Introduction to

Information Retrieval

ΠΛΕ70: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Διάλεξη 3: Δομές για Λεξικά. Ανάκτηση Ανεκτική στα Σφάλματα (υποστήριξη *)

Επανάληψη προηγούμενης διάλεξης

1. Προ-επεξεργασία εγγράφων της συλλογής για την κατασκευή του αντεστραμμένου ευρετηρίου
2. Πιο γρήγορες λίστες καταχώρησης με λίστες παράλειψης
3. Υποστήριξη ερωτημάτων φράσεων (phrase queries) και θέσης (positional queries)

1. Προσδιορισμός Λεξιλογίου όρων

1. Συλλέγουμε τα έγγραφα για τα οποία θα κατασκευαστεί το ευρετήριο

Friends, Romans, countrymen. | So let it be with Caesar | . . .

2. Tokenize το κείμενο, αποτέλεσμα: μια λίστα από tokens:

Friends | Romans | countrymen | So | . . .

3. Γλωσσική επεξεργασία ώστε να παραχθεί μια λίστα από κανονικοποιημένα tokens που θα είναι οι όροι που εισαχθούν στο ευρετήριο

countryman | so | . . .

friend | roman

4. Κατασκευή αντεστραμμένου ευρετηρίου

1. Προσδιορισμός Λεξιλογίου όρων

- Token – η εμφάνιση μια λέξης ή ενός όρου σε ένα έγγραφο
- Type (τύπος) – μια κλάση ισοδυναμίας από tokens
  - *Παράδειγμα: In June, the dog likes to chase the cat in the barn.*
  - 12 word tokens, 9 word types

Tokenization - Προβλήματα

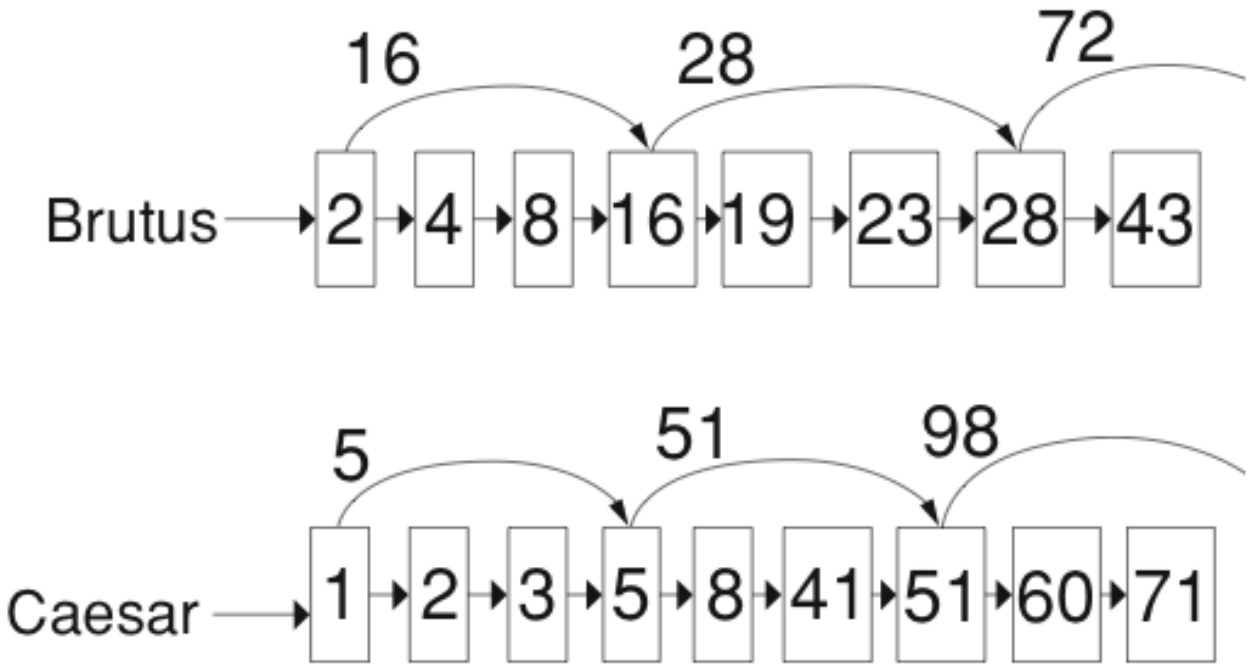
➢ Ποια είναι τα διαχωριστικά (κενό, απόστροφος, ενωτικά (hyphen))

