

Introduction to Information Retrieval

ΠΛΕ70: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Διάλεξη 3: Δομές για Λεξικά. Ανάκτηση Ανεκτική στα Σφάλματα.

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

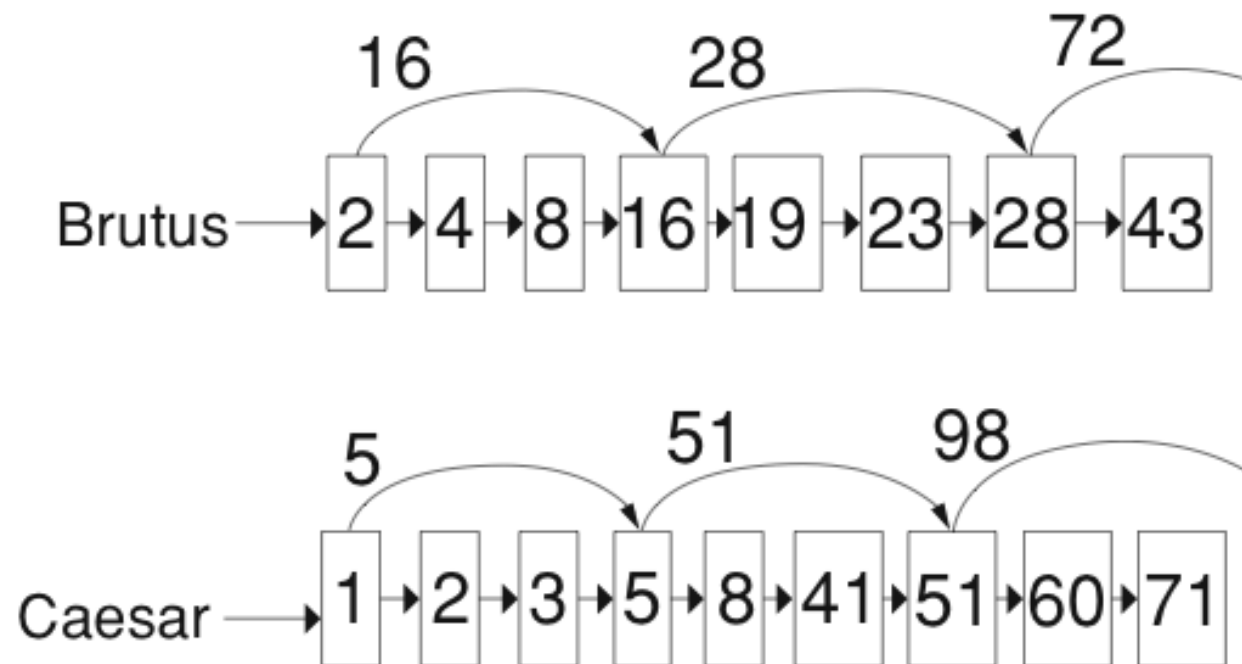
- Ποια είναι τα διαχωριστικά (κενό, απόστροφος, ενωτικά (hyphen))

1. Προσδιορισμός Λεξιλογίου όρων

Από τύπους ισοδύναμων tokens σε 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: “*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>; . . . >

Τι θα δούμε σήμερα;

- Δομές δεδομένων για Λεξικά
- Ανάκτηση Ανεκτική (Tolerant) σε Σφάλματα
 - Ερωτήματα με wild-card («χαρακτήρων μπαλαντέρ»)*
 - Ορθογραφικά λάθη
 - Απόσταση μεταξύ όρων
 - Φωνητική διόρθωση

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;

BRUTUS	→	1	2	4	11	31	45	173	174	
CAESAR	→	1	2	4	5	6	16	57	132	...
CALPURNIA	→	2	31	54	101					

⋮

Λεξικό
υπλοισιπαιγ

postings

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

Κριτήρια για την επιλογή δομής:

- Αποδοτική αναζήτηση ενός όρου (κλειδιού) στο λεξικό.
- Σχετικές συχνότητες 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

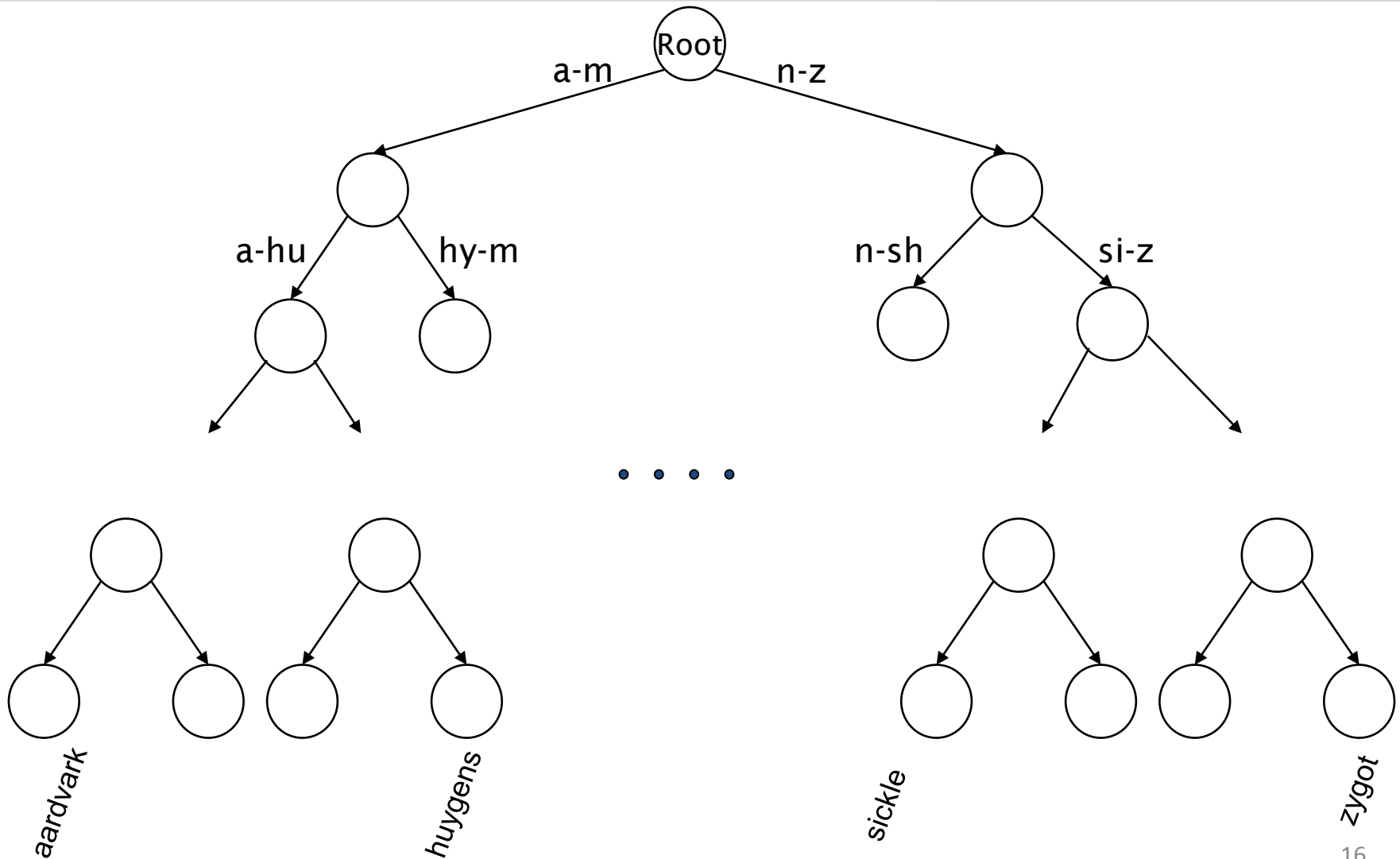
+:

