Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
Άνοιξη 2006

HY463 - Συστήματα Ανάκτησης Πληροφοριών
Information Retrieval (IR) Systems

(A) Προεπεξεργασία Κειμένου (Text Preprocessing)
(B) Γλώσσες Επερώτησης για Ανάκτηση Πληροφοριών

Γιάννης Τζίτζικας

Διάλεξη : 5
Ημερομηνία : 8-3-2006

(A) Προεπεξεργασία κειμένου : Διάρθρωση Διάλεξης

- Εισαγωγή
- Λεξιλογική ανάλυση (Lexical Analysis)
- Αποκλεισμός Λέξεων (Stopwords)
- Στελέχωση Κειμένου (Stemming)
  - Manual
  - Table Lookup
  - Successor Variety
  - n-Grams
  - Affix Removal (Porter's algorithm)

Προεπεξεργασία Κειμένου

- Σκεπτικό
  - δεν είναι όλες οι λέξεις ενός κειμένου κατάλληλες για την παράσταση του περιεχομένου του (μερικές λέξεις φέρουν περισσότερο νόημα από άλλες)

- Στόχοι της προεπεξεργασίας κειμένου
  - βελτίωση της αποτελεσματικότητας (effectiveness)
  - βελτίωση της αποδοτικότητας (efficiency) της ανάκτησης
  - προσπάθεια ελέγχου (κυρίως μείωσης) του λεξιλογίου
    - και εκ τούτου μείωσης του μεγέθους των ευρετηρίων

Προεπεξεργασία και Ευρετήριο

- **Λειτουργίες Κειμένου (Text Operations)** σχηματίζουν τις λέξεις ευρετηρίου (tokens, index terms).

Documents	Indexing Items k_1	k_2	...	k_i	...	k_t
d_1	$c_{1,1}$	$c_{2,1}$	...	$c_{i,1}$	...	$c_{t,1}$
d_2	$c_{1,2}$	$c_{2,2}$	...	$c_{i,2}$	...	$c_{t,2}$
...	...	...	...	...	...	...
d_i	$c_{1,j}$	$c_{2,j}$	...	$c_{i,j}$	...	$c_{t,j}$
...	...	...	...	...	...	...
d_N	$c_{1,N}$	$c_{2,N}$	...	$c_{i,N}$	...	$c_{t,N}$

Φάσεις Προεπεξεργασίας

[α] Λεξιλογική ανάλυση

- αναγνώριση *αριθμών*, *λέξεων*, *διαχωριστικών*, *σημείων στίξεως*, κλπ

[β] Αποκλεισμός λέξεων (stopwords)

- απαλοιφή λέξεων με πολύ μικρή διακριτική ικανότητα (*άρθρα*, *αντωνυμίες*, *κτητικές αντωνυμίες*, κλπ)

[γ] Στελέχωση (stemming) των εναπομεινάντων λέξεων

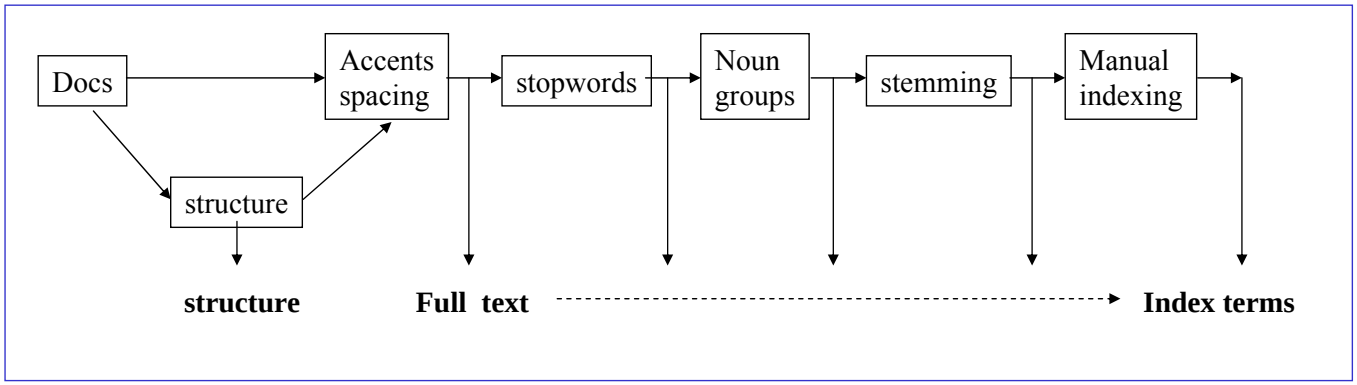
- απαλοιφή *καταλήξεων/προθεμάτων* (αυτοκίνητο, αυτοκίνητα, αυτοκινήτων) για την ανάκτηση των κειμένων που περιέχουν μορφολογικές παραλλαγές των λέξεων της επερώτησης