1. Προσδιορισμός Λεξιλογίου όρων

Κλάσεις ισοδύναμων tokens -> όρους που θα εισαχθούν στο ευρετήριο

- Αριθμοί
- Κεφαλαία/Μικρά
- Λημματοποίηση και Περιστολή (Stemming)
- Stop words?
- Κλάσεις ισοδύναμων όρων (για συνώνυμα) κατά την επεξεργασία του ερωτήματος ή στο ευρετήριο

2. Δείκτες παράλειψης



3. Ερωτήματα Φράσεων και Θέσης

- Ευρετήρια Biword για ερωτήματα φράσεων
- Ευρετήρια Θέσης (positional indexes) για ερωτήματα φράσεων και θέσης (γειτονικότητας)

3. Ερωτήματα Φράσεων και Θέσης

- Στις λίστες καταχωρήσεων σε ένα nonpositional ευρετήριο, κάθε καταχώρηση είναι μόνο ένα docID
- Στις λίστες καταχωρήσεων σε ένα positional ευρετήριο, κάθε καταχώρηση είναι ένα docID και μια λίστα από θέσεις
- Παράδειγμα ερωτήματος: *"to$_1$ be$_2$ or$_3$ not$_4$ to$_5$ be$_6$"*

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›; . . . ›

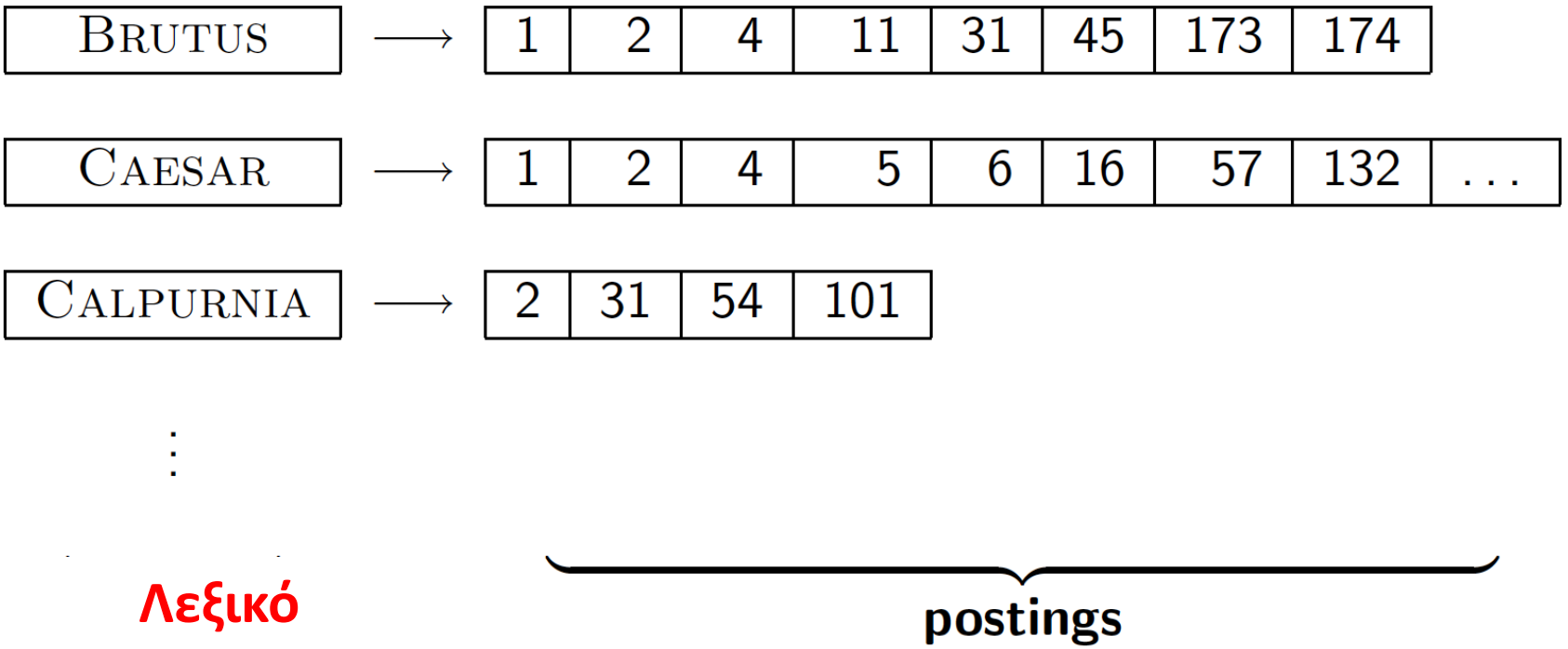
BE, 178239:
‹ 1: ‹17, 25›;
4: ‹17, 191, 291, 430, 434›;
5: ‹14, 19, 101›; . . . ›