- 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

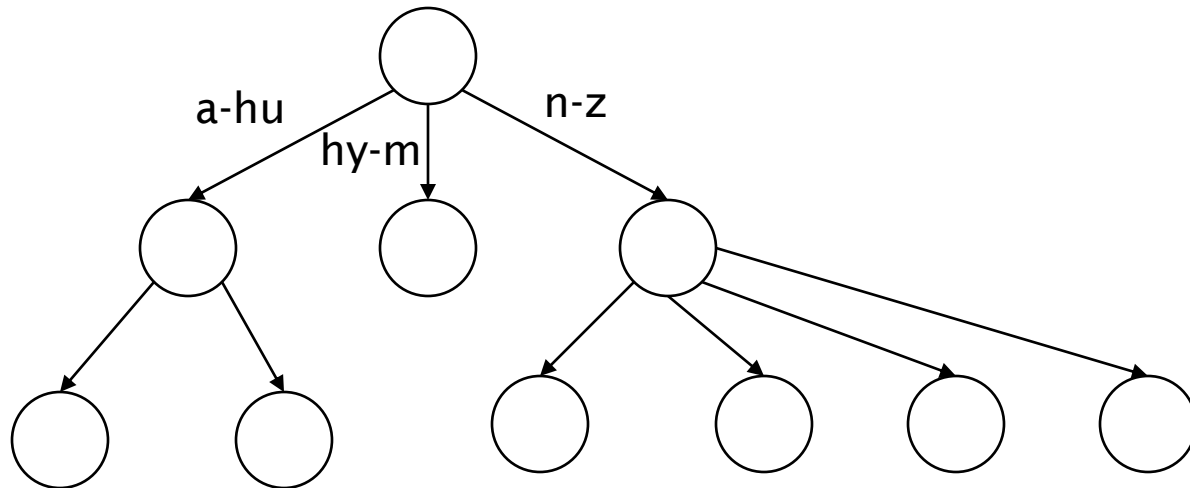
Δέντρα αναζήτησης: Δυαδικό δέντρο



Δέντρα αναζήτησης: Δυαδικό δέντρο

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

Δέντρα: Β-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 ισοζύγηση (rebalancing) των δυαδικών δέντρων είναι ακριβή
 - Αλλά τα B-δέντρα καλύτερα

Λεξικά (σύνοψη)

Λεξιλόγιο (vocabulary): το σύνολο των όρων

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

- Δυο βασικές επιλογές:
 - Πίνακες Κατακερματισμού (Hashtables)
 - Δέντρα (Trees)
- Μερικά Συστήματα Ανάκτησης Πληροφορίας χρησιμοποιούν πίνακες κατακερματισμού άλλα δέντρα

ΕΡΩΤΗΜΑΤΑ ΜΕ *

Ερωτήματα με Wild-card (*)

- (1) Δεν είμαστε σίγουροι για την ορθογραφία της λέξης (πχ Sydney? Sidney?) ή
 - (2) Υπάρχουν πολλαπλές εκδοχές της ορθογραφίας της λέξης και θέλουμε να ανακτήσουμε τα έγγραφα που περιέχουν οποιαδήποτε από αυτές (πχ color, colour)
- Δεν είμαστε σίγουροι αν έχει γίνει stemming
 - Ορθογραφία ξένης λέξης (Σ*ξπ*ρ)

Ερωτήματα με Wild-card (*)

Trailing wild card query

- Π.χ., ***mon****: Βρες όλα τα έγγραφα που περιέχουν οποιαδήποτε λέξη αρχίζει με “*mon*”
 - 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

Ερωτήματα με Wild-card (*)

Leading wild card queries

- **Π.χ., *mon:** Βρες όλα τα έγγραφα που περιέχουν οποιαδήποτε λέξη τελειώνει σε “mon”
- **πιο δύσκολο**
 - Διατήρησε ένα επιπρόσθετο B-tree για τους όρους ανάποδα (*backwards*), πχ ο όρος *demon* -> *nomed*
 - *Reverse B-tree*
 - Ανάκτησε όλους τους όρους t στο διάστημα: **$nom \leq t < non$** .

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**?)

Γενικά ερωτήματα με *

- * στη μέση του όρου
 - *co*tion*
- Αναζήτησε το *co** AND **tion* σε ένα B-tree και ένα reverse και υπολόγισε την τομή των συνόλων
 - Ακριβό!
- Δύο λύσεις – και οι δύο:
 - I. Έχουν ένα *ειδικό ευρετήριο*
 - II. Μετατρέπουν την αρχική ερώτησης q^* σε μια Boolean ερώτηση Q στο ειδικό ευρετήριο έτσι ώστε η απάντηση στο Q να είναι *υπερσύνολο* της απάντησης στο q^*
 - III. Στη συνέχεια ελέγχουν τις απαντήσεις

Γενικά ερωτήματα με *

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

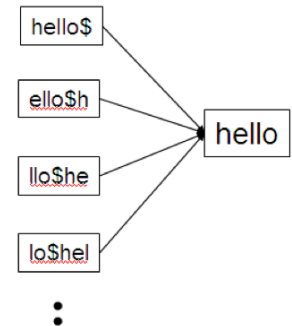
Permuterm Index (ευρετήριο αντιμετατεθειμένων όρων)

Ευρετήριο αντιμετατεθειμένων όρων

Κατασκευάζουμε ένα **ευρετήριο αντιμετατεθειμένων όρων (permuter index)** στο οποίο όλες οι παραλλαγές που προκύπτουν από την (δεξιόστροφη) περιστροφή (rotation) του όρου συνδέονται με τον αρχικό όρο

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

- **hello\$, \$hello, o\$hell , lo\$hel , llo\$he, ello\$h**
 - όπου \$ ένα ειδικός χαρακτήρας που σηματοδοτεί το τέλος μιας λέξης
 - Permuter vocabulary



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

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

Ψάχνουμε το $lo\$hel^*$

Ευρετήριο αντιμετατεθειμένων όρων

Παράδειγμα

Ευρετήριο αντιμετατεθειμένων όρων για moron, man

Εισάγουμε στο λεξικό όλες τις περιστροφές των όρων ώστε να δείχνουν στον αρχικό όρο

moron -> moron\$ -> στο ευρετήριο: **moron\$, \$moron, n\$mor, on\$mor, ron\$mor, oron\$m**

man -> man\$ -> στο ευρετήριο: **man\$, \$man, n\$ma, an\$m**

Ερώτημα

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

Ερώτημα: $mo*n$ -> $n\$mo*$

Match?

Ερώτημα: $m*$ -> $\$m*$

Match?

Ευρετήριο αντιμεταθετιμένων όρων

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

Εμπειρική παρατήρηση για τα Αγγλικά

Ευρετήρια k -γραμμμάτων (k -gram indexes)

k -gram: ακολουθία k χαρακτήρων

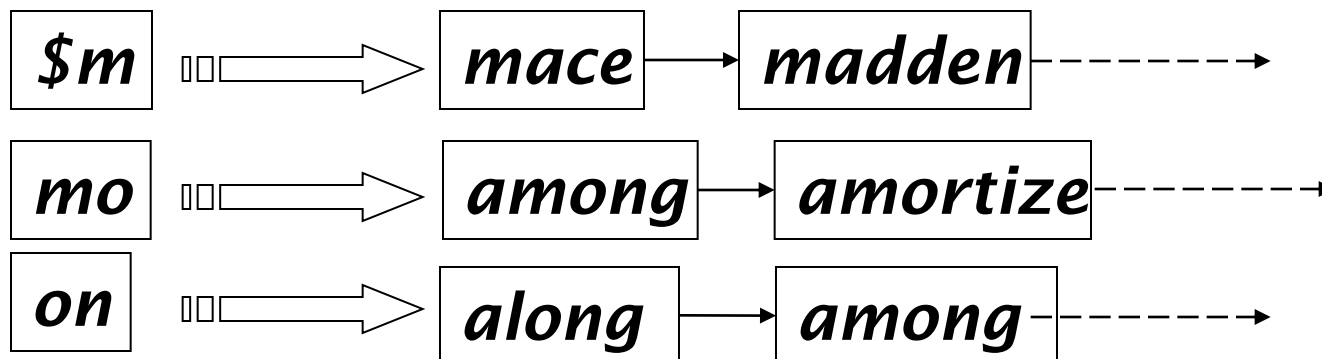
- Απαρίθμησε όλα τα k -γράμματα που εμφανίζονται σε κάθε όρο
 - π.χ., για το κείμενο “***April is the cruelest month***” έχουμε τα 2-γράμματα (*bigrams*)

