Ανάκληση Πληροφορίας

Information Retrieval

Διδάσκων –
Δημήτριος Κατσαρός

Διάλεξη 5η

WILD-CARD Ερωτήματα

Ερωτήματα με χαρακτήρες wild-card: *

- ***mon*:** να βρεθούν όλα τα έγγραφα που περιέχουν οποιαδήποτε λέξη αρχίζει με “mon”
- Εύκολο με binary tree (ή B-tree) lexicon: ανάκτηση όλων των λέξεων στην range: ***mon*** ≤ ***w*** < ***moo***
- ****mon:*** να βρεθούν όλα τα έγγραφα που περιέχουν οποιαδήποτε λέξη τελειώνει σε “mon”
  - Διατήρηση ενός ακόμη B-tree για τους όρους ανάστροφα

Μπορεί ν’ ανακτήσει όλες τις λέξεις στην range: ***nom*** ≤ ***w*** < ***non***.

Άσκηση: Με βάση αυτά, πώς μπορούμε να εντοπίσουμε όλες τις λέξεις που ικανοποιούν το wild-card ερώτημα ***pro*cent*** ?

Επεξεργασία ερωτήματος

- Μέχρι στιγμής επιτύχαμε την απαρίθμηση όλων των όρων του dictionary που ταιριάζουν με το wild-card ερώτημα
- Πρέπει όμως να ψάξουμε και την αντίστοιχη postings για κάθε τέτοιο όρο
- Π.χ., θεωρήστε το ερώτημα:

  se*ate** AND **fil*er

Μπορεί να οδηγήσει στην εκτέλεση πολλαπλών Boolean *AND* ερωτημάτων

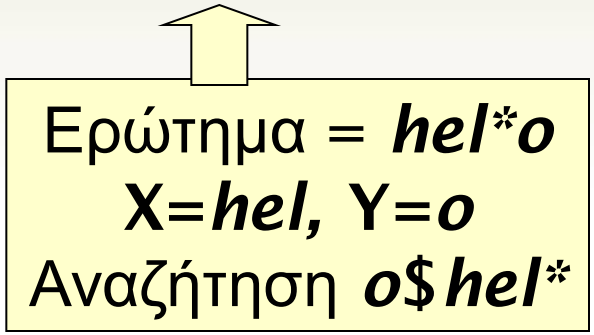
Τα B-trees χειρίζονται * στο τέλος ενός όρου ερωτήματος

- Πώς μπορούμε να χειριστούμε * στο μέσον ενός όρου ερωτήματος;
  - ***co*tion***
- Θα μπορούσαμε να ψάξουμε για ***co**** AND ****tion*** σε ένα B-tree και να εκτελέσουμε συγχώνευση των δυο συνόλων όρων
  - Είναι ακριβή επιλογή
- Η λύση: μετασχηματισμός των wild-card ερωτημάτων, έτσι ώστε τα * να προκύπτουν στο τέλος
- Έτσι προκύπτει το **Permuterm** ευρετήριο

Permuterm index

- Τον όρο ***hello***, κάντε τον index κάτω από όλα τα παρακάτω:
  - ***hello$, ello$h, llohe, lohel, o$hell***

  όπου $ είναι ένας ειδικός χαρακτήρας
- Ερωτήματα:
  - **X** lookup on **X$** **X*** lookup on **$X***
  - ***X** lookup on **X$*** ***X*** lookup on **X***
  - **X*Y** lookup on **Y$X*** **X*Y*Z**



Επεξεργασία ερωτήματος στον Permuterm

- Περιστροφή το wild-card του ερωτήματος προς τα δεξιά
- Κάντε χρήση ενός B-tree lookup για αναζήτηση, όπως πριν
- *Permuterm problem:* ≈ *quadruples lexicon size*

Εμπειρική παρατήρηση για τα Αγγλικά

Bigram (*k*-gram) ευρετήρια

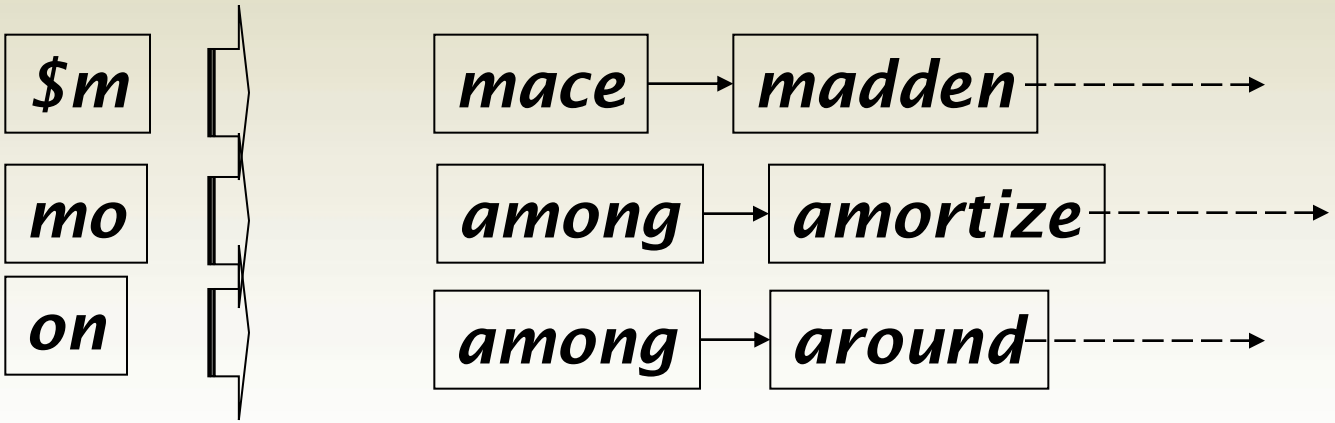
- Απαρίθμηση όλων των *k*-grams (ακολουθία *k* χαρακτήρων) που προκύπτουν σε έναν όρο
- Π.χ., από το κείμενο "***April is the cruelest month***" προκύπτουν τα ακόλουθα 2-grams (*bigrams*)