Τι θα δούμε σήμερα;

- Δομές δεδομένων για Λεξικά
- Ανάκτηση Ανεκτική σε Σφάλματα “Tolerant”
  - Ερωτήματα με Wild-card («χαρακτήρων μπαλαντέρ)*

Δομές Δεδομένων για Λεξικά

- Οι δομές δεδομένων για το λεξικό περιέχουν το *λεξιλόγιο όρων (λήμμα)*, τη συχνότητα εγγράφου (document frequency), δείκτες σε κάθε λίστα καταχωρήσεων … 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;

Πως το χρησιμοποιούμε;

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 (λήμμα) στο λεξικό γρήγορα κατά την εκτέλεση του ερωτήματος;
  - ο όρος είναι το κλειδί (σε ορολογία δομών δεδομένων)

Δομές Δεδομένων για Λεξικά

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

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

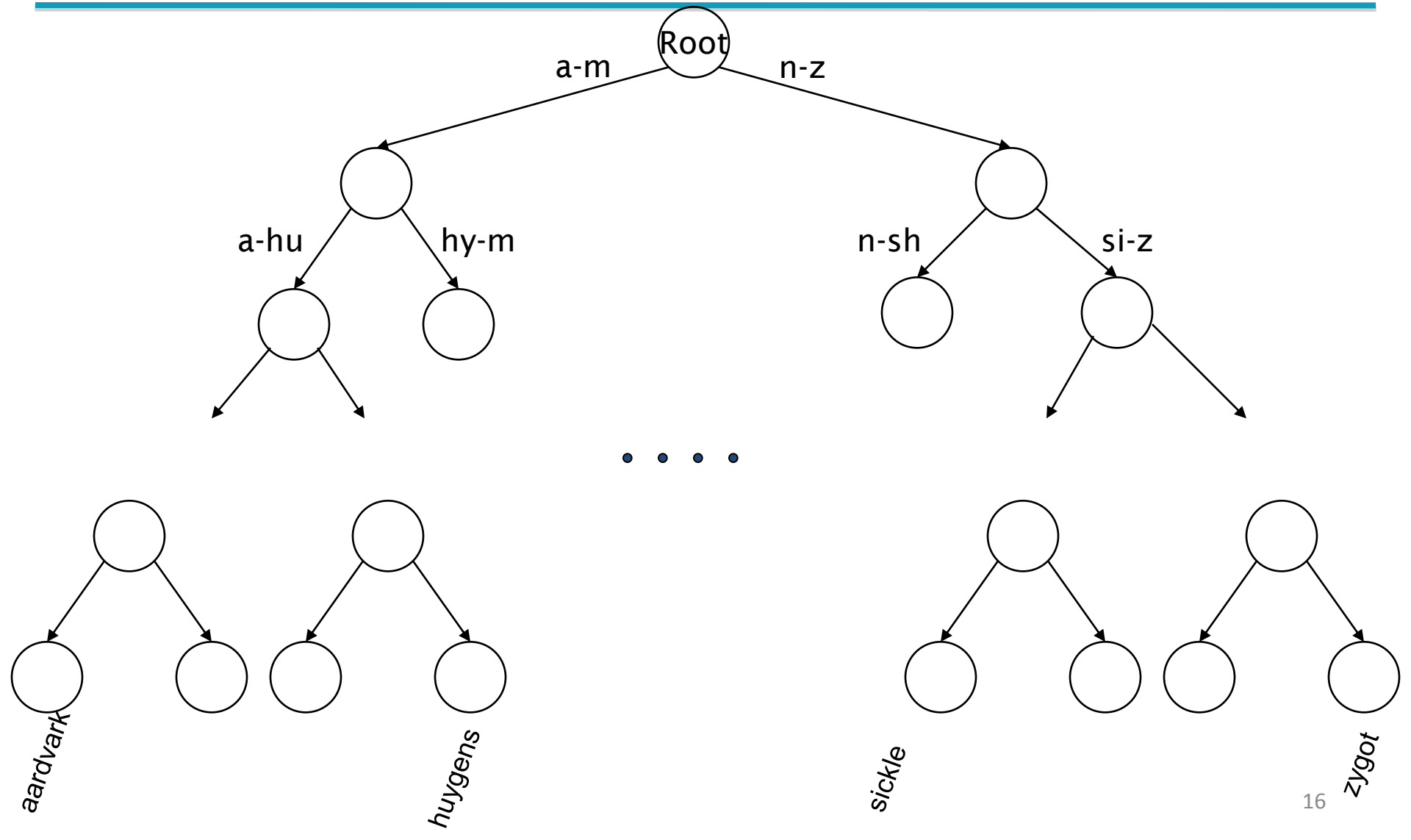
+:

- 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

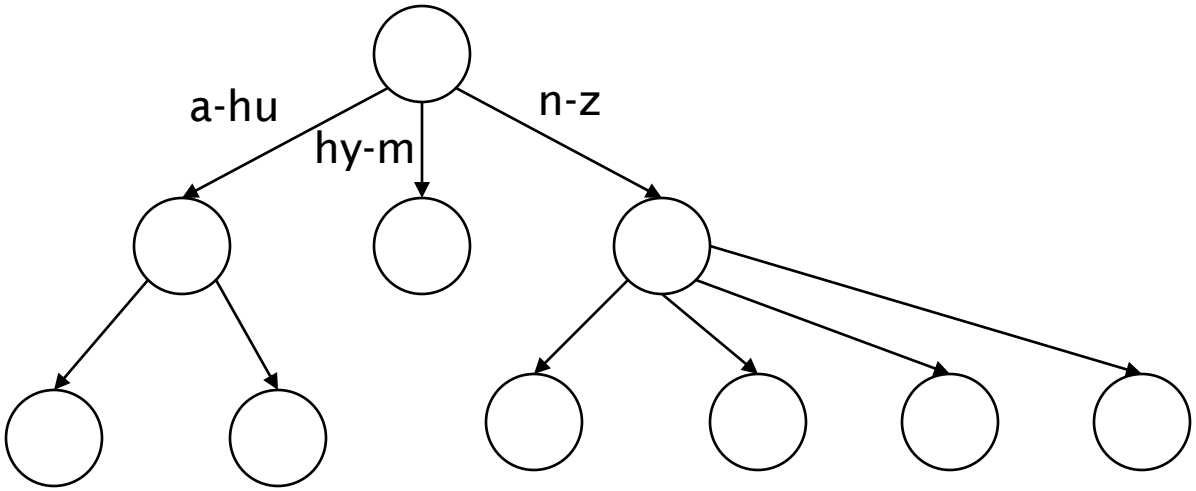
Δέντρα αναζήτησης: Δυαδικό δέντρο



Δέντρα αναζήτησης: Δυαδικό δέντρο

- O(log M), M: αριθμός των όρων (το μέγεθος του λεξικού)
- Ανάγκη για 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