\$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
- Σε ένα k -gram ευρετήριο για το λεξικό που περιέχει όλα τα k -grams που εμφανίζονται σε οποιονδήποτε όρο
- Διατήρησε ένα δεύτερο αντεστραμμένο ευρετήριο από τα 2-γράμματα στους όρους του λεξικού που τα περιέχουν

Παράδειγμα ευρετηρίου k -γραμμάτων


- 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!

Search

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

- Ποιες μηχανές αναζήτησης επιτρέπουν τέτοια ερωτήματα;

ΔΙΟΡΘΩΣΗ ΟΡΘΟΓΡΑΦΙΚΩΝ ΛΑΘΩΝ

Διόρθωση ορθογραφικών λαθών

- Δύο βασικές χρήσεις
 - Διόρθωση των *εγγράφων* που ευρετηριοποιούνται
 - Διόρθωση των *ερωτημάτων* ώστε να ανακτηθούν «σωστές» απαντήσεις
- Δυο βασικές κατηγορίες:
 - *Μεμονωμένες λέξεις (isolated term)*
 - Εξέτασε κάθε λέξη μόνη της για λάθη
 - Δεν πιάνει typos που έχουν ως αποτέλεσμα σωστά γραμμένες λέξεις
 - π.χ., *from* → *form*
 - Βασισμένη σε *συμφραζόμενα* (context sensitive)
 - Κοιτά τις λέξεις γύρω,
 - π.χ., *I flew form Heathrow to Narita.*

Διόρθωση εγγράφων

- Χρήσιμη ιδιαίτερα για έγγραφα μετά από OCR
 - Αλγόριθμοι διόρθωσης ρυθμισμένοι για αυτό: rn μοιάζει με m
 - Μπορεί να χρησιμοποιούν ειδική γνώση (domain-specific)
 - Π.χ., OCR μπερδεύει το O με το D πιο συχνά από το O και το I (που είναι γειτονικά στα QWERTY πληκτρολόγιο), οπότε πιο πιθανή η ανταλλαγή τους στην πληκτρολόγηση
- Αλλά συχνά: web σελίδες αλλά και τυπωμένο υλικό έχουν typos
- Στόχος: το λεξικό να περιέχει λιγότερα ορθογραφικά λάθη
 - Αλλά συχνά δεν αλλάζουμε τα έγγραφα αλλά επεκτείνουμε την απεικόνιση ερωτήματος – εγγράφου

Διόρθωση μεμονωμένης λέξης

- Θεμελιώδης υπόθεση – *υπάρχει ένα λεξικό που μας δίνει τη σωστή ορθογραφία*
- Δυο βασικές επιλογές για αυτό το λεξικό
 - Ένα standard λεξικό όπως
 - Webster's English Dictionary
 - Ένα "industry-specific" λεξικό – hand-maintained
 - Το λεξικό της συλλογής (corpus)
 - Π.χ., όλες οι λέξεις στο web
 - Όλα τα ονόματα, ακρώνυμα κλπ.
 - (συμπεριλαμβανομένων και των ορθογραφικών λαθών)

Γενικά θέματα

- (1) Στο ερώτημα *carot* πάντα επέστρεψε τα έγγραφα που περιέχουν το *carot* καθώς και τα έγγραφα όλες τις διορθωμένες εκδοχές του όρου, πχ *carrot* and *tarot*.
- (2) Όπως στο (1) , αλλά διορθώσεις μόνο αν το *carot* δεν είναι στο λεξικό
- (3) Όπως στο (1), αλλά μόνο αν η αρχική ερώτηση επιστρέφει λίγα (πχ λιγότερο από 5) έγγραφα.
- (4) Όταν η αρχική ερώτηση επιστρέφει λιγότερα από έναν προκαθορισμένο αριθμό από έγγραφα επιστρέφει «*spelling suggestions*” : “Did you mean *carrot*?” (και όχι επιπρόσθετα έγγραφα)

Γενικά θέματα

- (1) Επιστρέφουμε τη λέξη (λέξεις) που είναι πιο «κοντά»
- (2) Όταν ισοπαλία
 - (1) Την πιο συχνή (συχνές) στη συλλογή
 - (2) Την πιο συχνή στα ερωτήματα

Δείτε στο

<http://www.google.com/jobs/archive/britney.html>

στατιστικά για misspellings του Britney Spears

Διόρθωση μεμονωμένης λέξης

Δοθέντος ενός **λεξικού** και μιας ακολουθίας χαρακτήρων Q , επέστρεψε τις λέξεις του λεξικού που είναι **πιο κοντά** στο Q

- Τι σημαίνει “πιο κοντά”?
- Θα εξετάσουμε δύο ορισμούς εγγύτητας:
 - Την **απόσταση διόρθωσης -- edit distance** (Levenshtein distance) και την **σταθμισμένη εκδοχή της -- weighted edit distance**
 - **Επικάλυψη (overlap) n-γραμμάτων**

Απόσταση διόρθωσης (Edit distance)

ΟΡΙΣΜΟΣ: Δοθέντων δυο αλφαριθμητικών (strings) S_1 and S_2 , ο ελάχιστος αριθμός πράξεων για τη μετατροπή του ενός στο άλλο

- Συνήθως, οι πράξεις είναι σε επίπεδο χαρακτήρα
 - **Levenshtein distance:** (1) Insert – Εισαγωγή, (2) Delete - Διαγραφή και (3) Replace – Αντικατάσταση ενός χαρακτήρα
 - **Damerau-Levenshtein distance:** + **Transposition** - Αντιμετάθεση ενός χαρακτήρα
- Π.χ., η απόσταση διόρθωσης από **do***f***** σε **do***g***** είναι 1
 - Από **cat** σε **act** είναι 2 (Μόνο 1 με αντιμετάθεση)
 - Από **cat** σε **dog** είναι 3.

Απόσταση Διόρθωσης (Edit distance)

Παράδειγμα

Levenshtein distance: *dog-do*: 1, *cat-cart*: 1, *cat-cut*: 1, *cat-act*: 2

Damerau-Levenshtein distance: cat-act: 1

- Γενικά υπολογίζεται με Δυναμικό Προγραμματισμό.
- Κοιτάξτε το <http://www.merriampark.com/ld.htm> για ένα παράδειγμα και ένα applet.