[δ] Επιλογή των λέξεων που θα χρησιμοποιηθούν στον ευρετηριασμό

- συχνά γίνεται βάσει του μέρους του λόγου (ουσιαστικά, επίθετα, επιρρήματα, ρήματα)

Φάσεις Προεπεξεργασίας (II)

Από το πλήρες κείμενο στους όρους ευρετηρίου



[α] **Λεξιλογική Ανάλυση** (Lexical Analysis)

[α] **Λεξιλογική Ανάλυση** (Lexical Analysis)

Σκοπός: αναγνώριση ελάχιστων μονάδων (identify tokens)

- αναγνώριση *αριθμών*, *λέξεων*, *διαχωριστικών*, *σημείων στίξεως*, κλπ

Περιπτώσεις που απαιτούν προσοχή:

- Λέξεις που περιέχουν ψηφία
  - O2, βιταμίνη B6, B12, Windows98
- Παύλες (hyphens)
  - "state of the art" vs "state-of-the-art"
  - "Jean-Luc Hainaut", "Jean-Roch Meurisse", F-16, MS-DOS
- Σημεία στίξεως (punctuations)
  - OS/2, .NET, command.com
- Μικρά-κεφαλαία
  - συνήθως όλα μετατρέπονται σε μικρά

[α] Λεξιλογική Ανάλυση (II)

- Λεξιλογική Ανάλυση για Επερωτήσεις
  - Όπως και για το κείμενο, συν αναγνώριση χαρακτήρων ελέγχου, όπως
    - λογικοί τελεστές, π.χ. AND, OR, NOT,
    - τελεστές εγγύτητας (proximity operators),
    - κανονικές εκφράσεις (regular expressions), κτλ.

- Τρόποι υλοποίησης ενός Λεξιλογικού Αναλυτή
  - (α) χρήση μιας γεννήτριας λεξιλογικών αναλυτών (lexical analyzer generator), όπως τον lex
    - η καλύτερη επιλογή αν υπάρχουν σύνθετες περιπτώσεις
  - (β) συγγραφή (προγραμματισμός) ενός λεξιλογικού αναλυτή με το χέρι
    - η χειρότερη επιλογή (επιρρεπή σε σφάλματα)
  - (γ) συγγραφή (προγραμματισμός) ενός λεξιλογικού αναλυτή σαν μια μηχανή πεπερασμένων καταστάσεων (finite state machine)

[β] Λέξεις Αποκλεισμού (**Stopwords**)

[β] Λέξεις Αποκλεισμού (**Stopwords**)

Απαλοιφή λέξεων με πολύ μικρή διακριτική ικανότητα (*άρθρα*, *αντωνυμίες*, *κτητικές αντωνυμίες*, κλπ)

- e.g. “a”, “the”, “in”, “to”; pronouns: “I”, “he”, “she”, “it”.

- Οφέλη
  - μείωση μεγέθους ευρετηρίου **(έως και 40%**)
- Παρατηρήσεις
  - Οι λέξεις αποκλεισμού εξαρτώνται από τη γλώσσα και τη συλλογή
  - Not every frequent english word should be in the list
    - Top 200 English words include «time, war, home, life, water, world»
    - In a CS corpus we could add to the stoplist the words: «computer, program, source, machine, language»
- Προβλήματα
  - q=“**to be or not to be**”
  - (για το λόγο αυτό μερικές Μηχανές Αναζήτησης του Ιστού δεν κάνουν προεπεξεργασία)

Example: Stopwords for the English language

a be had it only she was about because has its of some we after been have last on such were all but he more one than when also by her most or that which an can his mr other the who any co if mrs out their will and corp in ms over there with are could inc mz s they would as for into no so this up at from is not says to