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

+:

- Λύνουν το πρόβλημα προθέματος (π.χ., όροι που αρχίζουν με *hyp*)

-:

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

ΕΡΩΤΗΜΑΤΑ ΜΕ *

Ερωτήματα με Wild-card (*)

- Δεν είμαστε σίγουροι για την ορθογραφία της λέξης
- Πολλαπλές εκδοχές της ορθογραφίας της λέξης
- Δεν είμαστε σίγουροι αν έχει γίνει stemming
- Ορθογραφία ξένης λέξης (Σ*ξπ*ρ)

Ερωτήματα με Wild-card (*)

- ***mon*:** Βρες όλα τα έγγραφα που περιέχουν οποιαδήποτε λέξη *αρχίζει με "mon" (trailing wild card query).*
  - Εύκολο όταν το λεξικό με δυαδικό δέντρο (ή B-δέντρο):
    - ανάκτησε όλους τους όρους t στο διάστημα: ***mon ≤ t < moo***
    - Για κάθε όρο, αναζήτησε το αντεστραμμένο ευρετήριο σε ποια έγγραφα εμφανίζεται
- **** mon:*** Βρες όλα τα έγγραφα που περιέχουν οποιαδήποτε λέξη τελειώνει σε "mon" (leading wind card queries): *πιο δύσκολο*
  - Διατήρησε ένα επιπρόσθετο B-tree για τους όρους ανάποδα (*backwards), πχ ο όρος demon -> nomed*
  - Ανάκτησε όλους τους όρους t στο διάστημα: ***nom ≤ t < non***.

Ερωτήματα με Wild-card (*)

Πως μπορούμε να απαντήσουμε ερωτήσεις με ένα * στη μέση της λέξης, π.χ., ***pro*cent*** ?

+ διατρέχουμε τους όρους που ανήκουν στην τομή και απορρίπτουμε όσους ταιριάζουν και με το πρόθεμα και με το επίθημα (αρκεί; ba*ba και όρος ba?)

Επεξεργασία ερωτημάτων

- Π.χ., Θεωρείστε το ερώτημα:

  se*ate *AND* ***fil*er***

Μπορεί να οδηγήσει στην εκτέλεση πολλών Boolean *AND* ερωτημάτων (πιθανοί συνδυασμοί όρων).

Γενικά ερωτήματα με *

- * στη μέση του όρου
  - ***co*tion***
- Αναζήτησε το ***co**** AND ****tion*** σε ένα B-tree και υπολόγισε την τομή των συνόλων
  - Ακριβό!
- Δύο γενικές λύσεις

  Μετατροπή της ερώτησης q* σε Boolean ερώτηση Q σε ένα *ειδικό ευρετήριο* τέτοιο ώστε η απάντηση στο Q να είναι υπερσύνολο της απάντησης στο q* και στη συνέχεια ελέγχουμε

Γενικά ερωτήματα με *

- Πρώτη εναλλακτική λύση: Μετάτρεψε τις ερωτήσεις έτσι ώστε τα * να εμφανίζονται στο τέλος

Permuterm Index (ευρετήριο αντιμετατεθειμένων όρων)