Δυναμικός προγραμματισμός

- ✓ Εκφράζουμε το πρόβλημα ως συνδυασμό υπό-προβλημάτων – η βέλτιστη λύση βασίζεται στη βέλτιστη λύση του υπό-πρόβληματος
 - ✓ Στην περίπτωση των αποστάσεων διόρθωσης – το υπό-πρόβλημα δυο προθεμάτων:

Ο βέλτιστος τρόπος από μια λέξη σε μια άλλη, βασίζεται στο βέλτιστο τρόπο από κάποιο πρόθεμα της πρώτης σε πρόθεμα της δεύτερης
- ✓ Έναν Πίνακα
 - ✓ Γραμμές: Γράμματα (προθέματα) της πρώτη λέξης
 - ✓ Στήλες: Γράμματα (προθέματα) της δεύτερης λέξης
 - ✓ Θέσεις του πίνακα: βέλτιστο κόστος (απόσταση)

Υπολογισμός απόστασης διόρθωσης

String s_2

		f	a	s	t	
	0	1	2	3	4	
String s_1	c	1	1	2	3	4
	a	2	2	1	2	3
	t	3	3	2	2	2
	s	4	4	3	2	3

cats – fast

Κάθε στοιχείο $m[i, j]$ του πίνακα μας δίνει το βέλτιστο κόστος (απόσταση) για να πάμε από το πρόθεμα μήκους i του s_1 στο πρόθεμα μήκους j του s_2

Δυναμικός προγραμματισμός

- ✓ Πως υπολογίζουμε τα στοιχεία του πίνακα;
- ✓ **Επικαλυπτόμενες υπό-λύσεις:**
 - ✓ Βέλτιστο κόστος $m[i, j]$
 - ✓ Πχ $m[2, 3]$ ca \rightarrow fas
 - ✓ 3 διαφορετικοί τρόποι
 - ✓ $m[i, j-1]$ από αριστερά (γραμμή)
 - ✓ $m[i-1, j]$ από πάνω (στήλη)
 - ✓ $m[i-1, j-1]$ (διαγώνια)

$i-1, j-1$	$i-1, j$
$i, j-1$	i, j

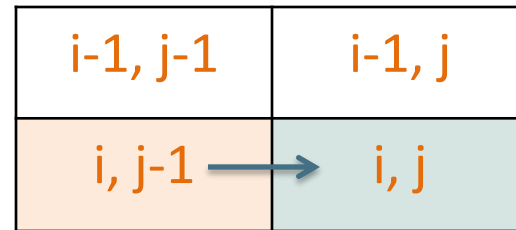
Αλγόριθμος (από αριστερά)

LEVENSHTEINDISTANCE(s_1, s_2)

```

1  for  $i \leftarrow 0$  to  $|s_1|$ 
2  do  $m[i, 0] = i$ 
3  for  $j \leftarrow 0$  to  $|s_2|$ 
4  do  $m[0, j] = j$ 
5  for  $i \leftarrow 1$  to  $|s_1|$ 
6  do for  $j \leftarrow 1$  to  $|s_2|$ 
7     do if  $s_1[i] = s_2[j]$ 
8         then  $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]\}$ 
9         else  $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1\}$ 
10 return  $m[|s_1|, |s_2|]$ 

```



Operations: **insert (cost 1)**, delete (cost 1), replace (cost 1), copy (cost 0)

Αλγόριθμος (από πάνω)

LEVENSHTEINDISTANCE(s_1, s_2)

```

1  for  $i \leftarrow 0$  to  $|s_1|$ 
2  do  $m[i, 0] = i$ 
3  for  $j \leftarrow 0$  to  $|s_2|$ 
4  do  $m[0, j] = j$ 
5  for  $i \leftarrow 1$  to  $|s_1|$ 
6  do for  $j \leftarrow 1$  to  $|s_2|$ 
7     do if  $s_1[i] = s_2[j]$ 
8         then  $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]\}$ 
9         else  $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1\}$ 
10 return  $m[|s_1|, |s_2|]$ 

```

$i-1, j-1$	$i-1, j$
$i, j-1$	i, j

Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)

Υπολογισμός απόστασης διόρθωσης

LEVENSHTEINDISTANCE(s_1, s_2)

```

1  for  $i \leftarrow 0$  to  $|s_1|$ 
2  do  $m[i, 0] = i$ 
3  for  $j \leftarrow 0$  to  $|s_2|$ 
4  do  $m[0, j] = j$ 
5  for  $i \leftarrow 1$  to  $|s_1|$ 
6  do for  $j \leftarrow 1$  to  $|s_2|$ 
7     do if  $s_1[i] = s_2[j]$ 
8         then  $m[i, j] = \min$ 
9             else  $m[i, j] = \min$ 
10 return  $m[|s_1|, |s_2|]$ 
Operations: insert (cost 1), delet
(cost 0)

```

Αρχικοποίηση

String s_2

		f	a	s	t
	0	1	2	3	4
c	1	1	2	3	4
a	2	2	1	2	3
t	3	3	2	2	2
s	4	4	3	2	3

Κόστος διόρθωσης για τα
προθέματα

Υπολογισμός απόστασης Levenshtein

Για να υπολογίσουμε το $m[i, j]$

	$j - 1$	j
$i - 1$	Εξαρτάται από το επόμενο γράμμα	Κόστος από τον πάνω γείτονα (delete) $[i - 1, j]$
i	κόστος από τον αριστερό γείτονα (insert) $[i, j - 1]$	Το μικρότερο από τις 3 πιθανές για να φτάσουμε στο $[i, j]$

Αλγόριθμος (διαγώνια)

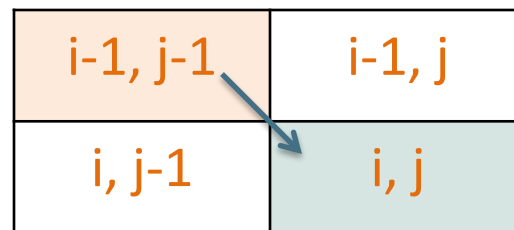
LEVENSHTEINDISTANCE(s_1, s_2)

Αν το i -οστό στοιχείο του s_1 είναι ίδιο με το j -οστό στοιχείο του s_2

```

1  for  $i \leftarrow 0$  to  $|s_1|$ 
2  do  $m[i, 0] = i$ 
3  for  $j \leftarrow 0$  to  $|s_2|$ 
4  do  $m[0, j] = j$ 
5  for  $i \leftarrow 1$  to  $|s_1|$ 
6  do for  $j \leftarrow 1$  to  $|s_2|$ 
7     do if  $s_1[i] = s_2[j]$ 
8         then  $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]\}$ 
9         else  $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1\}$ 
10 return  $m[|s_1|, |s_2|]$ 

```

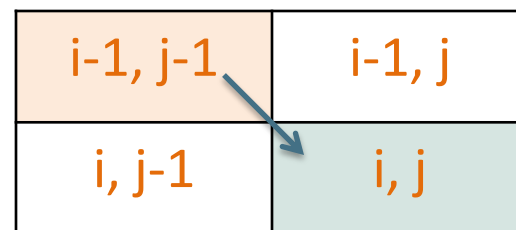


Operations: insert (cost 1), delete (cost 1), replace (cost 1), **copy** (cost 0)

Αλγόριθμος (διαγώνια)

LEVENSHTEINDISTANCE(s_1, s_2)

Αν το i -οστό στοιχείο του s_1 είναι διαφορετικό από το j -οστό στοιχείο του s_2



1 **for** $i \leftarrow 0$ **to** $|s_1|$

2 **do** $m[i, 0] = i$

3 **for** $j \leftarrow 0$ **to** $|s_2|$

4 **do** $m[0, j] = j$

5 **for** $i \leftarrow 1$ **to** $|s_1|$

6 **do for** $j \leftarrow 1$ **to** $|s_2|$

7 **do if** $s_1[i] = s_2[j]$

8 **then** $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]\}$

9 **else** $m[i, j] = \min\{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1\}$

10 **return** $m[|s_1|, |s_2|]$

Operations: insert (cost 1), delete (cost 1), **replace (cost 1)**, copy (cost 0)

Υπολογισμός απόστασης Levenshtein

Κόστος από τον πάνω αριστερό γείτονα Copy ή Replace	Κόστος από τον πάνω γείτονα Delete
Κόστος από τον αριστερό γείτονα Insert	Το μικρότερο από τα 3 κόστη