Example: Stopwords for the French language

a afin ah ai aie aient aies ailleurs ainsi ait alentour alias allais allaient allait allons allez alors Ap. Apr. aprs aprs demain arrire as assez attendu au aucun aucune au dedans au dehors au dela au dessous au dessus au devant audit aujourd aujourdhui auparavant auprs auquel aura aurai auraient aurais aurait auras aurez auriez aurions aurons auront aussi aussitôt autant autour autre autrefois autres autrui aux auxdites auxdits auxquelles auxquels avaient avais avait avant avant hier avec avez aviez avions avoir avons ayant ayez ayons B bah banco be beaucoup ben bien bientôt bis bon C c Ca ça ça cahin caha car ce ce ceans ceci cela celle celle ci celle la celles celles ci celles la celui celui ci celui la cent cents cependant certain certaine certaines certains certes ces cest a dire cet cette ceux ceux ci ceux la cf. cg cgr chacun chacune chaque cher chez ci ci ci aprs ci dessous ci dessus cinq cinquante cinquante cinq cinquante deux cinquante et un cinquante huit cinquante neuf cinquante quatre cinquante sept cinquante six cinquante trois cl cm cm combien comme comment contrario contre crescendo D d dabord daccord daffilee dailleurs dans daprs darrache pied davantage de debout dedans dehors deja dela demain demblee depuis derechef derrire des ds desdites desdits desormais desquelles desquels dessous dessus deux devant devers dg die differentes differents dire dis disent dit dito divers diverses dix dix huit dix neuf dix sept dl dm donc dont dorenavant douze du dû dudit duquel durant E eh elle elle elles elles en en en encore enfin ensemble ensuite entre entre temps envers environ es s est et et/ou etaient etais etait etant etc ete êtes etiez etions être eu eue eues euh eûmes eurent eus eusse eussent eusses eussiez eussions eut eût eûtes eux exprs extenso extremis F facto fallait faire fais faisais faisait faisaient faisons fait faites faudrait faut fi flac fors fort forte fortiori frais fûmes fur furent fus fusse fussent fusses fussiez fussions fut fût fûtes G GHz gr grosso gure H ha han haut he hein hem heu hg hier hl hm hm hola hop hormis hors hui huit hum I ibidem ici ici bas idem il il illico ils ils ipso item J j jadis jamais je je jusqu jusqua jusquau jusquaux jusque juste K kg km km² L l la la la la la bas la dedans la dehors la derrire la dessous la dessus la devant la haut laquelle lautre le le lequel les les ls lesquelles lesquels leur leur leurs lez loin lon longtemps lors lorsqu lorsque lui lui lun lune M m m m ma maint mainte maintenant maintes maints mais mal malgre me même mêmes mes mg mgr MHz mieux mil mille milliards millions minima ml mm mm² modo moi moi moins mon moult moyennant mt N n nagure ne neanmoins neuf ni n° non nonante nonobstant nos notre nous nous nul nulle O ô octante oh on on ont onze or ou où ouais oui outre P par parbleu parce par ci par dela par derrire par dessous par dessus par devant parfois par la parmi partout pas passe passim pendant personne petto peu peut peuvent peux peut être pis plus plusieurs plutôt point posteriori pour pourquoi pourtant prealable prs presqu presque primo priori prou pu puis puisqu puisque Q qu qua quand quarante quarante cinq quarante deux quarante et un quarante huit quarante neuf quarante quatre quarante sept quarante six quarante trois quasi quatorze quatre quatre vingt quatre vingt cinq quatre vingt deux quatre vingt dix quatre vingt dix huit quatre vingt dix neuf quatre vingt dix sept quatre vingt douze quatre vingt huit quatre vingt neuf quatre vingt onze quatre vingt quatorze quatre vingt quatre quatre vingt quinze quatre vingts quatre vingt seize quatre vingt sept quatre vingt six quatre vingt treize quatre vingt trois quatre vingt un quatre vingt une que quel quelle quelles quelqu quelque quelquefois quelques quelques unes quelques uns quelquun quelquune quels qui quiconque quinze quoi quoiqu quoique R revoici revoila rien S s sa sans sauf se secundo seize selon sensu sept septante sera serai seraient serais serait seras serez seriez serions serons seront ses si sic sine sinon sitôt situ six soi soient sois soit soixante soixante cinq soixante deux soixante dix soixante dix huit soixante dix neuf soixante dix sept soixante douze soixante et onze soixante et un soixante et une soixante huit soixante neuf soixante quatorze soixante quatre soixante quinze soixante seize soixante sept soixante six soixante treize soixante trois sommes son sont soudain sous souvent soyez soyons stricto suis sur sur le champ surtout sus T t t ta tacatac tant tantôt tard te tel telle telles tels ter tes toi toi ton tôt toujours tous tout toute toutefois toutes treize trente trente cinq trente deux trente et un trente huit trente neuf trente quatre trente sept trente six trente trois trs trois trop tu tu U un une unes uns USD V va vais vas vers veut veux via vice versa vingt vingt cinq vingt deux vingt huit vingt neuf vingt quatre vingt sept vingt six vingt trois vis a vis vite vitro vivo voici voila voire volontiers vos votre vous vous W X y y Z zero

