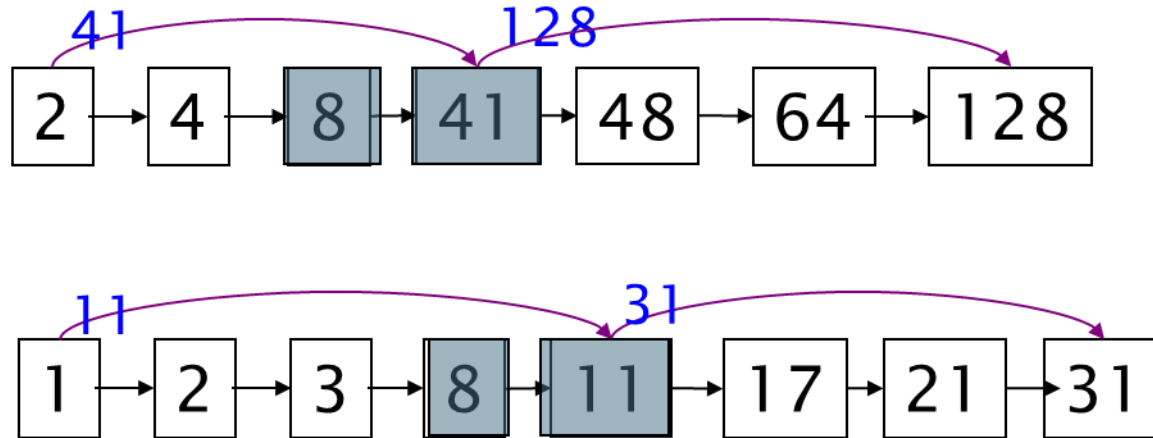


Υποθέστε ότι έχουμε διατρέξει τις λίστες και έχουμε βρει το κοινό στοιχείο **8** σε κάθε λίστα, το ταιριάζουμε και προχωράμε

Έχουμε **41** και **11**. Το **11** είναι το μικρότερο.

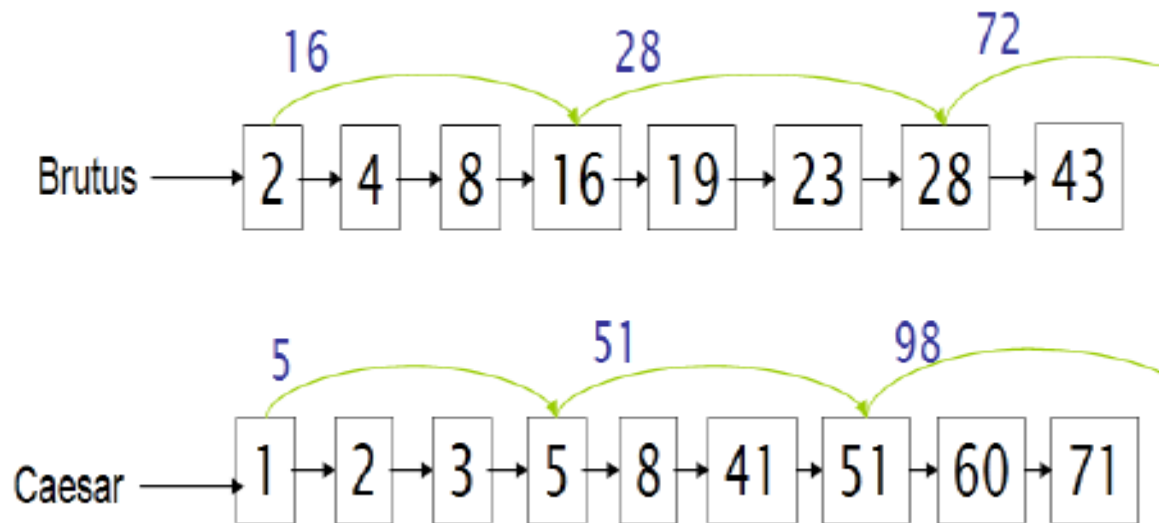
Ο δείκτης παράλειψης του **11** είναι το **31**, οπότε μπορούμε να παραβλέψουμε τις ενδιάμεσες καταχωρήσεις

