ΜΥΕ003: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Κατασκευή Λεξιλογίου Όρων.

Ακαδημαϊκό Έτος 2024-2025

Περιεχόμενα

- Σύντομη επανάληψη

- Προεπεξεργασία: Parsing
- Επεκτάσεις των βασικών δομών (λεξικό, ευρετήριο)

Τι είναι η Ανάκτηση Πληροφορίας (Information Retrieval);

Ανάγκη πληροφόρησης

Information need

Συλλογή Εγγράφων

Corpus, collection, repository

Ερώτημα

λέξεις κλειδιά (keywords)

ΣΑΠ

Σύστημα Ανάκτησης Πληροφορίας
Information Retrieval System

Απάντηση (συναφή έγγραφα σε διάταξη με βάση τη συνάφεια τους)

Μηχανές Αναζήτησης

Φάση 1: Δημιουργία της μηχανής αναζήτησης

- Δημιουργία συλλογής (αν δεν υπάρχει):
  - crawl, scrap, use specific APIs (social media)
- Ανάλυση του κειμένου (parsing)
  - Ποιοι θα είναι οι όροι, περιλαμβάνει και γλωσσολογική επεξεργασία
- Κατασκευή ευρετηρίων (indexing)
  - 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

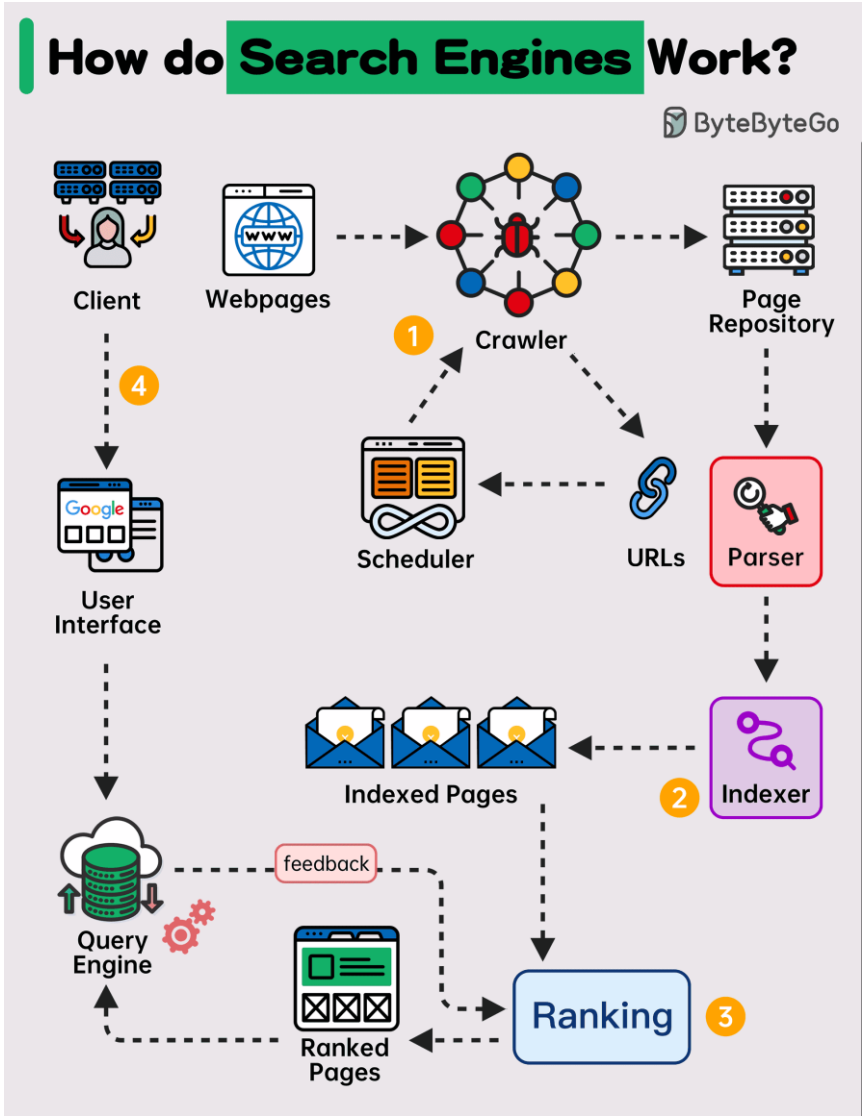
Ο χρήστης υποβάλει ένα ερώτημα

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:

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



https://blog.bytebytego.com/p/ep104-how-do-search-engines-work

Μηχανές Αναζήτησης

Αξιολόγηση

Ποιότητα των αποτελεσμάτων με βάση τη συνάφεια τους με το ερώτημα
Τυπικά με χρήση benchmark

Πολλές διαφορετικές μετρικές, δύο βασικές

- *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

Παράδειγμα

Έστω μια συλλογή με 1.000 έγγραφα. Ο χρήστης υποβάλει μια ερώτηση για την οποία υπάρχουν 20 συναφή έγγραφα. Το σύστημα επιστρέφει 15 έγγραφα από τα οποία το 12 είναι συναφή και τα 3 δεν είναι συναφή.

Βασικά Βήματα

(προεπεξεργασία)

- Δημιουργία συλλογής (corpus, collection, repository): σύλλεξε τα έγγραφα (προαιρετικά)
- Ανάλυση (parsing): εύρεση όρων
- Κατασκεύασε βοηθητικές δομές (indexing) – ευρετήρια

(λειτουργία)

- Επεξεργασία ερωτήσεων

Βασικά Βήματα

Για το απλούστερο μοντέλο (Boolean)
όροι συνδεδεμένοι με AND, OR, NOT

(προεπεξεργασία)

- Δημιουργία συλλογής (corpus, repository): σύλλεξε τα έγγραφα (προαιρετικά)
- Ανάλυση (parsing): εύρεση όρων
- Κατασκεύασε βοηθητικές δομές (indexing) – ευρετήρια

(λειτουργία)

- Επεξεργασία ερωτήσεων

Το απλό Boolean μοντέλο

d1 a b a a d
d2 b d e g
d3 a c c d
d4 a b c d g g
d5 a a e 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

Τι θα δούμε σήμερα;

Προ-επεξεργασία (parsing) για τη δημιουργία του λεξιλογίου όρων

ΜΥΕ003: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Προεπεξεργασία

Ακαδημαϊκό Έτος 2024-2025

Βασικά Βήματα

(προεπεξεργασία)

- Δημιουργία συλλογής (corpus, repository): σύλλεξε τα έγγραφα (προαιρετικά)
- Ανάλυση (parsing): εύρεση όρων
- Κατασκεύασε βοηθητικές δομές (indexing) – ευρετήρια

(λειτουργία)

- Επεξεργασία ερωτήσεων

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 1η 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.