Υπολογισμός απόστασης Levenshtein: παράδειγμα

		f	a	s	t
	<u>0</u>	<u>1</u> <u>1</u>	<u>2</u> <u>2</u>	<u>3</u> <u>3</u>	<u>4</u> <u>4</u>
c	<u>1</u> <u>1</u>	<u>1</u> <u>2</u> <u>2</u> <u>1</u>	<u>2</u> <u>3</u> <u>2</u> <u>2</u>	<u>3</u> <u>4</u> <u>3</u> <u>3</u>	<u>4</u> <u>5</u> <u>4</u> <u>4</u>
a	<u>2</u> <u>2</u>	<u>2</u> <u>2</u> <u>3</u> <u>2</u>	<u>1</u> <u>3</u> <u>3</u> <u>1</u>	<u>3</u> <u>4</u> <u>2</u> <u>2</u>	<u>4</u> <u>5</u> <u>3</u> <u>3</u>
t	<u>3</u> <u>3</u>	<u>3</u> <u>3</u> <u>4</u> <u>3</u>	<u>3</u> <u>2</u> <u>4</u> <u>2</u>	<u>2</u> <u>3</u> <u>3</u> <u>2</u>	<u>2</u> <u>4</u> <u>3</u> <u>2</u>
s	<u>4</u> <u>4</u>	<u>4</u> <u>4</u> <u>5</u> <u>4</u>	<u>4</u> <u>3</u> <u>5</u> <u>3</u>	<u>2</u> <u>3</u> <u>4</u> <u>2</u>	<u>3</u> <u>3</u> <u>3</u> <u>3</u>

Δυναμικός προγραμματισμός

1. Βέλτιστη υπό-δομής (**Optimal substructure**): Η βέλτιστη λύση σε ένα πρόβλημα περιέχει τις υπό-λύσεις, δηλαδή τις βέλτιστες λύσεις σε υπό-προβλήματα
2. Επικαλυπτόμενες υπό-λύσεις (**Overlapping subsolutions**): Οι υπο-λύσεις υπολογίζονται ξανά και ξανά όταν υπολογίζονται οι ολικές βέλτιστες λύσεις στον brute-force αλγόριθμο.

Δυναμικός προγραμματισμός

- ✓ Στην περίπτωση των αποστάσεων διόρθωσης – το υπό-πρόβλημα δυο προθεμάτων
- ✓ Οι επικαλυπτόμενες υπό-λύσεις: χρειαζόμαστε τις περισσότερες αποστάσεις 3 φορές: κίνηση δεξιά, στη διαγώνιο, κάτω

Υπολογισμός απόστασης: παράδειγμα

Από OSLO σε SNOW

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{0}{0}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$				
s	$\frac{2}{2}$				
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{0}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{?}$			
s	$\frac{2}{2}$				
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 1	1 2 2 1			
s	2 2				
l	3 3				
o	4 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> 1	<u>1 2</u> 2 1	<u>2 3</u> 2 ?		
s	<u>2</u> 2				
l	<u>3</u> 3				
o	<u>4</u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{\quad}$ 0	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$		
s	$\frac{2}{2}$				
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{\quad}$ 0	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{2}{3}$ $\frac{4}{?}$	
s	$\frac{2}{2}$				
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	
s	<u> 2 </u> 2				
l	<u> 3 </u> 3				
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{0}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{2}{3}$ $\frac{4}{2}$	$\frac{4}{3}$ $\frac{5}{?}$
s	$\frac{2}{2}$				
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{\quad}$ 0	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{2}{3}$ $\frac{4}{2}$	$\frac{4}{3}$ $\frac{5}{3}$
s	$\frac{2}{2}$				
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{0}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{3}$ $\frac{2}{2}$	$\frac{2}{4}$ $\frac{3}{2}$	$\frac{4}{5}$ $\frac{3}{3}$
s	$\frac{2}{2}$	$\frac{1}{3}$ $\frac{2}{?}$			
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1			
l	<u> 3 </u> 3				
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 ?		
l	<u> 3 </u> 3				
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> 1	<u>1 2</u> 2 1	<u>2 3</u> 2 2	<u>2 4</u> 3 2	<u>4 5</u> 3 3
s	<u>2</u> 2	<u>1 2</u> 3 1	<u>2 3</u> 2 2		
l	<u>3</u> 3				
o	<u>4</u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> 1	<u>1 2</u> 2 1	<u>2 3</u> 2 2	<u>2 4</u> 3 2	<u>4 5</u> 3 3
s	<u>2</u> 2	<u>1 2</u> 3 1	<u>2 3</u> 2 2	<u>3 3</u> 3 ?	
l	<u>3</u> 3				
o	<u>4</u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	
l	<u> 3 </u> 3				
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{0}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{3}$ $\frac{2}{2}$	$\frac{2}{4}$ $\frac{3}{2}$	$\frac{4}{5}$ $\frac{3}{3}$
s	$\frac{2}{2}$	$\frac{1}{3}$ $\frac{2}{1}$	$\frac{2}{3}$ $\frac{2}{2}$	$\frac{3}{3}$ $\frac{3}{3}$	$\frac{3}{4}$ $\frac{4}{?}$
l	$\frac{3}{3}$				
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3				
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{\quad}$ 0	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{2}{3}$ $\frac{4}{2}$	$\frac{4}{3}$ $\frac{5}{3}$
s	$\frac{2}{2}$	$\frac{1}{3}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{3}{3}$ $\frac{3}{3}$	$\frac{3}{4}$ $\frac{4}{3}$
l	$\frac{3}{3}$	$\frac{3}{4}$ $\frac{2}{?}$			
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>			
o	<u>4</u> <u>4</u>				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	$\frac{\quad}{\quad}$ 0	$\frac{1}{1}$ $\frac{1}{1}$	$\frac{2}{2}$ $\frac{2}{2}$	$\frac{3}{3}$ $\frac{3}{3}$	$\frac{4}{4}$ $\frac{4}{4}$
o	$\frac{1}{1}$	$\frac{1}{2}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{2}{3}$ $\frac{4}{2}$	$\frac{4}{3}$ $\frac{5}{3}$
s	$\frac{2}{2}$	$\frac{1}{3}$ $\frac{2}{1}$	$\frac{2}{2}$ $\frac{3}{2}$	$\frac{3}{3}$ $\frac{3}{3}$	$\frac{3}{4}$ $\frac{4}{3}$
l	$\frac{3}{3}$	$\frac{3}{4}$ $\frac{2}{2}$	$\frac{2}{3}$ $\frac{3}{?}$		
o	$\frac{4}{4}$				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3	<u> 3 2 </u> 4 2	<u> 2 3 </u> 3 2		
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 ?</u>	
o	<u>4</u> <u>4</u>				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3	<u> 3 2 </u> 4 2	<u> 2 3 </u> 3 2	<u> 3 4 </u> 3 3	
o	<u> 4 </u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> 1	<u>1 2</u> 2 1	<u>2 3</u> 2 2	<u>2 4</u> 3 2	<u>4 5</u> 3 3
s	<u>2</u> 2	<u>1 2</u> 3 1	<u>2 3</u> 2 2	<u>3 3</u> 3 3	<u>3 4</u> 4 3
l	<u>3</u> 3	<u>3 2</u> 4 2	<u>2 3</u> 3 2	<u>3 4</u> 3 3	<u>4 4</u> 4 ?
o	<u>4</u> 4				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4</u> <u>4</u>				

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1 1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2 2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3 3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4 4</u>	<u>4 3</u> <u>5 ?</u>			

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4</u> <u>4</u>	<u>4 3</u> <u>5 3</u>			

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4</u> <u>4</u>	<u>4 3</u> <u>5 3</u>	<u>3 3</u> <u>4 ?</u>		

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3	<u> 3 2 </u> 4 2	<u> 2 3 </u> 3 2	<u> 3 4 </u> 3 3	<u> 4 4 </u> 4 4
o	<u> 4 </u> 4	<u> 4 3 </u> 5 3	<u> 3 3 </u> 4 3		

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 1	1 2 2 1	2 3 2 2	2 4 3 2	4 5 3 3
s	2 2	1 2 3 1	2 3 2 2	3 3 3 3	3 4 4 3
l	3 3	3 2 4 2	2 3 3 2	3 4 3 3	4 4 4 4
o	4 4	4 3 5 3	3 3 4 3	2 4 4 ?	