Επεξεργασία ερωτήματος με skip pointers



Αριθμός συγκρίσεων χωρίς και με χρήση δεικτών παράβλεψης

Επεξεργασία ερωτήματος με skip pointers

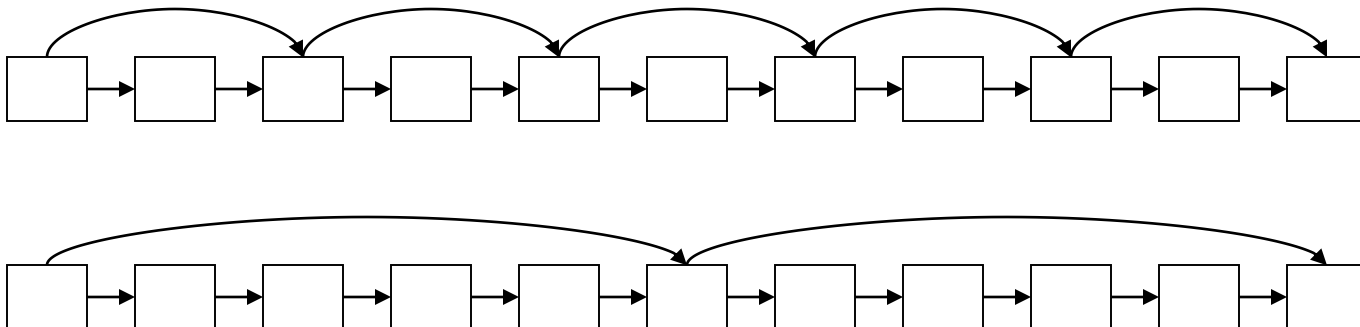


Αριθμός συγκρίσεων χωρίς και με χρήση δεικτών παράβλεψης

Που να τοποθετήσουμε τους δείκτες?

- Tradeoff:

- Πολλοί δείκτες παράβλεψης → μικρότερα διαστήματα παράβλεψης ⇒ μεγαλύτερη πιθανότητα παράβλεψης. Πολλές συγκρίσεις για να παραλείψουμε δείκτες.
- Λιγότεροι δείκτες παράβλεψης → λιγότερες συγκρίσεις δεικτών αλλά μεγαλύτερα διαστήματα ⇒ λίγες επιτυχημένες παραβλέψεις.



Τοποθέτηση των δεικτών

Απλώς ευριστικός: για καταχωρήσεις μήκους L , χρησιμοποίησε \sqrt{L} δείκτες παράβλεψης σε ίδια απόσταση μεταξύ τους (evenly spaced), δηλαδή σε απόσταση \sqrt{L}

- Αγνοεί την κατανομή των όρων της ερώτησης.
- Εύκολο αν το ευρετήριο είναι σχετικά στατικό. Δύσκολο αν το L αλλάζει συνεχώς λόγω τροποποιήσεων.
- Βοηθούσε (λόγω πιο αργής CPU). Όχι τόσο με το νέο υλικό εκτός αν *memory-based*
 - Το I/O κόστος για να φορτωθεί μια μεγαλύτερη (λόγω skip pointers) λίστα καταχωρήσεων μπορεί να υπερβαίνει το κέρδος από τη γρηγορότερη συγχώνευση

Ευρετήρια φράσεων

Ερωτήματα Φράσεων (phrase queries)

- Θέλουμε να μπορούμε να απαντάμε σε ερωτήματα όπως “*stanford university*” – ως φράση
- Οπότε το έγγραφο “*I went to university at Stanford*” δεν αποτελεί ταίριασμα.
 - Η έννοια των ερωτημάτων φράσεων έχει αποδειχθεί **πολύ δημοφιλής** και εύκολα κατανοητή από τους χρήστες,
 - Από τις λίγες μορφές αναζήτησης πέρα της βασικής που υιοθετήθηκαν (ερωτήσεις με «» αποτελούν το **10%**)
 - Ακόμα περισσότερες είναι *έμμεσα* ερωτήματα φράσεων
- Για να τα υποστηρίξουμε, δεν αρκούν εγγραφές της μορφής `<term : docs>`

Μια πρώτη προσέγγιση: Ευρετήρια ζευγών λέξεων (Biword indexes)

- Εισήγαγε στο ευρετήριο *κάθε διαδοχικό ζεύγος όρων* στο κείμενο ως φράση
- Για παράδειγμα το κείμενο “Friends, Romans, Countrymen” παράγει τα biwords
 - *friends romans*
 - *romans countrymen*
- Κάθε τέτοιο biword είναι τώρα ένας όρος του ευρετηρίου
- Επιτρέπει την επεξεργασία ερωτημάτων φράσεων με δύο λέξεις.