Τους όρους που θα εισάγουμε στο ευρετήριο

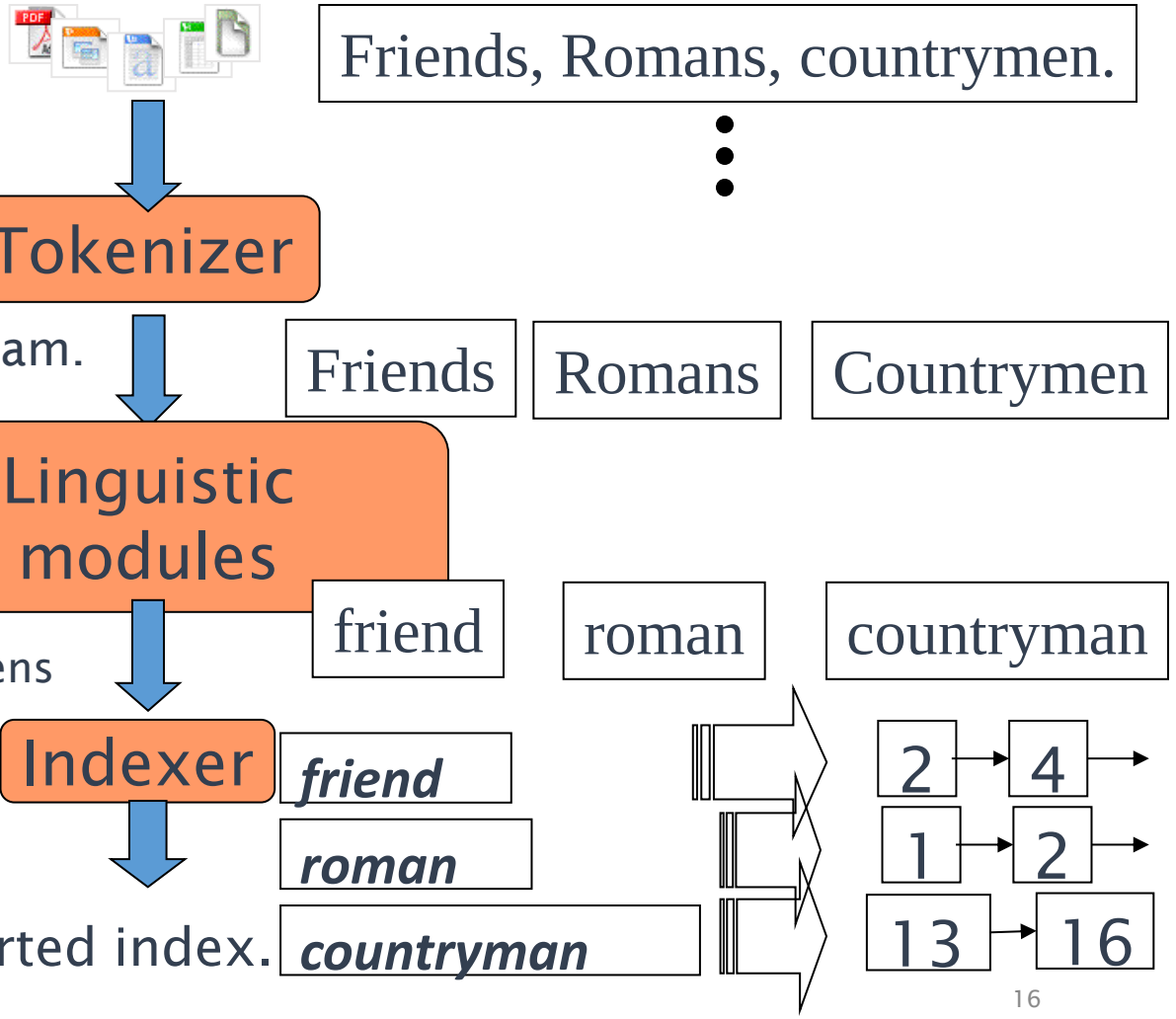
Τα βασικά βήματα για την κατασκευή του ευρετηρίου

1. Συλλέγουμε τα έγγραφα που θέλουμε να συμπεριλάβουμε στο ευρετήριο

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

3. Γλωσσολογική προ-επεξεργασία 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

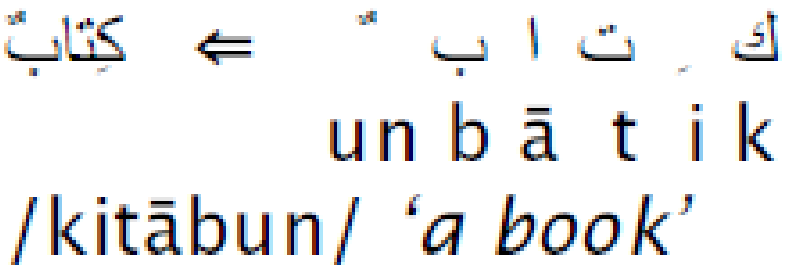
- Να αγνοήσουμε τα ειδικά σύμβολα (mark up)
  - JSON, XML
  - &amp -> & (XML)

Complications: Format/language

- Τα έγγραφα για τα οποία κατασκευάζουμε το ευρετήρια 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



Αραβικά: δισδιάστατη ακολουθία χαρακτήρων και χαρακτήρες σε μεικτή σειρά (φωνήεντα διακριτικά σημεία πάνω και κάτω από τα γράμματα, μεταλλάξεις όπως συνδυάζονται)
Από δεξιά στα αριστερά
Η αντίστοιχη *ακουστική* γραμμική ακολουθία
Πιθανή απουσία φωνηέντων (τα σύντομα απουσιάζουν ή εμφανίζονται κατά βούληση)

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 χωρίζουνε ένα αρχείο σε πολλά, (PPT ή LaTeX σε πολλαπλές HTML σελίδες) ίσως ένωση τους

Μονάδα εγγράφου

Αναλυτικότητα ευρετηρίασης (indexing granularity)