[β] Απαλοιφή λέξεων Αποκλεισμού: Τρόποι

Τρόποι Υλοποίησης

1/ Απαλοιφή των λέξεων αποκλεισμού μετά το τέλος της λεξιλογικής ανάλυσης

- Μπορούμε να αποθηκεύσουμε τις λέξεις αυτές σε έναν hashtable για να τις αναγνωρίζουμε γρήγορα (σε σταθερό χρόνο)

2/ Απαλοιφή των λέξεων αποκλεισμού κατά τη διάρκεια της λεξιλογικής ανάλυσης

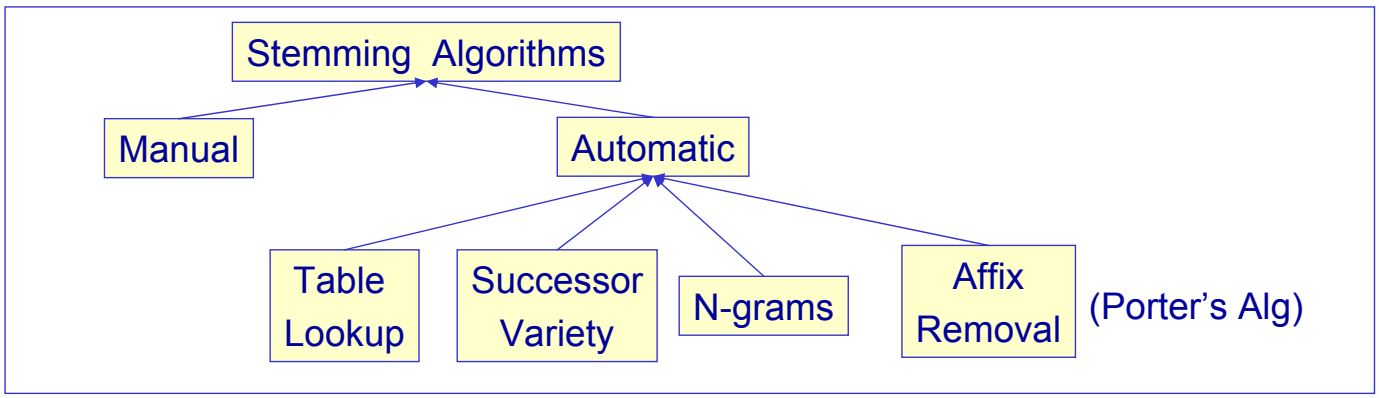
- Πιο γρήγορη προσέγγιση αφού η λεξιλογική ανάλυση θα γίνει έτσι και αλλιώς και η αφαίρεση των λέξεων αποκλεισμού δεν απαιτεί επιπλέον χρόνο

[γ] Στελέχωση Κειμένου (Stemming)

[γ] Στελέχωση Κειμένου (Stemming)

- Υποβίβαση λέξεων στη ρίζα τους για ανεξαρτησία από τις μορφολογικές παραλλαγές των λέξεων
  - «αυτοκίνητο», «αυτοκίνητα», «αυτοκινήτων»
  - “computer”, “computational”, “computation” all reduced to same token “compute”
- Στόχοι
  - Βελτίωση αποτελεσματικότητας (κυρίως της ανάκλησης)
  - Μείωση του μεγέθους του ευρετηρίου
    - Συγκεκριμένα του λεξιλογίου του ευρετηρίου