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3	<u> 3 2 </u> 4 2	<u> 2 3 </u> 3 2	<u> 3 4 </u> 3 3	<u> 4 4 </u> 4 4
o	<u> 4 </u> 4	<u> 4 3 </u> 5 3	<u> 3 3 </u> 4 3	<u> 2 4 </u> 4 2	

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3	<u> 3 2 </u> 4 2	<u> 2 3 </u> 3 2	<u> 3 4 </u> 3 3	<u> 4 4 </u> 4 4
o	<u> 4 </u> 4	<u> 4 3 </u> 5 3	<u> 3 3 </u> 4 3	<u> 2 4 </u> 4 2	<u> 4 5 </u> 3 ?

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4</u> <u>4</u>	<u>4 3</u> <u>5 3</u>	<u>3 3</u> <u>4 3</u>	<u>2 4</u> <u>4 2</u>	<u>4 5</u> <u>3 3</u>

Υπολογισμός απόστασης: παράδειγμα

		s	n	o	w
	<u> </u> 0	<u> 1 </u> 1	<u> 2 </u> 2	<u> 3 </u> 3	<u> 4 </u> 4
o	<u> 1 </u> 1	<u> 1 2 </u> 2 1	<u> 2 3 </u> 2 2	<u> 2 4 </u> 3 2	<u> 4 5 </u> 3 3
s	<u> 2 </u> 2	<u> 1 2 </u> 3 1	<u> 2 3 </u> 2 2	<u> 3 3 </u> 3 3	<u> 3 4 </u> 4 3
l	<u> 3 </u> 3	<u> 3 2 </u> 4 2	<u> 2 3 </u> 3 2	<u> 3 4 </u> 3 3	<u> 4 4 </u> 4 4
o	<u> 4 </u> 4	<u> 4 3 </u> 5 3	<u> 3 3 </u> 4 3	<u> 2 4 </u> 4 2	<u> 4 5 </u> 3 3

Πως μπορώ να δω τις πράξεις που οδήγησαν από OSLO σε SNOW?

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1</u> <u>1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2</u> <u>2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3</u> <u>3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4</u> <u>4</u>	<u>4 3</u> <u>5 3</u>	<u>3 3</u> <u>4 3</u>	<u>2 4</u> <u>4 2</u>	<u>4 5</u> <u>3 3</u>

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1 1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2 2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3 3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4 4</u>	<u>4 3</u> <u>5 3</u>	<u>3 3</u> <u>4 3</u>	<u>2 4</u> <u>4 2</u>	<u>4 5</u> <u>3 3</u>

cost	operation	input	output
1	insert	*	w

		s	n	o	w
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>
o	<u>1 1</u>	<u>1 2</u> <u>2 1</u>	<u>2 3</u> <u>2 2</u>	<u>2 4</u> <u>3 2</u>	<u>4 5</u> <u>3 3</u>
s	<u>2 2</u>	<u>1 2</u> <u>3 1</u>	<u>2 3</u> <u>2 2</u>	<u>3 3</u> <u>3 3</u>	<u>3 4</u> <u>4 3</u>
l	<u>3 3</u>	<u>3 2</u> <u>4 2</u>	<u>2 3</u> <u>3 2</u>	<u>3 4</u> <u>3 3</u>	<u>4 4</u> <u>4 4</u>
o	<u>4 4</u>	<u>4 3</u> <u>5 3</u>	<u>3 3</u> <u>4 3</u>	<u>2 4</u> <u>4 2</u>	<u>4 5</u> <u>3 3</u>

cost	operation	input	output
0	(copy)	o	o
1	insert	*	w

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 1	1 2 2 1	2 3 2 2	2 4 3 2	4 5 3 3
s	2 2	1 2 3 1	2 3 2 2	3 3 3 3	3 4 4 3
l	3 3	3 2 4 2	2 3 3 2	3 4 3 3	4 4 4 4
o	4 4	4 3 5 3	3 3 4 3	2 4 4 2	4 5 3 3

cost	operation	input	output
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 1	1 2 2 1	2 3 2 2	2 4 3 2	4 5 3 3
s	2 2	1 2 3 1	2 3 2 2	3 3 3 3	3 4 4 3
l	3 3	3 2 4 2	2 3 3 2	3 4 3 3	4 4 4 4
o	4 4	4 3 5 3	3 3 4 3	2 4 4 2	4 5 3 3

cost	operation	input	output
0	(copy)	s	s
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 1	1 2 2 1	2 3 2 2	2 4 3 2	4 5 3 3
s	2 2	1 2 3 1	2 3 2 2	3 3 3 3	3 4 4 3
l	3 3	3 2 4 2	2 3 3 2	3 4 3 3	4 4 4 4
o	4 4	4 3 5 3	3 3 4 3	2 4 4 2	4 5 3 3

cost	operation	input	output
1	delete	o	*
0	(copy)	s	s
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

Πως μπορώ να δω τις πράξεις που οδήγησαν από CAT σε CATCAT?

		c	a	t	c	a	t	
	$\frac{\quad}{\quad}$	$\frac{0}{\quad}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{3}{3}$	$\frac{4}{4}$	$\frac{5}{5}$	$\frac{6}{6}$
c	$\frac{1}{1}$	$\frac{0}{2}$ $\frac{2}{0}$	$\frac{2}{1}$ $\frac{3}{1}$	$\frac{3}{2}$ $\frac{4}{2}$	$\frac{3}{3}$ $\frac{5}{3}$	$\frac{5}{4}$ $\frac{6}{4}$	$\frac{6}{5}$ $\frac{7}{5}$	
a	$\frac{2}{2}$	$\frac{2}{3}$ $\frac{1}{1}$	$\frac{0}{2}$ $\frac{2}{0}$	$\frac{2}{1}$ $\frac{3}{1}$	$\frac{3}{2}$ $\frac{4}{2}$	$\frac{3}{3}$ $\frac{5}{3}$	$\frac{5}{4}$ $\frac{6}{4}$	
t	$\frac{3}{3}$	$\frac{3}{4}$ $\frac{2}{2}$	$\frac{2}{3}$ $\frac{1}{1}$	$\frac{0}{2}$ $\frac{2}{0}$	$\frac{2}{1}$ $\frac{3}{1}$	$\frac{3}{2}$ $\frac{4}{2}$	$\frac{3}{3}$ $\frac{5}{3}$	

		c	a	t	c	a	t
	<u>0</u>	<u>1 1</u>	<u>2 2</u>	<u>3 3</u>	<u>4 4</u>	<u>5 5</u>	<u>6 6</u>
c	<u>1 1</u>	<u>0 2</u> <u>2 0</u>	<u>2 3</u> <u>1 1</u>	<u>3 4</u> <u>2 2</u>	<u>3 5</u> <u>3 3</u>	<u>5 6</u> <u>4 4</u>	<u>6 7</u> <u>5 5</u>
a	<u>2 2</u>	<u>2 1</u> <u>3 1</u>	<u>0 2</u> <u>2 0</u>	<u>2 3</u> <u>1 1</u>	<u>3 4</u> <u>2 2</u>	<u>3 5</u> <u>3 3</u>	<u>5 6</u> <u>4 4</u>
t	<u>3 3</u>	<u>3 2</u> <u>4 2</u>	<u>2 1</u> <u>3 1</u>	<u>0 2</u> <u>2 0</u>	<u>2 3</u> <u>1 1</u>	<u>3 4</u> <u>2 2</u>	<u>3 5</u> <u>3 3</u>