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

Προσέγγιση 1: Μονάδα = κεφάλαιο
Προσέγγιση 2: Μονάδα = βιβλίο

Ακρίβεια/ανάκληση

Πρόβλημα με μεγάλα έγγραφα, θα δούμε πληροφορία εγγύτητας

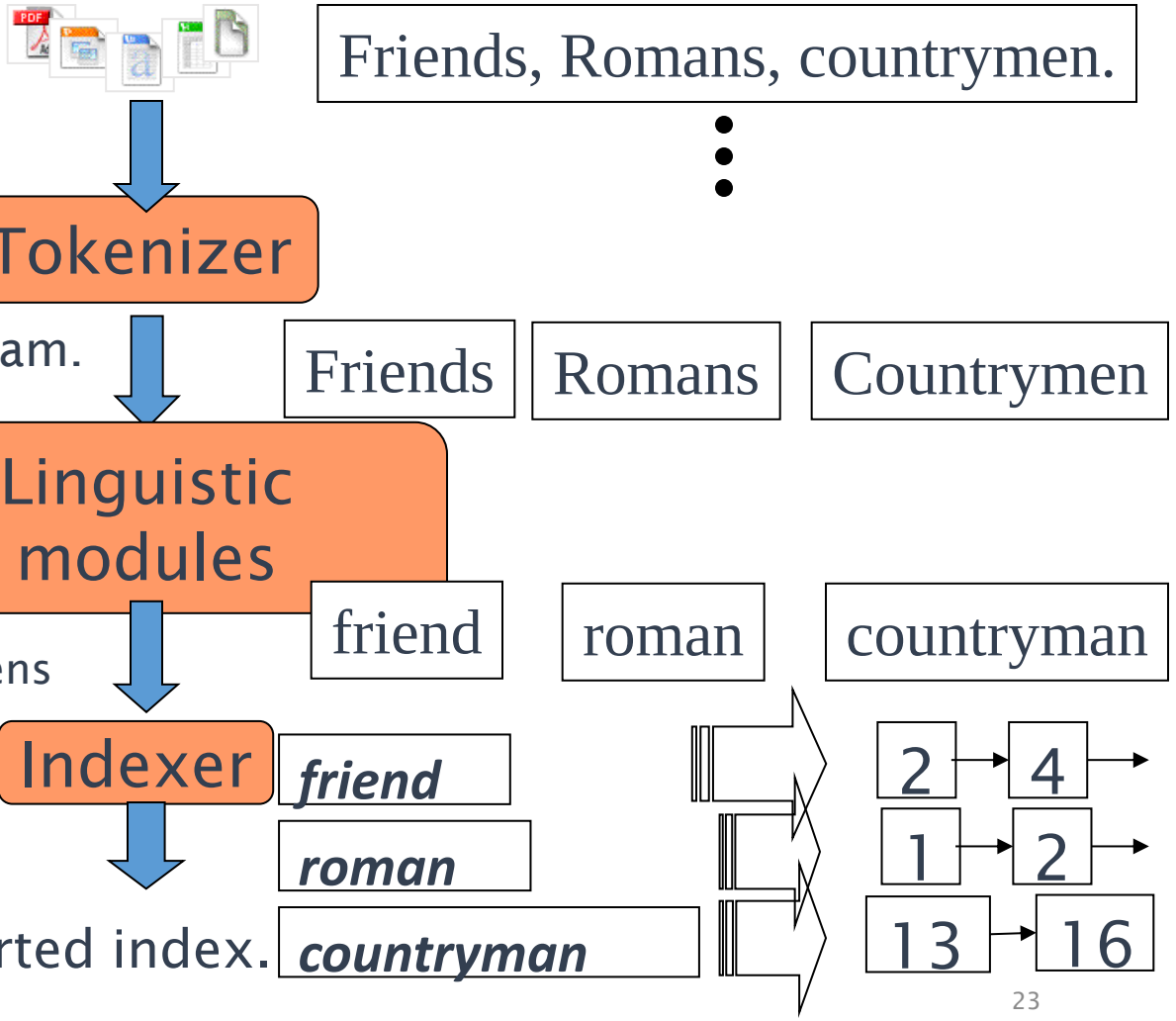
Τα βασικά βήματα για την κατασκευή του ευρετηρίου

1. Συλλέγουμε τα έγγραφα που θέλουμε να συμπεριλάβουμε στο ευρετήριο

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

3. Γλωσσολογική προ-επεξεργασία των συμβόλων

4. Ευρετηριάζουμε τα έγγραφα στα οποία περιλαμβάνεται κάθε όρος



Tokenization

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

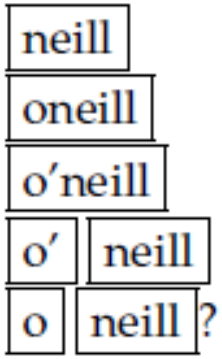
Αλλά ποια είναι τα κατάλληλα tokens;

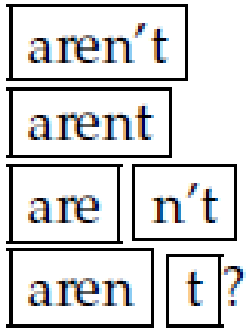
Αρκεί να χωρίσουμε το κείμενο στα κενά και στα σημεία στίξης;

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
- Ή και συνδυασμός
  - ***San Francisco-Los Angeles***

- ✓ lower case, lower-case, lowercase: ως “phrase queries” αν θέλουμε να επιστρέφουν το ίδιο αποτέλεσμα
- ✓ Μερικά εξειδικευμένα συστήματα ζητούν οι χρήστες πάντα το ‘–’ στην ερώτηση όταν θέλουν να εξεταστούν όλες οι περιπτώσεις
- ✓ Την ίδια πολιτική και στην ερώτηση και στο κείμενο

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#
  - C.A.T («κακό» ακρώνυμο)
- Αλλά και email και web, IP διευθύνσεις, κλπ δε θέλουμε να τις «σπάσουμε»

Άλλες γλώσσες

Percentages of the top 10 million websites on the World Wide Web using various content languages as of Feb 18, 2025

t

Rank	Language	15 May 2023	05 January 2025
1	English	55.5%	49.3%
2	Spanish	5.0%	6.0%
3	Russian	4.9%	3.9%
4	German	4.3%	5.6%
5	French	4.4%	4.4%
6	Japanese	3.7%	5.1%
7	Portuguese	2.4%	3.8%
8	Turkish	2.3%	1.8%
9	Italian	1.9%	2.7%
10	Persian	1.8%	1.2%
11	Dutch	1.5%	2.2%
12	Polish	1.4%	1.8%
13	Chinese	1.4%	1.2%
14	Vietnamese	1.3%	1.1%
15	Indonesian	0.7%	1.1%
16	Czech	0.7%	1%
17	Korean	0.7%	0.8%
18	Ukrainian	0.6%	0.6%
19	Arabic	0.7%	0.5%
20	Greek	0.5%	0.5%

Πηγή: wikipedia

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
- 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

- 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 ένα μοναδικό tokenization

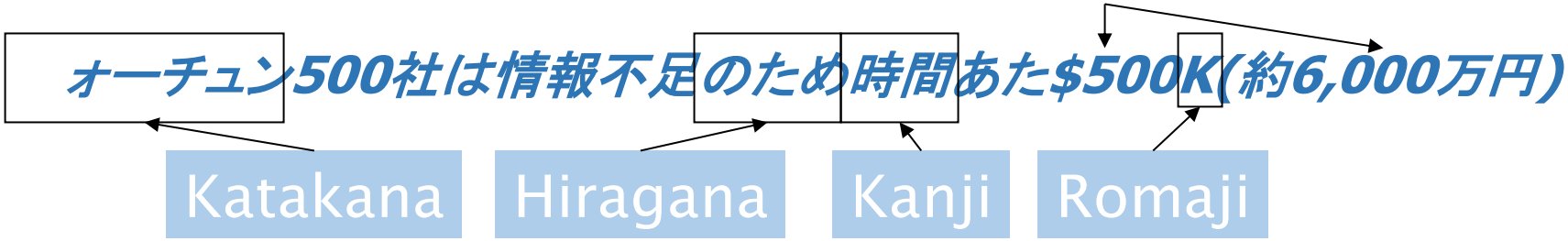
Tokenization: άλλες γλώσσες

和尚

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

Tokenization: άλλες γλώσσες

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



Γιαπωνέζικα - 4 διαφορετικά “αλφάβητα”:

- ***Chinese*** *characters,* ***hiragana*** *for inflectional endings and functional words,* ***katakana*** *for transcription of foreign words and other uses, and* ***latin***
- *δεν υπάρχουν κενά (όπως στα Κινέζικα).*
- *Οι χρήστες μπορεί μια 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

k-grams

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

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* (στάθμιση όρων και διάταξη όρων στα ευρετήρια με βάση τη σημαντικότητα) μειώνουν το κόστος στην εκτέλεσης 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”

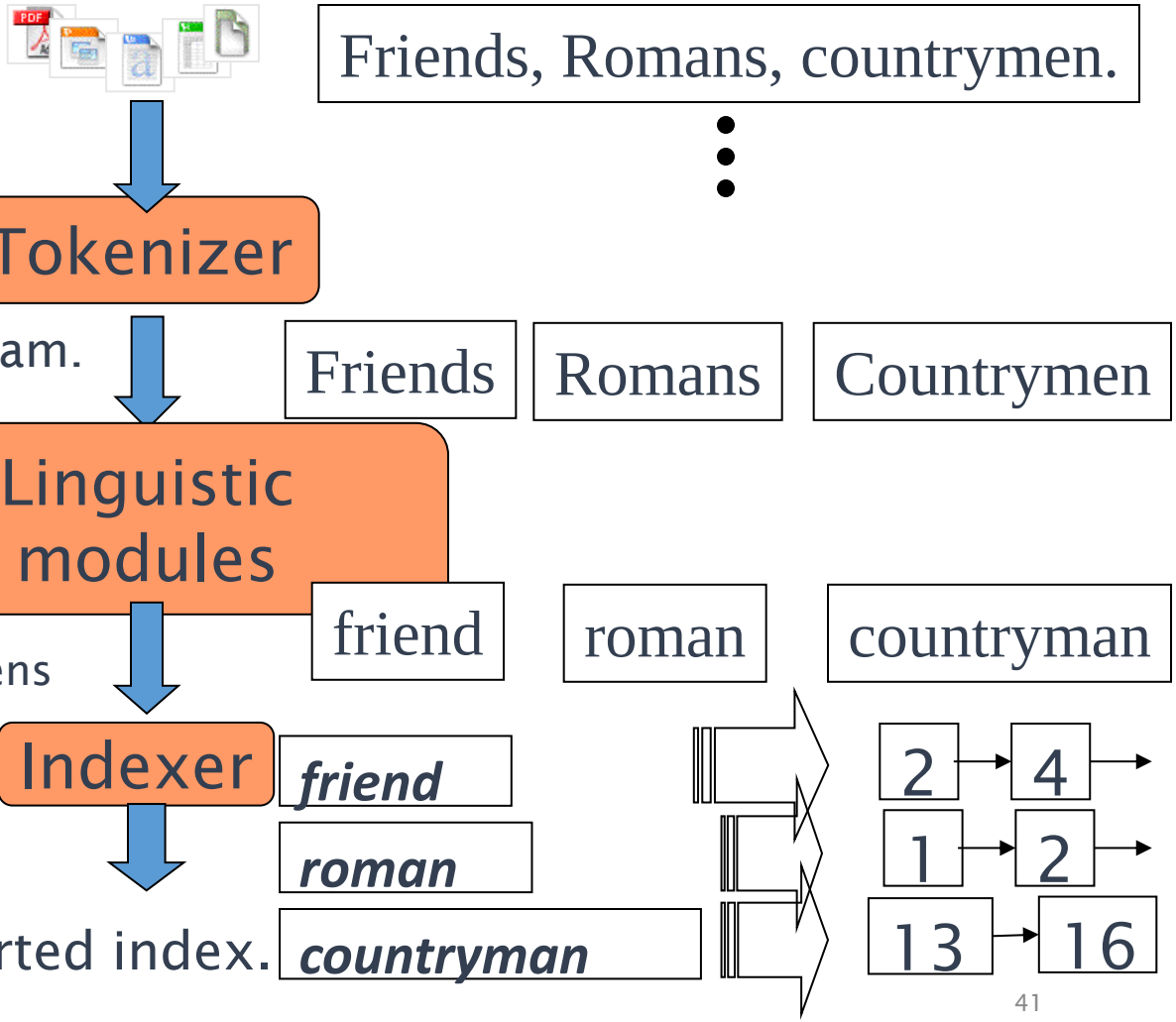
Τα βασικά βήματα για την κατασκευή του ευρετηρίου

1. Συλλέγουμε τα έγγραφα που θέλουμε να συμπεριλάβουμε στο ευρετήριο

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

3. Γλωσσολογική προ-επεξεργασία των συμβόλων

4. Ευρετηριάζουμε τα έγγραφα στα οποία περιλαμβάνεται κάθε όρος



Token normalization

Κανονικοποίηση (Token normalization)

- Token normalization: κανονικοποίηση των token ώστε να εντοπίζονται αντιστοιχίες και να ταιριάζουν tokens παρά κάποιες μικρές διαφορές (πχ U.S.A. USA)

- 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 (πχ, απλώς σβήνουμε το -)
- 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. Κατά την ερώτηση: (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***

- Λίστες πιο ισχυρές από τις κλάσεις αλλά λιγότερο αποδοτικές
- Είναι η κανονικοποίηση πάντα καλή, U.S.A, C.A.T?

Κανονικοποίηση: άλλες γλώσσες, τόνοι, διακριτικά

- Accents: π.χ., Γαλλικά ***résumé*** vs. ***resume*.**
- Umlauts: π.χ., Γερμανικά: ***Tuebingen*** vs. ***Tübingen***
  - Πρέπει να είναι ισοδύναμα

- Τόνοι στα Ελληνικά

Κανονικοποίηση: άλλες γλώσσες, τόνοι, διακριτικά

- Πιο σημαντικό 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 κανονικοποίηση μπορεί να εξαρτώνται 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

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)

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”