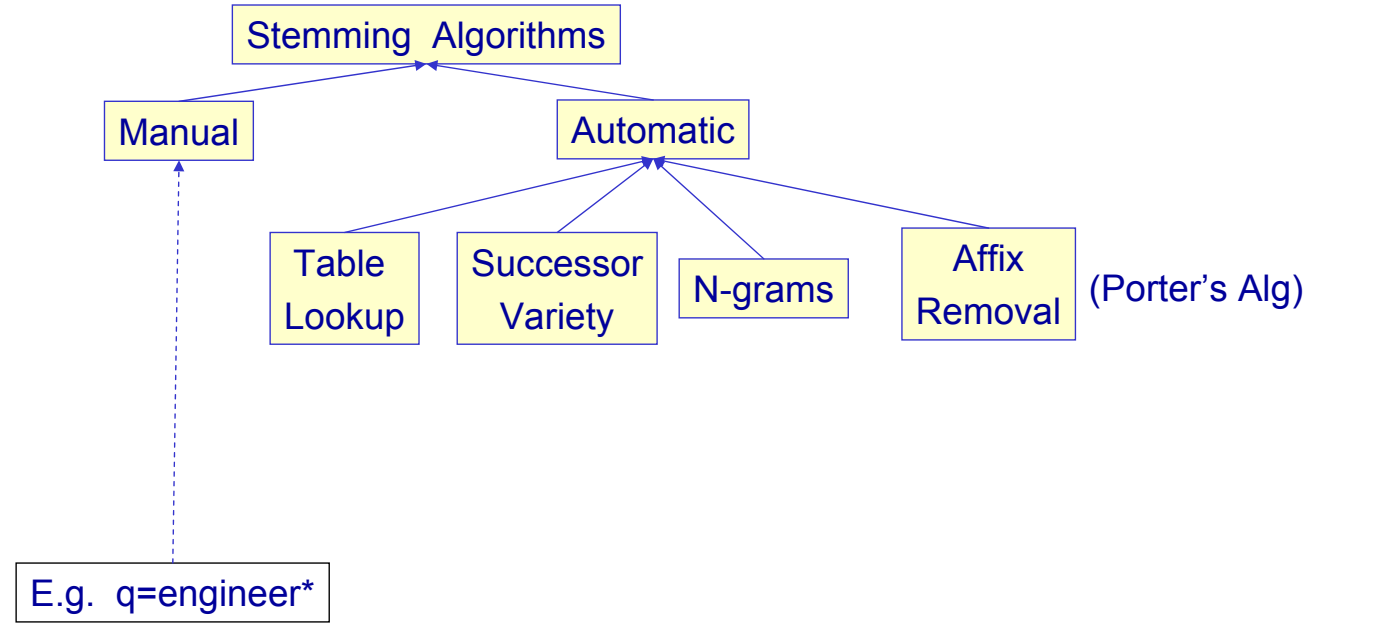
[γ] Αλγόριθμοι Στελέχωσης (Stemming Algorithms)



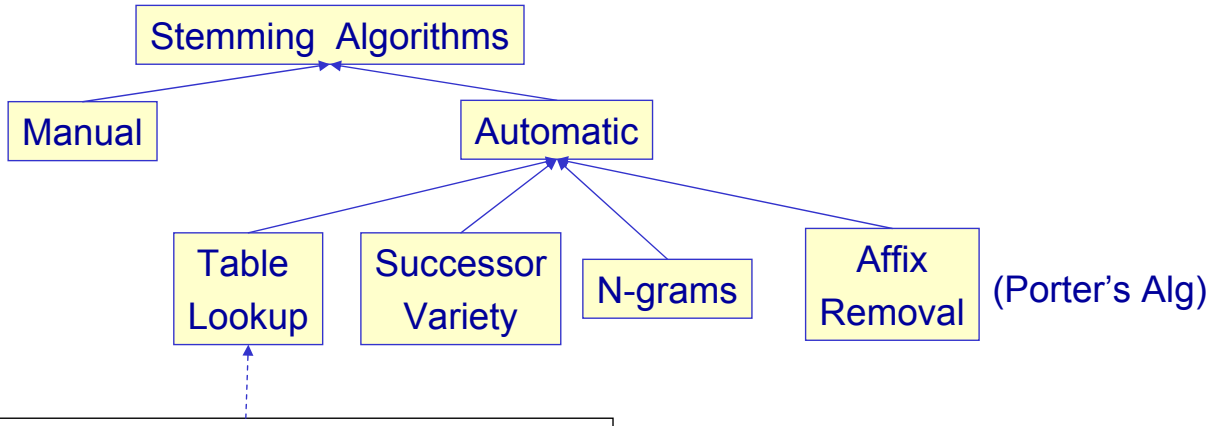
Πως αξιολογούμε έναν αλγόριθμο στελέχωσης;

- Ορθότητα (Correctness)
  - υπερστελέχωση (overstemming) έναντι υποστελέχωσης (understemming)
- Αποτελεσματικότητα ανάκτησης (Retrieval effectiveness)
- Δυνατότητα Συμπίεσης (Compression performance)

[γ] Αλγόριθμοι Στελέχωσης: Χειροκακτικός



[γ] Αλγόριθμοι Στελέχωσης:
Με Πίνακα



Terms and their corresponding stems are stored in a table (stemming dictionary) ,e.g.:

Term	**Stem**
engineering	engineer
engineered	engineer
engineer	engineer

(such tables are not easily available)

