Introduction to

Information Retrieval

ΠΛΕ70: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Διάλεξη 3: Λεξικά και 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

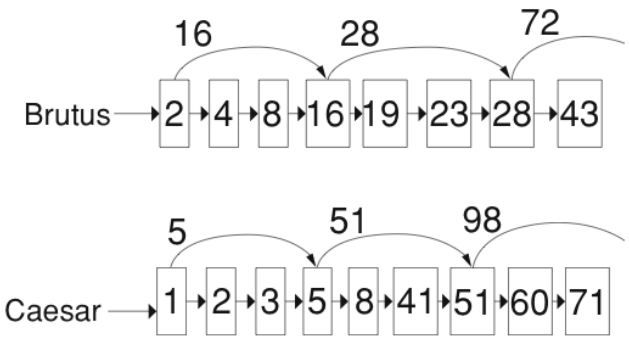
➢Ποια είναι τα διαχωριστικά (κενό, απόστροφος, ενωτικά (hyphen)

1. Προσδιορισμός Λεξιλογίου όρων

Κλάσεις ισοδύναμων tokens -> όρους που θα εισαχθούν στο ευρετήριο

- Αριθμοί
- Κεφαλαία/Μικρά
- Λημματοποίση 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$_6$"*

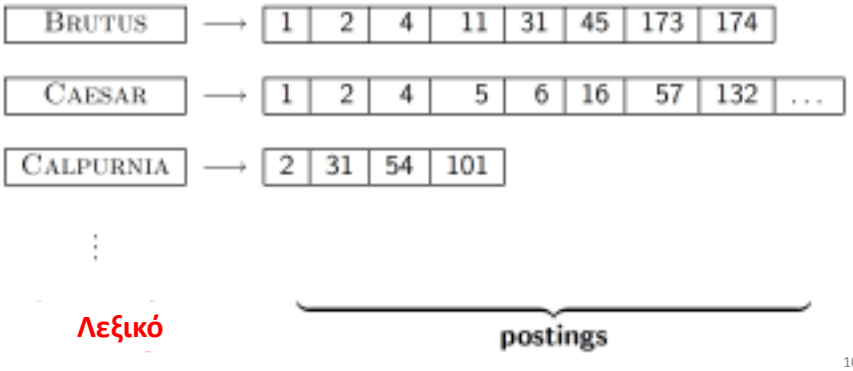
TO, 993427:
‹ 1: ‹7, 18, 33, 72, 86, 231›;
2: ‹1, 17, 74, 222, 255›;
4: ‹8, 16, 190, 429, 433›;
5: ‹363, 367›;
7: ‹13, 23, 191›; . . . ›
BE, 178239:
‹ 1: ‹17, 25›;
4: ‹17, 191, 291, 430, 434›;
5: ‹14, 19, 101›; . . . ›

Τι θα δούμε σήμερα;

- Δομές δεδομένων για Λεξικά
- Ανάκτηση Ανεκτική σε Σφάλματα "Tolerant"
  - Ερωτήματα με Wild-card («χαρακτήρων μπαλαντέρ)*
  - Διόρθωση ορθογραφικών λαθών
  - *Soundex – φωνητική αναζήτηση*

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

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

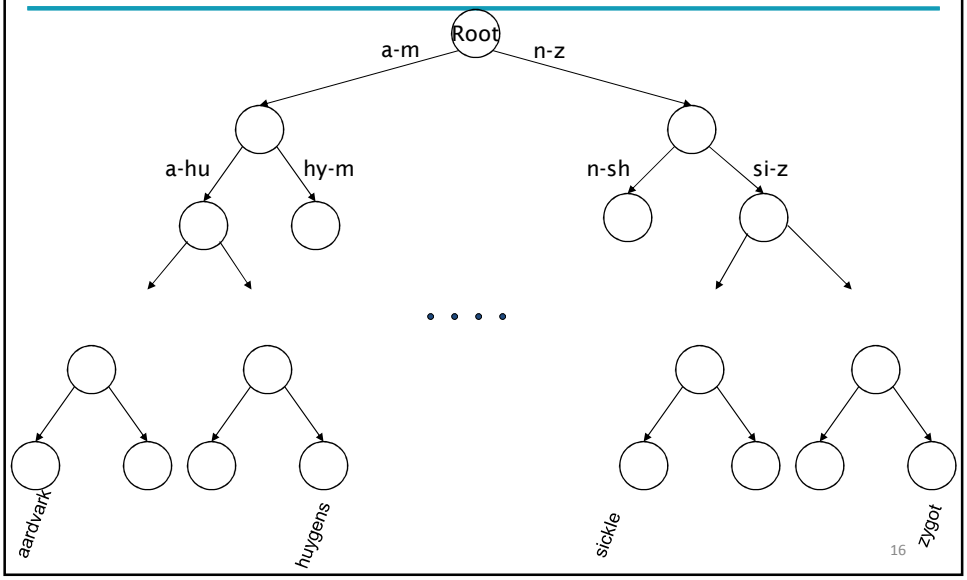
+:

- 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

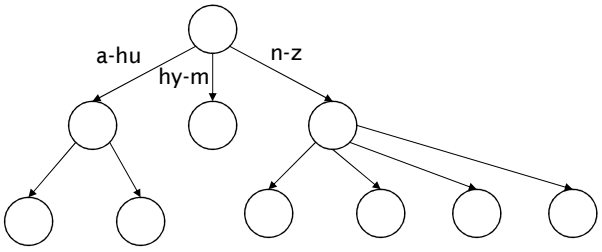
Δέντρα αναζήτησης: Δυαδικό δέντρο



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 ακριβό
  - Αλλά τα B-δέντρα καλύτερα

ΕΡΩΤΗΜΑΤΑ ΜΕ *

Ερωτήματα με Wild-card (*)

- ***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
- **** 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)*
  - 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 την 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!
- Εναλλακτική λύση: Μετάτρεψε τις ερωτήσεις έτσι ώστε τα * να εμφανίζονται στο τέλος

Permuterm Index (ευρετήριο αντιμετατεθειμένων όρων)

Ευρετήριο Permuterm

Βασική ιδέα: Δεξιά περιστροφή (rotation) του όρου του ερωτήματος προς τα δεξιά ώστε το * στο τέλος

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

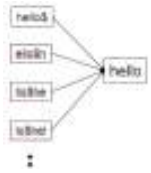
όπου $ ένα ειδικός χαρακτήρας που σηματοδοτεί το τέλος μιας λέξης

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



Ευρετήριο Permuterm

Παράδειγμα

Ερώτημα
m*n -> m*n$ -> n$m*
Ευρετήριο
moron -> moron$ -> στο ευρετήριο: $moron, n$moro, on$mor ron$mo oron$m moron$
man -> man$ -> στο ευρετήριο: man, nma an$m man$

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: ≈ τετραπλασιάζει το μέγεθος του λεξικού*

Εμπειρική παρατήρηση για τα Αγγλικά

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

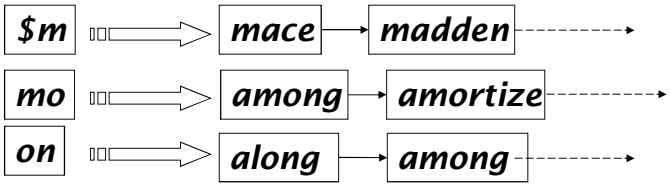
- Απαρίθμησε όλα τα *k*-γράμματα (ακολουθίες *k* γραμμάτων) που εμφανίζονται σε κάθε όρο
  - *π.χ.,* για το κείμενο "***April is the cruelest month***" έχουμε τα 2-γράμματα (*bigrams*)

    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

29

Επεξεργασία ερωτημάτων

- Ερώτημα ***mon**** τώρα γίνεται

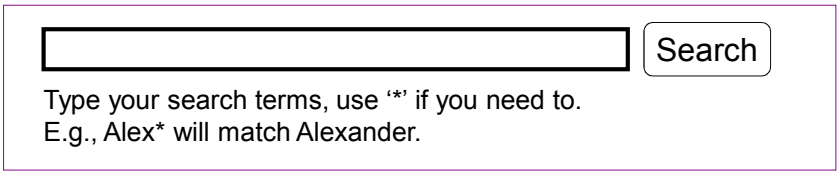
  $m *AND* ***mo*** *AND* ***on***

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

30

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;

Μερικά διαδικαστικά

Τρία σύνολα (40-50% του βαθμού) (ατομικά)

Ανάθεση	Προθεσμία
13/3	3/4
3/4	25/4
25/4	29/5

Απαλλακτική εργασία (50-60% του βαθμού) (ομάδες έως 2 ατόμων)
Προθεσμία την ημέρα της εξέτασης με μια σύντομη παρουσίαση πριν το Πάσχα
Θέμα: *Σχεδιασμός και υλοποίηση ενός απλό σύστημα ΑΠ για κάποια μικρή συλλογή δεδομένων (διαφορετικό για κάθε ομάδα)*
(ιδέες: email, προγράμματα σε C, Java, twitter text, απλά κείμενα, κλπ)
Τελική εξέταση (κλειστά βιβλία) (50-60% του βαθμού)