Ευρετήριο Permuterm

Βασική ιδέα: Περιστροφή (rotation) του όρου του ερωτήματος ώστε το * στο τέλος

π.χ., Ερώτημα he*lo -> he*lo$ -> lo$he*

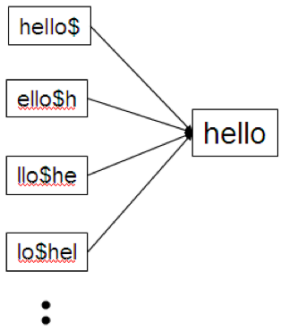
όπου $ ένα ειδικός χαρακτήρας που σηματοδοτεί το τέλος μιας λέξης

Ψάχνουμε το lo$hel*

Κατασκευάζουμε ένα ευρετήριο αντιμετατεθειμένων όρων στο οποίο οι διάφορες παραλλαγές που προκύπτουν από την περιστροφή του όρου συνδέονται με τον αρχικό

Πχ. για τον όρο **hello -> hello$**, εισάγουμε στο ευρετήριο τα:

- **hello$, $hello, o$hell , lo$hel (match), llo$he, ello$h**



Ευρετήριο Permuterm

Παράδειγμα

Ευρετήριο – όροι moron, man

Εισάγουμε στο λεξικό όλες τις περιστροφές των όρων να δείχνουν στον όρο στο αντεστραμμένο ευρετήριο

moron -> moron$ -> στο ευρετήριο: $moron, n$moro, on$mor ron$mo oron$m moron$

man -> man$ -> στο ευρετήριο: man, nma , anm, man

Ερώτημα

m*n -> m*n$ -> n$m*

Ερώτημα: mo*n -> n$mo*

Match?

Ερώτημα: m* -> $m*

Match?

Ευρετήριο Permuterm

- **X*Y*Z** πως γίνεται match?
  - X*Y*Z$ -> Z$X*
  - Ψάξε Z$X* και μετά έλεγξε κάθε υποψήφιο όρο για το Y
  - Πχ fi*mo*er -> ψάξε er$fi*, έλεγξε αν και mo (π.χ., fishmonger και fillbuster)
- Στην πραγματικότητα, permuterm B-tree
- *Πρόβλημα: ≈ δεκαπλασιάζει το μέγεθος του λεξικού*

Εμπειρική παρατήρηση για τα Αγγλικά

Ευρετήρια *k*-γραμμάτων (*k*-gram indexes)

k-gram: ακολουθία *k* χαρακτήρων

- Απαρίθμησε όλα τα *k*-γράμματα που εμφανίζονται σε κάθε όρο
  - *π.χ.,* για το κείμενο "***April is the cruelest month***" έχουμε τα 2-γράμματα (*bigrams*)

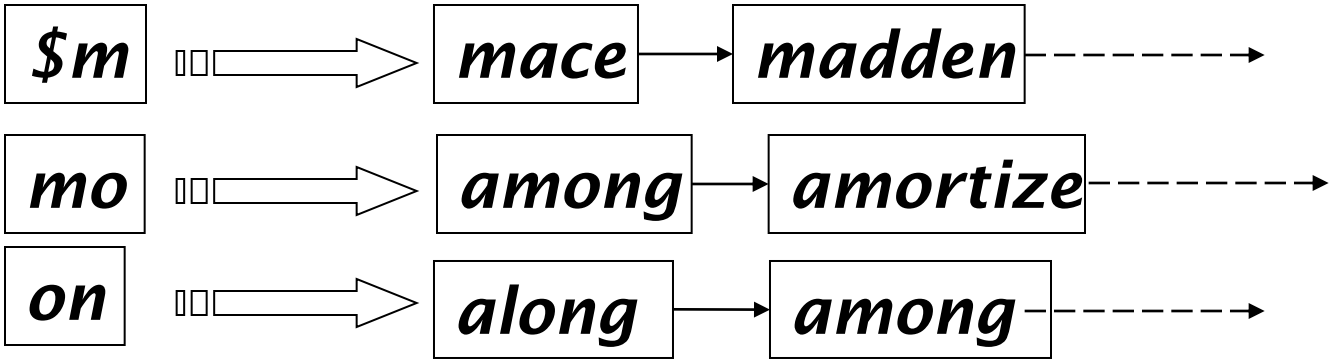
a,ap,pr,ri,il,l,i,is,s,t,th,he,e,$c,cr,ru,
ue,el,le,es,st,t$, m,mo,on,nt,h