[γ] Αλγόριθμοι Στελέχωσης:
Successor Variety

Ιδέα: **Στελέχωση βάσει των συχνοτήτων των ακολουθιών γραμμάτων σε ένα σώμα κειμένου**

Βήματα για Στελέχωση Κειμένου
[1] Δημιουργία του πίνακα Ποικιλίας Διαδόχων (successor variety table)
[2] Χρήση του πίνακα για τεμαχισμό των λέξεων
[3] Επιλογή ενός τεμαχίου ως ρίζα (as stem)

[γ] Αλγόριθμοι Στελέχωσης:
Successor Variety (II)

Βήματα για Στελέχωση Κειμένου

[1] Δημιουργία του πίνακα Ποικιλίας Διαδόχων (successor variety table)

Παράδειγμα

- Έστω ότι θέλουμε να βρούμε τη ρίζα της λέξης READABLE
- Έστω το εξής σώμα κειμένου: ABLE, APE, BEATABLE, FIXABLE, READ, READABLE, READING, READS, RED, ROPE, RIPE
  - Πρόθεμα Αριθμός Επόμενων Γραμμάτων Επόμενα Γράμματα

Prefix	**Successor Variety**	**Letters**
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,I,S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	*BLANK*

[γ] Αλγόριθμοι Στελέχωσης:
Successor Variety (III)

Βήματα για Στελέχωση Κειμένου

[1] Δημιουργία του πίνακα Ποικιλίας Διαδόχων (successor variety table)

[2] Χρήση του πίνακα για τεμαχισμό των λέξεων

Πρόθεμα Αριθμός Επόμενων Γραμμάτων Επόμενα Γράμματα

Prefix	**Successor Variety**	**Letters**
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,I,S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	*BLANK*

Μπορεί ο πίνακας να μας βοηθήσει να τεμαχίσουμε "σωστά";

[γ] Αλγόριθμοι Στελέχωσης:
Successor Variety (IV)

Βήματα για Στελέχωση Κειμένου

[1] Δημιουργία του πίνακα Ποικιλίας Διαδόχων (successor variety table)

[2] Χρήση του πίνακα για τεμαχισμό των λέξεων

Πρόθεμα	Αριθμός Επόμενων Γραμμάτων	Επόμενα Γράμματα
Prefix	**Successor Variety**	**Letters**
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,I,S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	*BLANK*

[2] Τεμαχισμός βάσει της μεθόδου «peak & plateau»:

- τεμαχισμός στο γράμμα που οι διάδοχοί του είναι **περισσότεροι** των διαδόχων του προηγούμενου γράμματος
  - REA (1), READ (3)

Άρα READABLE => READ ABLE

[γ] Αλγόριθμοι Στελέχωσης:
Successor Variety (V)

Βήματα για Στελέχωση Κειμένου

[1] Δημιουργία του πίνακα Ποικιλίας Διαδόχων (successor variety table)

[2] Χρήση του πίνακα για τεμαχισμό των λέξεων

[3] Επιλογή ενός τεμαχίου ως ρίζα (as stem)

READABLE => READ ABLE

Ευρετικός κανόνας:

"*if (first segment occurs in <=12 words in the corpus) select first segment, else the second*"

Δικαιολόγηση: Αν εμφανίζεται πάνω από 12 φορές τότε μάλλον είναι πρόθεμα.

[γ] Αλγόριθμοι Στελέχωσης:
Successor Variety (VI)

Βήματα για Στελέχωση Κειμένου

[1] Δημιουργία του πίνακα Ποικιλίας Διαδόχων (successor variety table)

[2] Χρήση του πίνακα για τεμαχισμό των λέξεων

π.χ. READABLE => READ ABLE

[3] Επιλογή ενός τεμαχίου ως ρίζα (as stem)

π.χ. READABLE => READ ABLE

Παρατήρηση:

Η τεχνική αυτή δεν απαιτεί καμία είσοδο από το σχεδιαστή. Άρα μπορεί να εφαρμοστεί αυτούσια σε πολλές διαφορετικές γλώσσες.

[γ] Αλγόριθμοι Στελέχωσης:
n-grams

Ιδέα: Ομαδοποίησε λέξεις βάσει του αριθμού των κοινών διγραμμάτων ή ν-γραμμάτων

Πχ: σύγκριση "**statistics**" με "**statistical**"

- "statistics":
  - digrams: st ta at ti is st ti ic cs (9)
  - unique digrams: at cs ic is st ta ti (7)
- "statistical":
  - digrams: st ta at ti is st ti ic ca al (10)
  - unique digrams: al at ca ic is st ta ti (8)

Οι λέξεις "statistics" και "statistical" έχουν 6 κοινά διγράμματα (digrams).