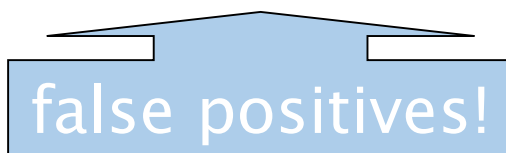
Μεγαλύτερες φράσεις

- Οι μεγαλύτερες φράσεις με κατάτμηση:

stanford university palo alto μπορεί να διασπαστεί ως ένα Boolean ερώτημα με biwords:

stanford university AND university palo AND palo alto

Χωρίς να εξετάσουμε τα έγγραφα, δεν μπορούμε να εξακριβώσουμε ότι τα έγγραφα που ικανοποιούν το παραπάνω ερώτημα περιέχουν τη φράση.



false positives!

Διευρυμένα biwords

- Επεξεργασία του κειμένου και εκτέλεση part-of-speech-tagging (POST).
- Ομαδοποιούμε τους όρους (έστω) σε ουσιαστικά- Nouns (N) και άρθρα/προθέσεις (X).
- **Διευρυμένο biword**: κάθε ακολουθία όρων της μορφής NX^*N
 - Κάθε τέτοιο διευρυμένο biword είναι τώρα ένας όρος του λεξικού

Διευρυμένα biwords

- Παράδειγμα: ***catcher in the rye***
N X X N
- Επεξεργασία ερωτήματος: χώρισε το σε N και X
 - Διαίρεσε την ερώτηση σε διευρυμένα biwords
 - Αναζήτησε στο ευρετήριο το: ***catcher rye***
- Παράδειγμα: ***cost overruns on a power plant***
 - “cost overruns” “overruns power” “power plant”

Θέματα

- False positives
- Περισσότερους από 2 όρους -> **Phrase index** (ευρετήριο φράσης)
- Δημιουργούνται πολύ μεγάλα λεξικά
 - Δεν είναι δυνατόν για μεγαλύτερες φράσεις από 2 λέξεις, μεγάλα ακόμα και για αυτές
- Τα ευρετήρια biword δεν είναι η συνήθης λύση (για όλα τα biwords) αλλά χρησιμοποιούνται ως μέρος πιο σύνθετων λύσεων

Λύση 2: Positional indexes (Ευρετήρια Θέσεων)

- Στις καταχωρήσεις, με κάθε όρο, αποθηκεύουμε και τη θέση (θέσεις) όπου εμφανίζονται τα tokens του:

<***term***, number of docs containing ***term***;

doc1: position1, position2 ... ;

doc2: position1, position2 ... ;

etc.>

Παράδειγμα

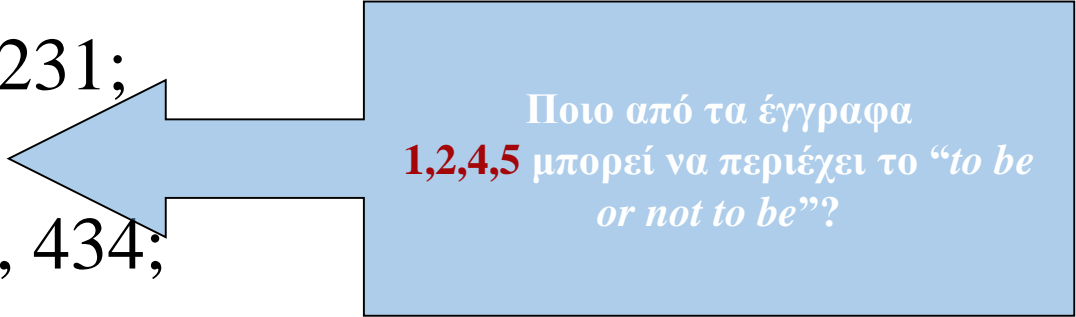
<*be*: 993427;

1: 7, 18, 33, 72, 86, 231;

2: 3, 149;

4: 17, 191, 291, 430, 434;

5: 363, 367, ...>



Ποιο από τα έγγραφα
1,2,4,5 μπορεί να περιέχει το “*to be
or not to be*”?