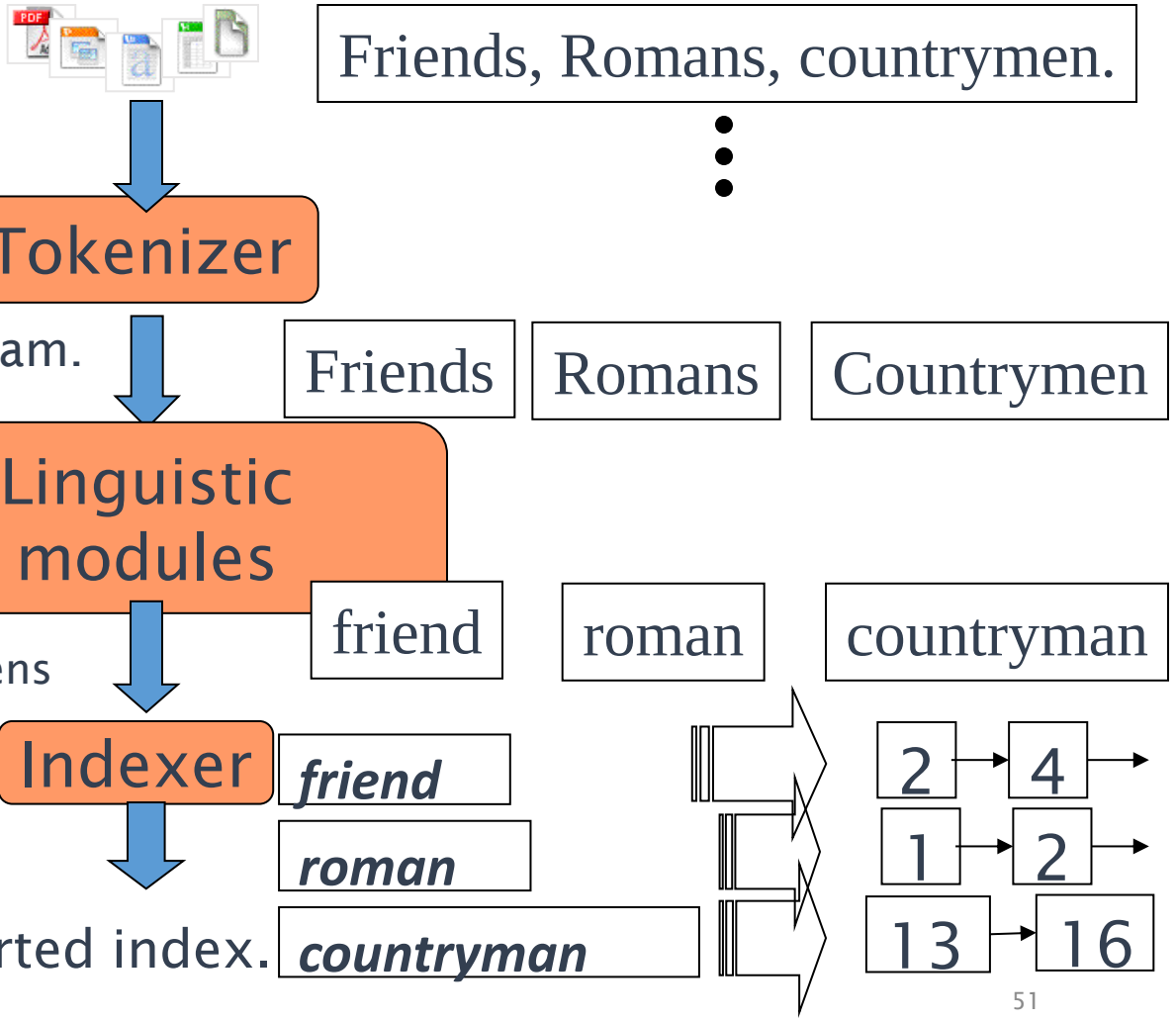
Τα βασικά βήματα για την κατασκευή του ευρετηρίου

1. Συλλέγουμε τα έγγραφα που θέλουμε να συμπεριλάβουμε στο ευρετήριο

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

3. Γλωσσολογική προ-επεξεργασία 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* (λημματοποίση)
- *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, Φιλανδικά
  - 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 analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to pictur of express that is more biolog transpar and access to interpret

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

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

Περίληψη

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 1η 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?

1. In a Boolean retrieval system, stemming never lowers precision.

1Α Σωστό 1Β Λάθος

2. In a Boolean retrieval system, stemming never lowers recall.

2Α Σωστό 2Β Λάθος

Επανάληψη (ερωτήσεις)

Άσκηση 2.1

Are the following statements true or false?

3. Stemming increases the size of the vocabulary

3A Σωστό 3B Λάθος

4. Stemming should be invoked at indexing time but not while processing a query.

4A Σωστό 4B Λάθος

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

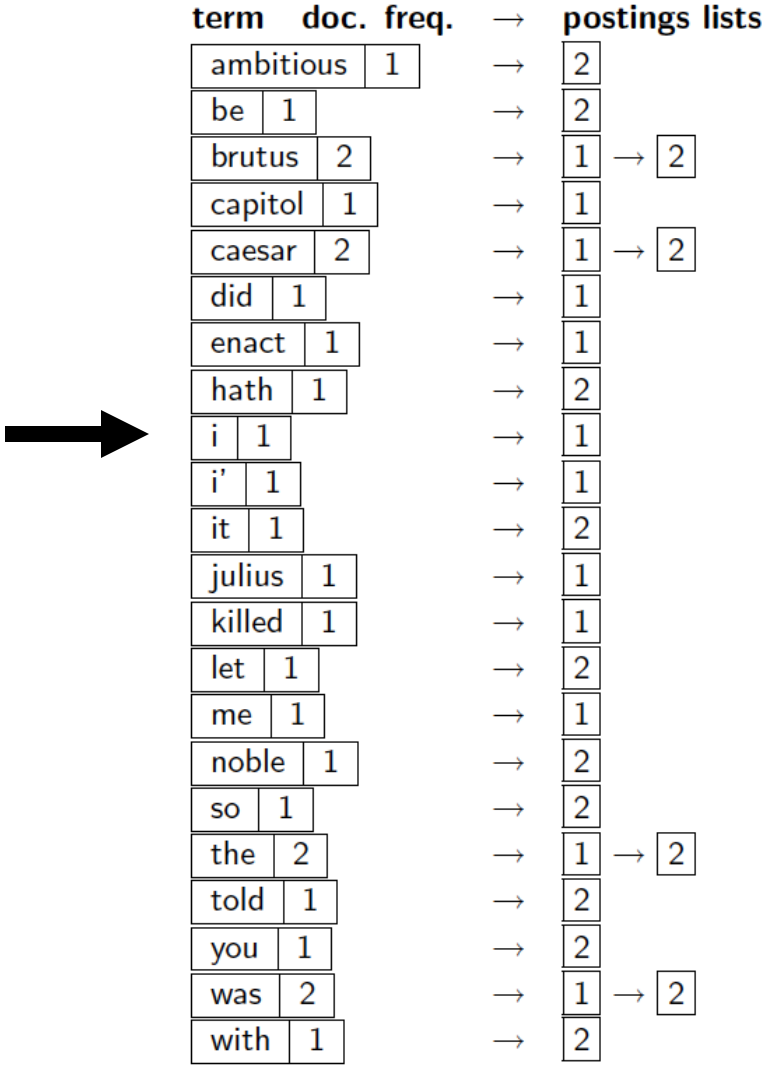
Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