[γ] Αλγόριθμοι Στελέχωσης:

n-grams (II)

Οι λέξεις “statistics” και “statistical” έχουν 6 κοινά διγράμματα (digrams).

Μπορούμε να μετρήσουμε τον βαθμό ομοιότητάς τους χρησιμοποιώντας μια μετρική, όπως:

- Μέγεθος τομής: $sim(X,Y) = |X \cap Y|$
- Dice similarity: $sim(X,Y) = 2\,|X \cap Y|/(|X|+|Y|)$
  - εδώ sim(statistics,statistical) = 2*6/(7+8)=0.8

Οι λέξεις της συλλογής ομαδοποιούνται με αυτόν τον τρόπο
(όλες οι λέξεις που έχουν την ίδια ρίζα καταχωρούνται στην ίδια ομάδα)

[γ] Αλγόριθμοι Στελέχωσης:

Affix Removal

Ιδέα: Απαλοιφή επιθεμάτων (suffixes) ή/και προθεμάτων (prefixes)

Porter’s Stemmer

- Simple procedure for removing known affixes in English without using a dictionary (i.e. without a lookup table).
- Can produce unusual stems that are not English words:
  - “computer”, “computational”, “computation” all reduced to same token “comput”
- May conflate (reduce to the same token) words that are actually distinct.
- Not recognize all morphological derivations.

[γ] Αλγόριθμοι Στελέχωσης:
Porter Stemmer

Παραδείγματα κανόνων:

- s → ∅ (for plural form)
- sses → ss (for plural form)

Εφαρμόζεται πρώτα η μακρύτερη ακολουθία

- e.g. stresses => stress,
- NOT stresses => stresse

RULES

suffix	replacement	example
1a		
sses	ss	caresses->caress
ies	i	ponies->poni, ties->ti
s	NUL	cats->cat
1b		
eed	ee	agreed->agree
ed	NUL	plastered->plaster
ing	NUL	motoring->motor
2		
ational	ate	relational->relate
tional	tion	conditional->condition
izer	ize	digitizer->digitize
ator	ate	operator->operate
….		

[γ] Αλγόριθμοι Στελέχωσης:
Porter Stemmer> Example

Original text:

Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales

- **After applying Porter's Stemmer (and eliminating stopwords):**

market strateg carr compan agricultur chemic report predict market share chemic report market statist agrochem pesticid herbicid fungicid insecticid fertil predict sale stimul demand price cut volum sale

[γ] Αλγόριθμοι Στελέχωσης:
Porter Stemmer >Errors

- Errors of "comission":
  - organization, organ → organ
  - police, policy → polic
  - arm, army → arm
- Errors of "omission":
  - cylinder, cylindrical
  - create, creation
  - Europe, European

[γ] Αλγόριθμοι Στελέχωσης:
Porter Stemmer > Code

- See [book MIR, Appendix]
- Demo available at:
  - http://snowball.tartarus.org/demo.php
- Implementation (C, Java, …) available at:
  - http://www.tartarus.org/~martin/PorterStemmer/

Αλγόριθμοι Στελέχωσης

Algorithm	*Language Independent*
Lookup table	NO
Successor Variety	YES
N-Grams	YES
Porter's Stemmer	NO

[δ] Επιλογή Λέξεων για την Ευρετηρίαση

[δ] Επιλογή Λέξεων για την Ευρετηρίαση

- Μια προσέγγιση είναι να θεωρήσουμε ως όρους ευρετηρίου ό,τι απέμεινε (αφαιρώντας λέξεις αποκλεισμού, κάνοντας στελέχωση)
- Μια άλλη προσέγγιση λέει ότι συνήθως τα ουσιαστικά είναι εκείνα που περιγράφουν κυρίως το νόημα μια πρότασης
  - Εκ τούτου θα μπορούσαμε να λάβουμε υπόψη (στην κατασκευή του ευρετηρίου) μόνο τα ουσιαστικά και άρα να παραλείψουμε τις αντωνυμίες, τα ρήματα και τα επίθετα.
  - Επίσης μπορούμε να θεωρήσουμε ομάδες ουσιαστικών που εμφανίζονται μαζί, π.χ. “computer science” , ως έναν όρο ευρετηρίου.
- Τέλος μια άλλη προσέγγιση είναι να καθορίσουμε το σύνολο των όρων ευρετηρίων από ελεγχόμενα λεξιλόγια (Θησαυρούς όρων)