- Για ερωτήματα φράσεων, χρησιμοποιούμε έναν αλγόριθμο φράσεων αναδρομικά στο επίπεδο εγγράφου
- Αλλά τώρα δεν αρκεί η ισότητα των doc id

Επεξεργασία ερωτήματος φράσης

- Βρες τις εγγραφές του ευρετηρίου για τους όρους του ερωτήματος
- Συγχώνευσε τις doc:position λίστες για απαρίθμηση όλων των πιθανών θέσεων

Παράδειγμα ερωτήματος: “to₁ be₂ or₃ not₄ to₅ be₆”

TO, 993427:

1: <7, 18, 33, 72, 86, 231>;

2: <1, 17, 74, 222, 255>;

4: <8, 16, 190, 429, 433>;

5: <363, 367>;

7: <13, 23, 191>; ... >

BE, 178239:

1: <17, 25>;

4: <17, 191, 291, 430, 434>;

5: <14, 19, 101>; ... >

Ερωτήματα γειτονικότητας (Proximity queries)

- Η ίδια γενική μέθοδος για ερωτήματα γειτονικότητας (proximity searches)
- LIMIT! /3 STATUTE /3 FEDERAL /2 TORT
 - Πάλι, / k means “within k words of”.
- Μπορούμε να χρησιμοποιήσουμε ευρετήρια θέσεων αλλά όχι ευρετήρια biword.

Πολυπλοκότητα ερώτησης

- Αυξάνει την πολυπλοκότητα της ερώτησης από $O(T)$, T αριθμός εγγράφων σε $O(N)$, N αριθμός token.

Μέγεθος ευρετηρίου

- Μπορούμε να συμπίεσουμε τα position values/offsets
- Παρόλα αυτά, σημαντική αύξηση του χώρου αποθήκευση των λιστών καταχωρήσεων
- Αλλά χρησιμοποιείται ευρέως
- *Η σχετική θέση των όρων χρησιμοποιείται και εμμέσως για την κατάταξη των αποτελεσμάτων.*

Μέγεθος ευρετηρίου

- Χρειάζεται μια εγγραφή για κάθε εμφάνιση στο έγγραφο αντί για μια για κάθε έγγραφο
- Το μέγεθος του ευρετηρίου εξαρτάται από το μέσο μέγεθος του αρχείου
 - Μέσο μέγεθος web σελίδας <1000 όροι
 - SEC filings, books, even some epic poems ... πάνω από 100,000 όρους
- Έστω ένας όρος με συχνότητα 0.01% (1 σε 1000 όρους) σε ένα έγγραφο

Document size	Postings	Positional postings
1 000	1	1
1 00,000	1	100

Rules of thumb

- Ένα ευρετήριο θέσεων είναι 2–4 μεγαλύτερο από ένα απλό ευρετήριο
- Το μέγεθος του *συμπιεσμένου* ευρετηρίου είναι το 35–50% του όγκου του αρχικού κειμένου
- Αυτά αφορούν την Αγγλική (και παρόμοιες) γλώσσες

Συνδυαστικές μέθοδοι

- Αυτές οι δυο προσεγγίσεις μπορεί να συνδυαστούν
 - Για συγκεκριμένες φράσεις (“*Michael Jackson*”, “*Britney Spears*”) η συνεχής συγχώνευση καταχωρήσεων ευρετηρίου θέσεων δεν είναι αποδοτική
 - Ακόμα περισσότερο για φράσεις όπως “*The Who*”

Πότε biwords αντί για positional indexes?

- Αυτά που συναντώνται συχνά
- Τις ποιο «ακριβές»

Βασικός αλγόριθμος συγχώνευσης **AND**

INTERSECT(p_1, p_2)

```
1  answer  $\leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then  $\text{ADD}(\text{answer}, \text{docID}(p_1))$ 
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8          then  $p_1 \leftarrow \text{next}(p_1)$ 
9          else  $p_2 \leftarrow \text{next}(p_2)$ 
10 return answer
```

Βασικός αλγόριθμος συγχώνευσης

Exercise 1.11 [] (αλλαγή για AND NOT)**

How should the Boolean query x **AND NOT** y be handled?

Why is naive evaluation of this query normally very expensive?

Write out *a postings merge algorithm* that evaluates this query efficiently.

Exercise 1.10 [] (αλλαγή για OR)**

Write out a postings merge algorithm, for an x **OR** y query

Βασικός αλγόριθμος συγχώνευσης

Exercise 1.4 [★]

For the queries below, can we still run through the intersection in time $O(x + y)$, where x and y are the lengths of the postings lists for Brutus and Caesar? If not, what can we achieve?

- a. Brutus AND NOT Caesar
- b. Brutus OR NOT Caesar

ΤΕΛΟΣ 2^{ου} Κεφαλαίου

Ερωτήσεις?

Χρησιμοποιήθηκε κάποιο υλικό των:

✓ *Pandu Nayak and Prabhakar Raghavan, CS276:Information Retrieval and Web Search (Stanford)*