cost	operation	input	output
1	insert	*	c
1	insert	*	a
1	insert	*	t
0	(copy)	c	c
0	(copy)	a	a
0	(copy)	t	t

		c	a	t	c	a	t
	<u> </u> 0	<u> </u> 1 1	<u> </u> 2 2	<u> </u> 3 3	<u> </u> 4 4	<u> </u> 5 5	<u> </u> 6 6
c	<u> </u> 1 <u> </u> 1	<u> </u> 0 2 <u> </u> 2 0	<u> </u> 2 3 <u> </u> 1 1	<u> </u> 3 4 <u> </u> 2 2	<u> </u> 3 5 <u> </u> 3 3	<u> </u> 5 6 <u> </u> 4 4	<u> </u> 6 7 <u> </u> 5 5
a	<u> </u> 2 <u> </u> 2	<u> </u> 2 1 <u> </u> 3 1	<u> </u> 0 2 <u> </u> 2 0	<u> </u> 2 3 <u> </u> 1 1	<u> </u> 3 4 <u> </u> 2 2	<u> </u> 3 5 <u> </u> 3 3	<u> </u> 5 6 <u> </u> 4 4
t	<u> </u> 3 <u> </u> 3	<u> </u> 3 2 <u> </u> 4 2	<u> </u> 2 1 <u> </u> 3 1	<u> </u> 0 2 <u> </u> 2 0	<u> </u> 2 3 <u> </u> 1 1	<u> </u> 3 4 <u> </u> 2 2	<u> </u> 3 5 <u> </u> 3 3

cost	operation	input	output
0	(copy)	c	c
1	insert	*	a
1	insert	*	t
1	insert	*	c
0	(copy)	a	a
0	(copy)	t	t

		c	a	t	c	a	t
	<u>0</u>	<u>1</u> <u>1</u>	<u>2</u> <u>2</u>	<u>3</u> <u>3</u>	<u>4</u> <u>4</u>	<u>5</u> <u>5</u>	<u>6</u> <u>6</u>
c	<u>1</u> <u>1</u>	<u>0</u> <u>2</u> <u>2</u> <u>0</u>	<u>2</u> <u>3</u> <u>1</u> <u>1</u>	<u>3</u> <u>4</u> <u>2</u> <u>2</u>	<u>3</u> <u>5</u> <u>3</u> <u>3</u>	<u>5</u> <u>6</u> <u>4</u> <u>4</u>	<u>6</u> <u>7</u> <u>5</u> <u>5</u>
a	<u>2</u> <u>2</u>	<u>2</u> <u>1</u> <u>3</u> <u>1</u>	<u>0</u> <u>2</u> <u>2</u> <u>0</u>	<u>2</u> <u>3</u> <u>1</u> <u>1</u>	<u>3</u> <u>4</u> <u>2</u> <u>2</u>	<u>3</u> <u>5</u> <u>3</u> <u>3</u>	<u>5</u> <u>6</u> <u>4</u> <u>4</u>
t	<u>3</u> <u>3</u>	<u>3</u> <u>2</u> <u>4</u> <u>2</u>	<u>2</u> <u>1</u> <u>3</u> <u>1</u>	<u>0</u> <u>2</u> <u>2</u> <u>0</u>	<u>2</u> <u>3</u> <u>1</u> <u>1</u>	<u>3</u> <u>4</u> <u>2</u> <u>2</u>	<u>3</u> <u>5</u> <u>3</u> <u>3</u>

cost	operation	input	output
0	(copy)	c	c
0	(copy)	a	a
1	insert	*	t
1	insert	*	c
1	insert	*	a
0	(copy)	t	t

		c	a	t	c	a	t
	0	1 1	2 2	3 3	4 4	5 5	6 6
c	1 1	0 2 2 0	2 3 1 1	3 4 2 2	3 5 3 3	5 6 4 4	6 7 5 5
a	2 2	2 1 3 1	0 2 2 0	2 3 1 1	3 4 2 2	3 5 3 3	5 6 4 4
t	3 3	3 2 4 2	2 1 3 1	0 2 2 0	2 3 1 1	3 4 2 2	3 5 3 3

cost	operation	input	output
0	(copy)	c	c
0	(copy)	a	a
0	(copy)	t	t
1	insert	*	c
1	insert	*	a
1	insert	*	t

Απόσταση διόρθωσης από όλους τους όρους του λεξικού;

- Δοθέντος ενός (ανορθόγραφου) ερωτήματος, υπολογίζουμε την απόσταση διόρθωσης από όλους τους όρους του λεξικού
 - Ακριβό και αργό
- Μπορούμε να μειώσουμε τον αριθμό των υποψήφιων όρων του ευρετηρίου;
 - Μόνο λέξεις που αρχίζουν από το ίδιο γράμμα
 - Να θεωρήσουμε ότι υπάρχει * και ένα permutation
 - Να βρούμε λέξεις με αρκετά κοινά k-γράμματα και να περιορίσουμε τον υπολογισμό απόστασης με αυτές

Χρήση των αποστάσεων διόρθωσης

1. Έχοντας μια ερώτηση, πρώτα απαρίθμησε όλες τις ακολουθίες χαρακτήρων μέσα σε μια προκαθορισμένη (σταθμισμένη) απόσταση διόρθωσης (π.χ., 2)
2. Βρες την τομή αυτού του συνόλου με τις «σωστές» λέξεις
3. *Πρότεινε τους όρους* που βρήκες στο χρήστη
 - Ψάξε όλες τις πιθανές διορθώσεις στο αντεστραμμένο ευρετήριο και επέστρεψε όλα τα έγγραφα ... αργό
 - Μπορούμε να *επιστρέψουμε τα έγγραφα* μόνο για την πιο πιθανή διόρθωση
 - Η εναλλακτική λύση παίρνει τον έλεγχο από το χρήστη αλλά κερδίζουμε ένα γύρο διάδρασης

Σταθμισμένη απόσταση διόρθωσης

- Το βάρος μιας πράξης εξαρτάται από τον ποιο χαρακτήρα (χαρακτήρες) περιλαμβάνει
 - Στόχος να λάβει υπόψη λάθη OCR ή πληκτρολόγησης
Παράδειγμα: m πιο πιθανό να πληκτρολογηθεί ως n παρά ως q
 - Οπότε η αντικατάσταση του m από n έχει μικρότερη απόσταση διόρθωσης από την απόσταση του από το q
 - Διατύπωση ως πιθανοτικό μοντέλο
- Προϋποθέτει ως είσοδο έναν **πίνακα βαρών**
- *Πως θα μετατρέψουμε το δυναμικό προγραμματισμό για να χειριστούμε τα βάρη;*