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
    - 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 στις 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/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 φτιάχνουμε την απεικόνιση ερωτήματος –εγγράφου

Διόρθωση λαθών στο ερώτημα

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

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)

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 διάφορους ορισμούς εγγύτητας:
  - Την απόσταση διόρθωσης -- edit distance (Levenshtein distance)
  - Την σταθμισμένη απόσταση διόρθωσης -- 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 ένα χαρακτήρα
- Π.χ., η απόσταση διόρθωσης από ***dof*** σε ***dog*** είναι 1
  - Από ***cat*** σε ***act*** είναι 2 (Μόνο 1 με αντιμετάθεση)
  - Από ***cat*** σε ***dog*** είναι 3.

39

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.

40

Υπολογισμός απόστασης διόρθωσης

String s_2

String s_1

		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

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

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

String s_1

		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

cost of getting here from my upper left neighbor (copy or replace)	cost of getting here from my upper neighbor (delete)
cost of getting here from my left neighbor (insert)	the minimum of the three possible "movements"; the cheapest way of getting here

Υπολογισμός απόστασης διόρθωσης

String s_2

String s_1

		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

Αν ο χαρακτήρας είναι ίδιος (s1[i] = s2[j])

- m[i-1, j-1] Nothing – just copy (c, f -> ca, fa) Cost: m[i-1, j-1] + 0
- m[i-1, j] Delete (c,fa ->ca, fa, delete a)) Cost: m[i-1, j] + 1
- m[i, j-1] Insert (πως από ca, f θα πάμε σε ca, fa -- insert a): Cost m[i, j-1] + 1

Υπολογισμός απόστασης διόρθωσης

```
LEVENSHTEINDISTANCE(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7      do if s1[i] = s2[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[|s1|, |s2|]
```

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

Υπολογισμός απόστασης διόρθωσης

String s_2

String s_1

		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

Αν ο χαρακτήρας δεν είναι ίδιος (s1[i] <> s2[j])

- m[i-1, j-1] (Replace) (*c, fa ->* ca,fas, replace a with s) Cost: m[i-1, j-1] + 1
- m[i-1, j] (Delete) (πως από c, fas πάμε σε ca, fas delete a) Cost: m[i-1, j] + 1
- m[i, j-1] (Insert) (*ca, fa ->* ca, fas – insert s) Cost: m[i, j-1] + 1

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** $m[i, j] = \min\{m[i\text{-}1, j]+1, m[i, j\text{-}1]+1, m[i\text{-}1, j\text{-}1]\}$
9 **else** $m[i, j] = \min\{m[i\text{-}1, j]+1, m[i, j\text{-}1]+1, m[i\text{-}1, j\text{-}1]+1\}$
10 **return** $m[|s_1|, |s_2|]$

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

47

Levenshtein distance: Algorithm

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** $m[i, j] = \min\{m[i\text{-}1, j]+1, m[i, j\text{-}1]+1, m[i\text{-}1, j\text{-}1]\}$
9 **else** $m[i, j] = \min\{m[i\text{-}1, j]+1, m[i, j\text{-}1]+1, m[i\text{-}1, j\text{-}1]+1\}$
10 **return** $m[|s_1|, |s_2|]$

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

48

Levenshtein distance: Algorithm

```
LevenshteinDistance(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7      do if s1[i] = s2[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[|s1|, |s2|]
```

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

Levenshtein distance: Algorithm

```
LevenshteinDistance(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7      do if s1[i] = s2[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[|s1|, |s2|]
```

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

Levenshtein distance: Algorithm

```
LevenshteinDistance(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7       do if s1[i] = s2[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[|s1|, |s2|]
```

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

51

Υπολογισμός απόστασης Levenshtein

cost of getting here from my upper left neighbor (copy or replace)	cost of getting here from my upper neighbor (delete)
cost of getting here from my left neighbor (insert)	the minimum of the three possible "movements"; the cheapest way of getting here

Υπολογισμός απόστασης Levenshtein: παράδειγμα

			f		a		s		t	
		0		**1**		**2**		**3**		**4**
c		**1**	*1*	*2*	**2**	3	**3**	4	**4**	5
		1	*2*	*1*	**2**	**2**	**3**	**3**	**4**	**4**
a		**2**	**2**	**2**	*1*	*3*	*3*	*4*	4	5
		2	3	**2**	*3*	*1*	*2*	*2*	**3**	**3**
t		**3**	**3**	**3**	3	**2**	*2*	*3*	*2*	*4*
		3	4	**3**	4	**2**	*3*	*2*	*3*	*2*
s		**4**	**4**	**4**	4	**3**	**2**	3	*3*	*3*
		4	5	**4**	5	**3**	4	**2**	*3*	*3*