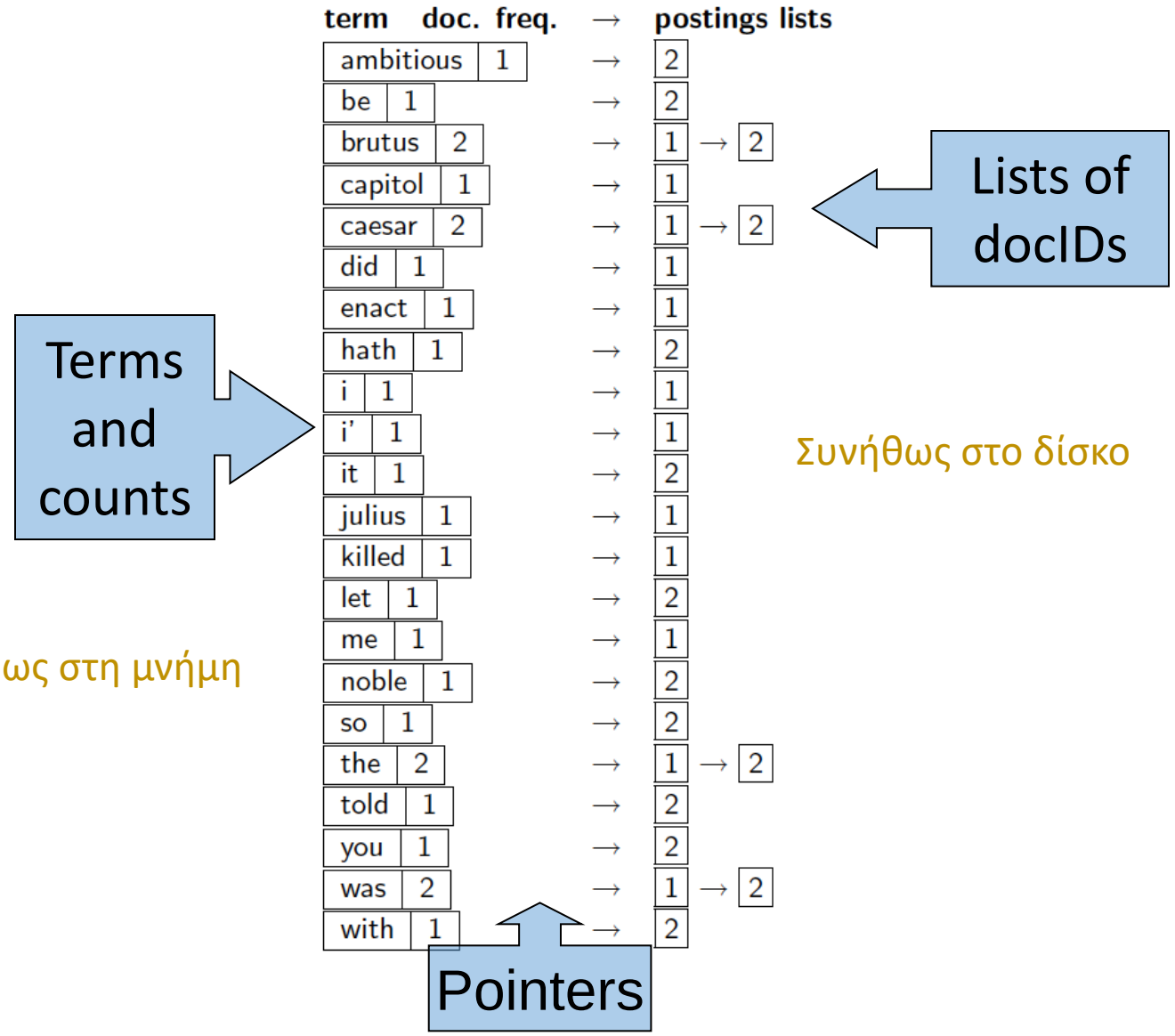
Βήματα του Indexer: Λεξικό & Καταχωρήσεις

- 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)

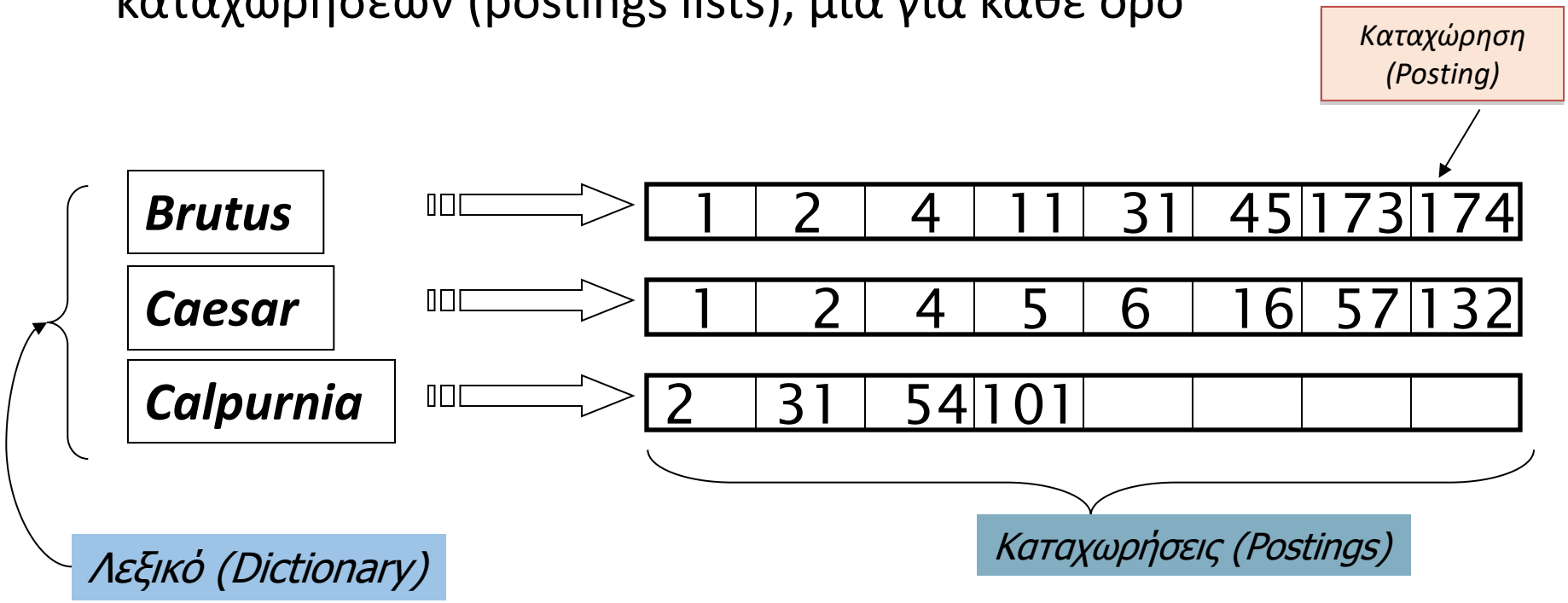
Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2




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
Terms and counts
Lists of docIDs
Pointers
Συνήθως στο δίσκο
Συνήθως στη μνήμη