  - Όπου $ ένα ειδικός χαρακτήρας που σηματοδοτεί το τέλος και την αρχή μιας λέξης

- Διατήρησε ένα δεύτερο αντεστραμμένο ευρετήριο από τα 2-γράμματα στους όρους του λεξικού που τα περιέχουν

Παράδειγμα 2-γράμματος

- Το ευρετήριο *k*-γραμμάτων βρίσκει τους όρους βασισμένο σε μια ερώτηση που αποτελείται από *k*-γράμματα (εδώ *k*=2).



k = 3

Επεξεργασία ερωτημάτων

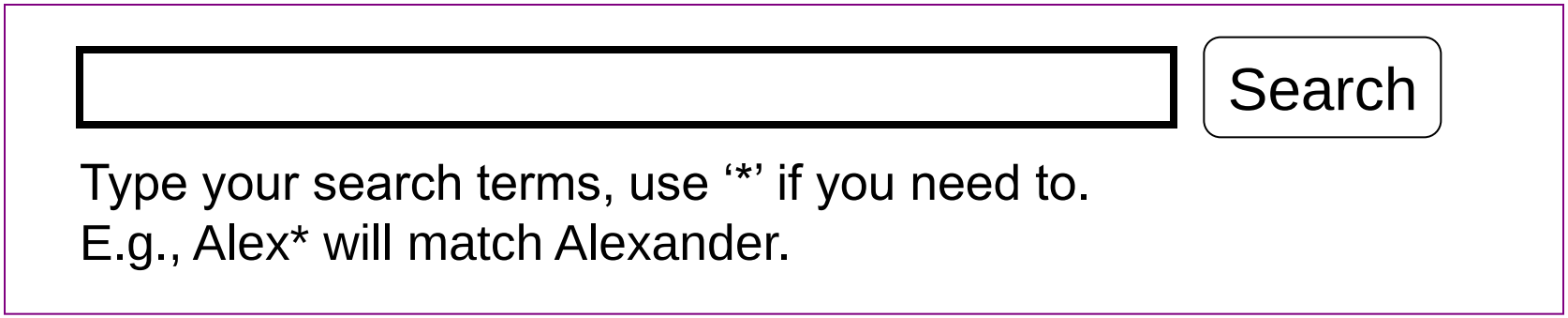
- Ερώτημα ***mon**** τώρα γίνεται

  $m** AND **mo** AND **on

  - Βρίσκει τους όρους που ταιριάζουν μια AND εκδοχή του wildcard ερωτήματος.
- Απαιτείται βήμα μετά-φιλτραρίσματος (post-filter)
  - False positive, π.χ., moon
- Οι όροι που απομένουν αναζητούνται στο γνωστό αντεστραμμένο ευρετήριο όρων-εγγράφων

Επεξεργασία ερωτημάτων

- Ένα Boolean ερώτημα για κάθε όρο
- Μπορεί να οδηγήσουν σε ακριβή επεξεργασία ερωτημάτων
  - pyth* AND prog*
- Αν ενθαρρύνουμε την “τεμπελιά” οι άνθρωποι θα ανταποκριθούν!



- Ποιες μηχανές αναζήτησης επιτρέπουν τέτοια ερωτήματα;

ΤΕΛΟΣ 3ου Μαθήματος

Ερωτήσεις?

Χρησιμοποιήθηκε κάποιο υλικό των:
- ✓*Pandu Nayak and Prabhakar Raghavan, CS276:Information Retrieval and Web Search (Stanford)*
- ✓*Hinrich Schütze and Christina Lioma, Stuttgart IIR class*