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
	0	1 1	2 2	3 3	4 4
o	1 / 1				
s	2 / 2				
l	3 / 3				
o	4 / 4				

Υπολογισμός απόστασης: παράδειγμα

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

Υπολογισμός απόστασης: παράδειγμα

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

Υπολογισμός απόστασης: παράδειγμα

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

Υπολογισμός απόστασης: παράδειγμα

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

Υπολογισμός απόστασης: παράδειγμα

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

87

Υπολογισμός απόστασης: παράδειγμα

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

88

Υπολογισμός απόστασης: παράδειγμα

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

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

		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**

		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	insert	*	w

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

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 / 2	3 / 2	2 / 3	4 / 2	4 / 3	5 / 3
s	2 / 2	1 / 3	2 / 1	2 / 2	3 / 2	3 / 3	3 / 3	4 / 3	3 / 4
l	3 / 3	3 / 4	2 / 2	2 / 3	3 / 2	3 / 3	4 / 3	4 / 4	4 / 4
o	4 / 4	4 / 5	3 / 3	3 / 4	3 / 3	2 / 4	4 / 2	4 / 3	5 / 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	2	2	3	2	4	4	5
	1	2	1	2	2	3	2	3	3
s	2	1	2	2	3	3	3	3	4
	2	3	1	2	2	3	3	4	3
l	3	3	2	2	3	3	4	4	4
	3	4	2	3	2	3	3	4	4
o	4	4	3	3	3	2	4	4	5
	4	5	3	4	3	4	2	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	2	2	3	2	4	4	5
	1	2	1	2	2	3	2	3	3
s	2	1	2	2	3	3	3	3	4
	2	3	1	2	2	3	3	4	3
l	3	3	2	2	3	3	4	4	4
	3	4	2	3	2	3	3	4	4
o	4	4	3	3	3	2	4	4	5
	4	5	3	4	3	4	2	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
	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**

		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
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
	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
1	insert	*	a
1	insert	*	t
1	insert	*	c
0	(copy)	a	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
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

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 είσοδος ένας πίνακας βαρών
- *Πως θα μετατρέψουμε το δυναμικό προγραμματισμό για να χειριστούμε τα βάρη;*

Χρήση των αποστάσεων διόρθωσης

1. Δοθείσας μιας ερώτησης, 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

Εναλλακτικά,

- 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

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;
  - Να χρησιμοποιήσουμε επικάλυψη με n-γράμματα
  - Ή Απαριθμούμε όλα σε απόσταση1, 2 κλπ
- Μπορεί να χρησιμοποιηθεί και για τη διόρθωση ορθογραφικών λαθών

Επικάλυψη *n*-γραμμάτων

- Απαρίθμησε όλα το *n*-γράμματα στον όρο της ερώτησης και στο λεξικό
- Χρησιμοποίησε το ευρετήριο *n*-γραμμάτων για να ανακτήσεις όλους τους όρους του λεξικού που ταιριάζουν κάποιο από τα *n*-γράμματα του ερωτήματος
- Κατώφλι (threshold) βασισμένο στον αριθμό των κοινών *n*-γραμμάτων

Παράδειγμα με 3-γράμματα

- Έστω ότι το κείμενο είναι ***november***
  - Τα τριγράμματα είναι *nov, ove, vem, 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)
- 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 (J.C.) ορίζεται ως:

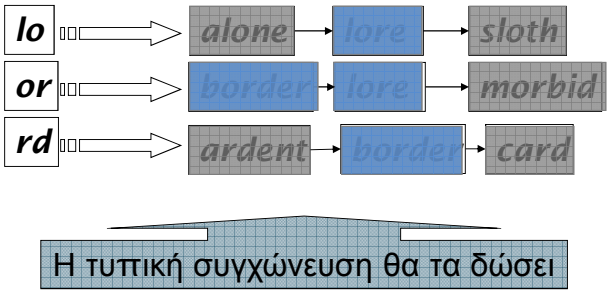
$$|X \cap Y| / |X \cup Y|$$

  - 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

Ταίριασμα τριγραμμάτων

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



Τροποποίηση ώστε να χρησιμοποιεί Jaccard (ή άλλη) μέτρηση.

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

Μια άλλη προσέγγιση

- Σπάσε της φράση σε σύζευξη biwords.
- Ψάξε τα biwords που χρειάζονται διόρθωση μόνο ενός όρου.
- 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

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*