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

  - $ είναι ένα ειδικό σύμβολο διαχωριστικό λέξεων
- Διατηρούμε έναν *δεύτερο* inverted index *από bigrams προς dictionary terms* που ταιριάζουν κάθε bigram

Παράδειγμα ευρετηρίου Bigram

- Το k-gram ευρετήριο βρίσκει *όρους* βασισμένους στο ερώτημα που αποτελείται από k-grams (εδώ k=2)



Επεξεργασία wild-cards

- Το ερώτημα ***mon**** μπορεί πλέον να εκτελεστεί ως
  - ***$m AND mo AND on***
- Επιστρέφει όρους που ταιριάζουν με την AND εκδοχή του wildcard ερωτήματος
- Αλλά, θα βρίσκαμε και το ***moon***
- Πρέπει να εκτελέσουμε μετα-διήθηση (post-filter) των όρων σε σχέση με το αρχικό διατυπωθέν ερώτημα
- Οι επιβιώσαντες όροι αναζητούνται στον term-document inverted index
- Γρήγορα, αποδοτικά σε χώρο, ως προς τον permuterm

Επεξεργασία wild-card ερωτημάτων

- Πρέπει να εκτελέσουμε ένα Boolean ερώτημα για κάθε enumerated, filtered όρο
- Τα wild-cards μπορεί να έχουν ως αποτέλεσμα ακριβή εκτέλεση ερωτήματος
  - pyth* AND prog*
- Εάν επιτραπεί “laziness”, σίγουρα θα αξιοποιηθεί!

Type your search terms, use ‘*’ if you need to.
E.g., Alex* will match Alexander.

Search

- Ποιες μηχανές αναζήτησης επιτρέπουν wildcard ερωτήματα;

SOUNDEX

Soundex

- Συλλογή ευριστικών για να επεκτείνουμε ένα ερώτημα σε φωνητικά ισοδύναμα
  - Ειδικά σε κάθε γλώσσα – κυρίως για ονόματα
  - Π.χ., ***chebyshev*** → ***tchebycheff***
- Επινοήθηκε για το U.S. census … το 1918

Soundex – τυπικός αλγόριθμος

- Μετατροπή κάθε token που θα μπεί στον index σε μια "ελαττωμένη" μορφή των 4 χαρακτήρων
- Κάνουμε το ίδιο και για τους όρους του ερωτήματος
- Χτίζουμε και ψάχνουμε σε έναν index με τους "ελαττωμένους" όρους
  - (όταν το ερώτημα σχετίζεται με "ακουστικό" ταίριασμα)

- http://www.creativyst.com/Doc/Articles/SoundEx1/SoundEx1.htm#Top

Soundex – τυπικός αλγόριθμος

1. Διατηρούμε το πρώτο γράμμα κάθε λέξης
2. Αλλάζουμε όλες τις εμφανίσεις των ακόλουθων γραμμάτων σε '0' (μηδέν):
   'A', E', 'I', 'O', 'U', 'H', 'W', 'Y'.
3. Αλλάζουμε γράμματα σε ψηφία ως εξής:
   - B, F, P, V $\rightarrow$ 1
   - C, G, J, K, Q, S, X, Z $\rightarrow$ 2
   - D,T $\rightarrow$ 3
   - L $\rightarrow$ 4
   - M, N $\rightarrow$ 5
   - R $\rightarrow$ 6

Soundex (συνέχεια)

4. Διαγράφουμε όλα τα ζεύγη συνεχόμενων ψηφίων
5. Διαγράφουμε όλα τα μηδενικά από το προκύπτον string
6. Γεμίζουμε το προκύπτον string με trailing μηδενικά και επιστρέφουμε τις 4 πρώτες θέσεις, οι οποίες θα είναι της μορφής <κεφαλαίο γράμμα> <ψηφίο> <ψηφίο> <ψηφίο>

Π.χ., ***Herman*** γίνεται H655

Soundex

- Ο Soundex είναι ένας κλασικός αλγόριθμος, παρέχεται από πολλά DBMSs (Oracle, Microsoft, …)
- Πόσο χρήσιμος είναι ο soundex;
- Όχι ιδιαίτερα – για το IR
- OK για “high recall” δουλειές (π.χ., Interpol), παρόλο που είναι πολωμένος προς ονόματα κάποιων εθνικοτήτων
- Οι Zobel and Dart (1996) έδειξαν ότι κάποιοι άλλοι αλγόριθμοι για φωνητικό ταίριασμα αποδίδουν καλύτερα στα πλαίσια της IR