Επικάλυψη k -γραμμάτων

Εναλλακτικός ορισμός απόστασης: βάση των κοινών k -γραμμάτων

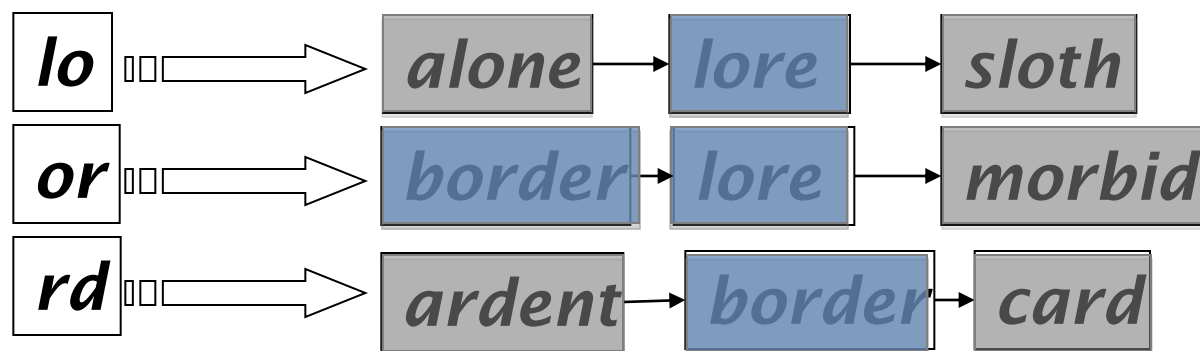
- Απαρίθμησε όλα τα k -γράμματα στον όρο της ερώτησης
- Χρησιμοποίησε το ευρετήριο k -γραμμάτων για να ανακτήσεις όλους τους όρους του λεξικού που ταιριάζουν κάποιο από τα k -γράμματα του ερωτήματος
- Ανέκτησε όλους τους όρους του λεξικού που ταιριάζουν κάποιο (\geq **κατώφλι**) αριθμό από τα k -γράμματα του ερωτήματος

Παράδειγμα με 3-γράμματα

- Έστω ότι το κείμενο είναι ***november***
 - Τα τριγράμματα είναι *nov*, *one*, *vet*, *emb*, *mbe*, *ber*.
- Για το ερώτημα ***december***
 - Τα τριγράμματα είναι *dec*, *ece*, *cem*, *emb*, *mbe*, *ber*.
- Άρα 3 τριγράμματα επικαλύπτονται (από τα 6 κάθε όρου)

Ταίριασμα τριγραμμμάτων

- Έστω το ερώτημα **lord** – θέλουμε να βρούμε τις λέξεις που ταιριάζουν τουλάχιστον 2 από τα 3 διγράμματα (**lo**, **or**, **rd**)



Η τυπική συγχώνευση θα τα δώσει

- Πως μπορούμε να το χρησιμοποιήσουμε ως ένα κανονικοποιημένο μέσω επικάλυψης;

Μια δυνατότητα – συντελεστής Jaccard

- Συνήθης μέτρηση της επικάλυψης

Έστω X και Y δύο σύνολα, ο **συντελεστής Jaccard** ορίζεται ως:

$$|X \cap Y| / |X \cup Y|$$

- Ίσος με 1 όταν τα X και Y έχουν τα ίδια στοιχεία και 0 όταν είναι ξένα
- Τα X and Y δε χρειάζεται να έχουν το ίδιο μέγεθος
- Πάντα μεταξύ του 0 και του 1
 - Το κατώφλι καθορίζει αν υπάρχει ταίριασμα, πχ., αν J.C. > 0.8, τότε ταίριασμα

Πως το υπολογίζουμε αποδοτικά στην περίπτωση των k -γραμμάτων;

Διόρθωση εξαρτώμενη από το περιβάλλον

Κείμενο: *I flew from Heathrow to Narita.*

- Θεωρείστε το ερώτημα-φράση “*flew form Heathrow*”
- Θα θέλαμε να απαντήσουμε
Did you mean “*flew from Heathrow*”?

Γιατί δεν υπήρχαν έγγραφα που να ταιριάζουν το ερώτημα φράση

Διόρθωση βασισμένη στα συμφραζόμενα

- Χρειάζεται συμφραζόμενο περιβάλλον για να το πιάσει αυτό.

Πρώτη ιδέα:

1. Ανέκτησε τους όρους του λεξικού που είναι κοντά (σε σταθμισμένη απόσταση διόρθωσης) από κάθε όρο του ερωτήματος
2. Δοκίμασε όλες τις πιθανές φράσεις που προκύπτουν κρατώντας κάθε φορά μια λέξη σταθερή
 - *flew from heathrow*
 - *fled form heathrow*
 - *flea form heathrow*
3. **Hit-based spelling correction**: Πρότεινε την εναλλακτική με τα περισσότερα hits

Διόρθωση βασισμένη στα συμφραζόμενα

Εναλλακτική Προσέγγιση

1. Σπάσε της φράση σε σύζευξη biwords.
2. Ψάξε τα biwords που χρειάζονται διόρθωση μόνο ενός όρου.
3. Απαρίθμησε μόνο τις φράσεις που περιέχουν «κοινά» biwords.

Γενικά Θέματα

- Θέλουμε να δούμε διαφορετικές απαντήσεις στο “Did you mean?”
- Ποιες θα επιλέξουμε να παρουσιάσουμε στο χρήστη;
 - Αυτή που εμφανίζεται στα περισσότερα έγγραφα
 - Ανάλυση του Query log

ΦΩΝΗΤΙΚΗ ΔΙΟΡΘΩΣΗ (SOUNDEX)

Soundex

Φωνητική διόρθωση: ερώτημα που «ακούγεται» όπως ο σωστός όρος

- Κλάση ευριστικών για την επέκταση ενός ερωτήματος σε φωνητικά (**phonetic**) ισοδύναμα
 - Εξαρτώνται από τη γλώσσα – κυρίως για ονόματα
 - Π.χ., *chebyshev* → *tchebycheff*
- Προτάθηκε από το U.S. census ... το 1918 (για ονόματα από εγληματίες!)

Βασική ιδέα: **“phonetic hash”**: όροι που «ακούγονται» ίδιοι κατακερματίζονται στην ίδια θέση

Soundex – τυπικός αλγόριθμος

- Μετέτρεψε κάθε token προς δεικτοδότηση σε μια μορφή 4-χαρακτήρων
- Το ίδιο και για τους όρους του ερωτήματος
- Κατασκεύασε ένα ανεστραμμένο ευρετήριο από αυτούς τους 4-χαρακτήρες στον αρχικό όρο και ψάξε στο ευρετήριο τις μειωμένες μορφές
 - (όταν το ερώτημα χρειάζεται φωνητικό ταίριασμα)
- <http://www.creativyst.com/Doc/Articles/SoundEx1/SoundEx1.htm#Top>

Soundex – τυπικός αλγόριθμος

1. Κράτησε τον πρώτο χαρακτήρα της λέξης
2. Μετάτρεψε όλες τις εμφανίσεις των παρακάτω όρων σε '0' (zero):
'A', 'E', 'I', 'O', 'U', 'H', 'W', 'Y'.
3. Άλλαξε τα γράμματα σε αριθμούς ως ακολούθως:
 - B, F, P, V → 1
 - C, G, J, K, Q, S, X, Z → 2
 - D, T → 3
 - L → 4
 - M, N → 5
 - R → 6

Soundex συνέχεια

4. Σβήσε όλα τα ζεύγη συνεχόμενων αριθμών
5. Σβήσε όλα τα υπομένοντα 0
6. Πρόσθεσε 0 στο τέλος και επέστρεψε τις τέσσερις πρώτες θέσεις που θα είναι της μορφής <uppercase letter> <digit> <digit> <digit>.

Π.χ., *Herman* γίνεται H655.

Το *hermann* δίνει τον ίδιο κωδικό;

ΤΕΛΟΣ 3^{ου} 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*

Επεξεργασία γενικών ερωτημάτων

- Π.χ., Θεωρείστε το ερώτημα:
se*ate AND fil*er

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

Διόρθωση λαθών στο ερώτημα

- Βασική **έμφαση στα ερωτήματα**
 - Π.χ., το ερώτημα ***Alanis Morissett***
- Μπορεί είτε
 - Να ανακτήσουμε τα έγγραφα που έχουν δεικτοδοτηθεί κάτω από τη σωστή ορθογραφία, ή
 - Να επιστρέψουμε διάφορα προτεινόμενα ερωτήματα με σωστή ορθογραφία
 - *Did you mean ... ?*