Βασικές Δομές

- Λεξικό
- Ανεστραμμένο ευρετήριο: αποτελείται από λίστες καταχωρήσεων (postings lists), μία για κάθε όρο



Σε διάταξη με βάση το docID

ΜΥΕ003: Ανάκτηση Πληροφορίας

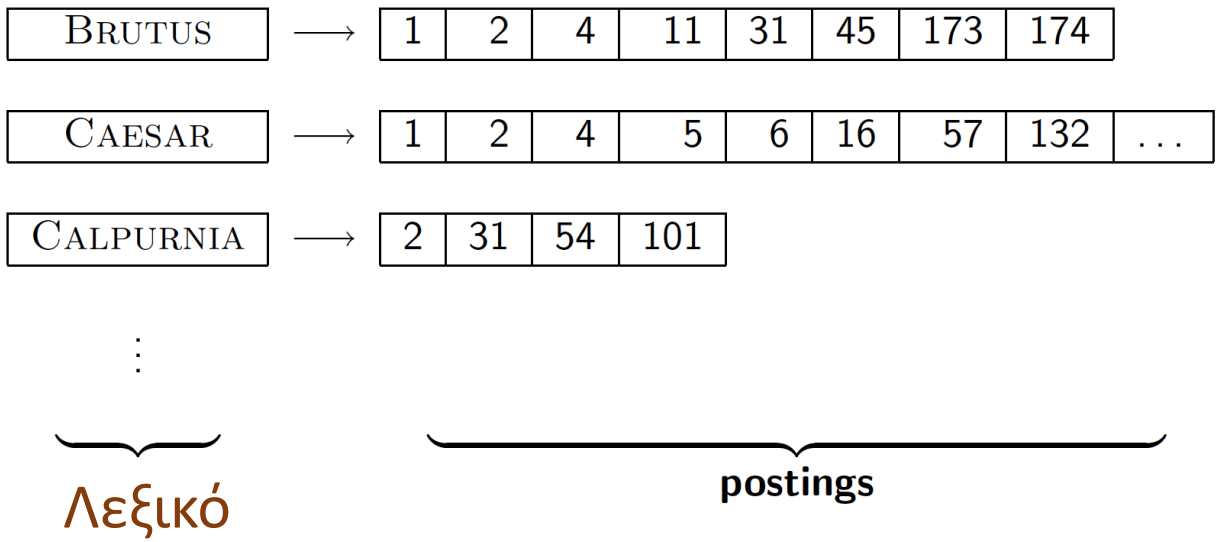
Διδάσκουσα: Ευαγγελία Πιτουρά

Δομές για Λεξικά

Ακαδημαϊκό Έτος 2024-2025

Δομές Δεδομένων για Λεξικά

- Το *λεξικό* περιέχει: το *λεξιλόγιο όρων* -- για κάθε όρο: τη συχνότητα εγγράφου (document frequency) και δείκτη στη λίστα καταχωρήσεων
- Σε κάθε ερώτημα, *αναζήτηση στο λεξικό* αν υπάρχει ο όρος και σε ποιες λίστες



Ποια δομή δεδομένων είναι κατάλληλη;

Δομές Δεδομένων για Λεξικά

Λεξιλόγιο (vocabulary): το σύνολο των όρων

Λεξικό (dictionary): μια δομή για την αποθήκευση του λεξιλογίου

Πως αποθηκεύουμε ένα λεξικό (στη μνήμη) αποδοτικά;

Μια απλοϊκή λύση

- array of struct:

term	document frequency	pointer to postings list
a	656,265	⟶
aachen	65	⟶
. . .	. . .	. . .
zulu	221	⟶

char[20] int Postings *

20 bytes 4/8 bytes 4/8 bytes

- Πως αναζητούμε έναν όρο (κλειδί, key) στο λεξικό γρήγορα κατά την εκτέλεση του ερωτήματος;

Δομές Δεδομένων για Λεξικά

Κριτήρια για την επιλογή δομής:

(ποιες λειτουργίες, workload, μέγεθος)

- Αποδοτική αναζήτηση ενός όρου (κλειδιού) στο λεξικό.
- Είναι στατικό ή έχουμε συχνά εισαγωγές/διαγραφές όρων ή και τροποποιήσεις; Μόνο εισαγωγές (insert only – append only)
- Πόσοι είναι οι όροι
- Σχετικές συχνότητες προσπέλασης των κλειδιών (πιο γρήγορα οι συχνοί όροι;)

Δομές Δεδομένων για το Λεξικό

- Δυο βασικές επιλογές:
  - Πίνακες Κατακερματισμού (Hashtables)
  - Δέντρα (Trees)

- Μερικά συστήματα ανάκτησης πληροφορίας χρησιμοποιούν πίνακες κατακερματισμού άλλα δέντρα

Πίνακες Κατακερματισμού

Κάθε όρος του λεξιλογίου κατακερματίζεται σε έναν ακέραιο

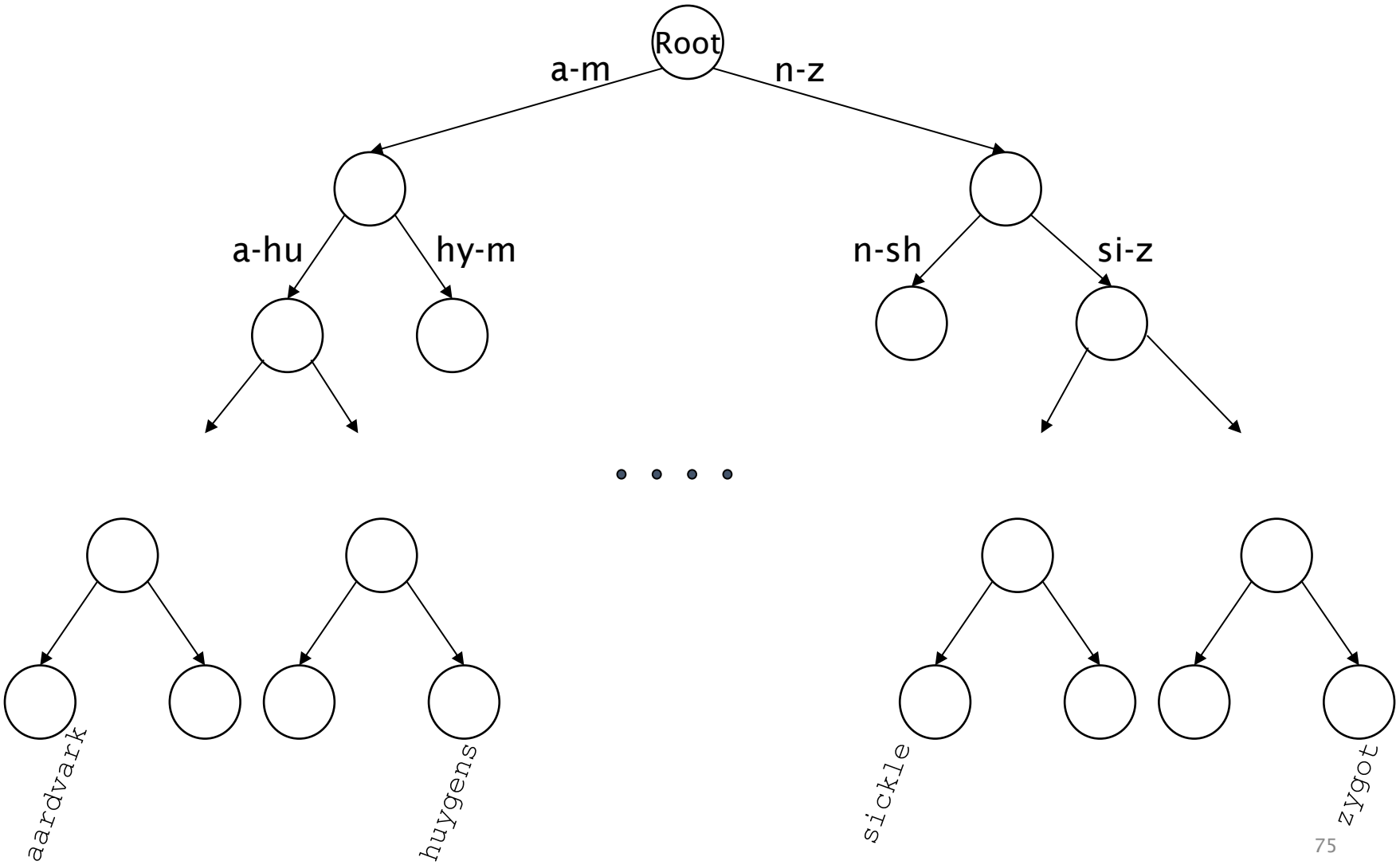
+:

- Η αναζήτηση είναι πιο γρήγορη από ένα δέντρο: O(1)

- :

- Δεν υπάρχει εύκολος τρόπος να βρεθούν μικρές παραλλαγές ενός όρου
  - judgment/judgement, *resume* vs. *résumé*
- Μη δυνατή η προθεματική αναζήτηση [ανεκτική ανάκληση]
- Αν το λεξιλόγιο μεγαλώνει συνεχώς, ανάγκη για να γίνει κατακερματισμός από την αρχή

Δέντρα Αναζήτησης: Δυαδικό Δέντρο

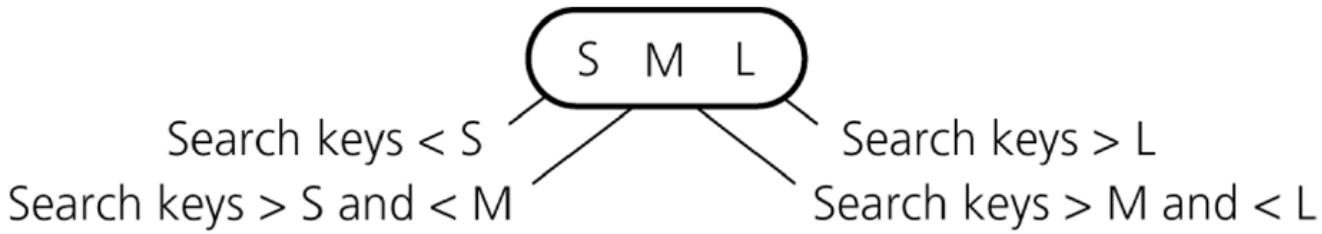


Δέντρα Αναζήτησης: Δυαδικό Δέντρο

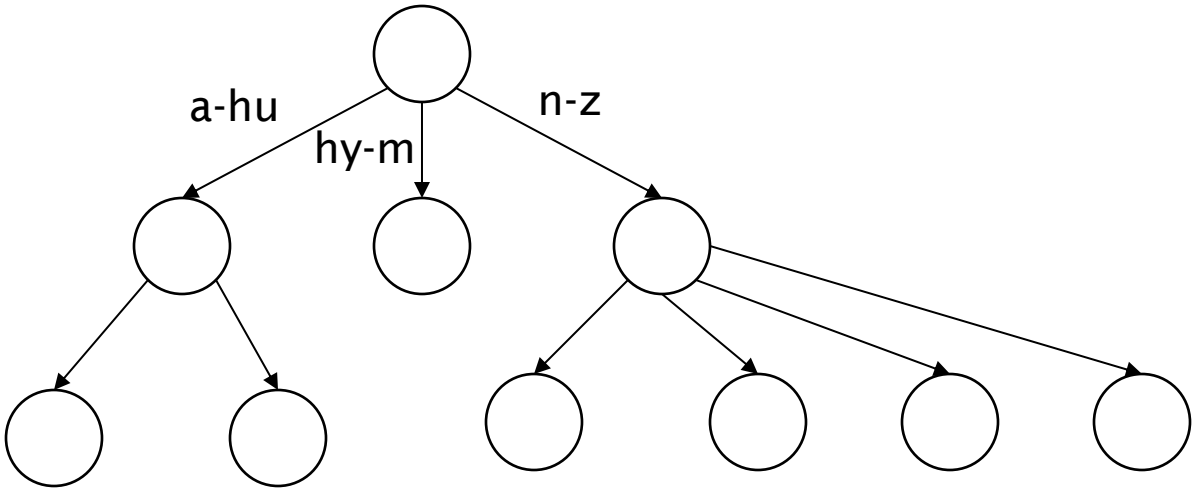
- O(log M), M: αριθμός των όρων (το μέγεθος του λεξικού)
  - προϋποθέτει ισοζύγιση

Δέντρα: B-δέντρα

Ορισμός: Κάθε εσωτερικός κόμβος έχει έναν αριθμό από παιδιά στο διάστημα [*a*, *b*] όπου *a, b* είναι κατάλληλοι φυσικοί αριθμοί, π.χ., [2, 4]



Δέντρα: B-δέντρα



Ορισμός: Κάθε εσωτερικός κόμβος έχει έναν αριθμό από παιδιά στο διάστημα [*a*, *b*] όπου *a, b* είναι κατάλληλοι φυσικοί αριθμοί, π.χ., [2, 4]

Δέντρα

- Το απλούστερο: δυαδικό δέντρο
- Το πιο συνηθισμένο: B-δέντρα
- Τα δέντρα απαιτούν ένα δεδομένο τρόπο διάταξης των χαρακτήρων (αλλά συνήθως υπάρχει ή μπορεί να οριστεί)

+:

- Λύνουν το πρόβλημα προθέματος (π.χ., όροι που αρχίζουν με *hyp*)
- Πλεονεκτούν όταν το λεξικό αποθηκεύεται στο δίσκο (τότε τα *a* και *b* καθορίζονται από το μέγεθος του block)

-:

- Πιο αργή: O(log *M*) [και αυτό απαιτεί (*ισοζυγισμένα* (*balanced)* δέντρα)
- Η επανα-ισοζύγιση (rebalancing) των δυαδικών δέντρων είναι ακριβή
  - Αλλά τα B-δέντρα καλύτερα