(Β) Γλώσσες Επερώτησης για Ανάκτηση Πληροφοριών (Query Languages for IR)

Γλώσσες Επερώτησης για Ανάκτηση Πληροφοριών

- Επερωτήσεις λέξεων (Keyword-based Queries)
  - Μονολεκτικές επερωτήσεις (Single-word Queries)
  - Επερωτήσεις φυσικής γλώσσας (Natural Language Queries)
  - Boolean Επερωτήσεις (Boolean Queries)
  - Επερωτήσεις Συμφραζομένων (Context Queries)
    - Φραστικές Επερωτήσεις (Phrasal Queries)
    - Επερωτήσεις Εγγύτητας (Proximity Queries)
- Ταίριασμα Προτύπου (Pattern Matching)
  - Απλό (Simple)
  - Ανεκτικές σε ορθογραφικά λάθη (Allowing errors)
    - Levenstein distance, LCS longest common subsequence
  - Κανονικές Εκφράσεις (Regular expressions)
- Δομικές Επερωτήσεις (Structural Queries)
  - *(θα καλυφθούν σε επόμενο μάθημα)*
- Πρωτόκολλα επερώτησης (Query Protocols)

Εισαγωγή

- Ο τύπος των επερωτήσεων που επιτρέπονται σε ένα σύστημα εξαρτάται σε ένα βαθμό και από το Μοντέλο Ανάκτησης που χρησιμοποιεί το σύστημα

  - Boolean model => boolean queries
  - Extended Boolean model => boolean queries (…)
  - Vector Space model => natural language queries (free text)
  - Probabilistic model => natural language queries
  - ...

- Εδώ θα δούμε τύπους επερωτήσεων που μπορεί χρήσιμοι για την ανάκτηση πληροφοριών.
  - Αργότερα θα δούμε τις δομές δεδομένων και αλγόριθμους για την αποτίμησή τους.

Επερωτήσεις φυσικής γλώσσας
("Natural Language" Queries)

- Full text queries as arbitrary strings.
- Typically just treated as a **bag-of-words** for a vector-space model.
- Typically processed using standard vector-space retrieval methods.

Boolean Queries

- Keywords combined with Boolean operators:
  - OR: (e_1 OR e_2)
  - AND: (e_1 AND e_2)
  - BUT: (e_1 BUT e_2) Satisfy e_1 but **not** e_2
- Negation only allowed using BUT to allow efficient use of inverted index by filtering another efficiently retrievable set.
- Naïve users have trouble with Boolean logic.

Context-Queries

- Ability to search words in a given context, that is, near other words
- Types of Context Queries
  - **Phrasal Queries**
  - **Proximity Queries**

Phrasal Queries

- Retrieve documents with a specific phrase (**ordered** list of contiguous words)
  - "information theory"
  - "to be or not to be"
- May allow intervening stop words and/or stemming.
  - For example, "**buy camera**" matches:
    - "buy a camera",
    - "buy a camera", (two spaces)
    - "buying the cameras" etc.

Proximity Queries (Επερωτήσεις Εγγύτητας)

- List of words with specific maximal distance constraints between words.
- For example:
  - **"dogs" and "race" within 4 words**
- will match
  - "…dogs will begin the race…"
- May also perform stemming and/or not count stop words.
- The order may or may not be important

Pattern Matching

- Allow queries that match strings rather than word tokens.
- Requires more sophisticated data structures and algorithms than inverted indices to retrieve efficiently.

Some types of simple patterns:

- **Prefixes**: Pattern that matches start of word.
  - "anti" matches "antiquity", "antibody", etc.
- **Suffixes**: Pattern that matches end of word:
  - "ix" matches "fix", "matrix", etc.
- **Substrings**: Pattern that matches arbitrary subsequence of characters.
  - "rapt" matches "enrapture", "velociraptor" etc.
- **Ranges**: Pair of strings that matches any word lexicographically (alphabetically) between them.
  - "tin" to "tix" matches "tip", "tire", "title", etc.

More Complex Patterns: Allowing Errors

- What if query or document contains typos or misspellings?
- Judge similarity of words (or arbitrary strings) using:
  - **Edit distance (Levenstein distance)**
  - **Longest Common Subsequence (LCS)**
- Allow proximity search with bound on string similarity.

Edit (Levenstein) **Distance**

- Edit Distance = Minimum number of character *deletions*, *additions,* or *replacements* needed to make two strings equivalent.
  - "misspell" to "mispell" is distance 1
  - "misspell" to "mistell" is distance 2
  - "misspell" to "misspelling" is distance 3
- Can be computed efficiently using *dynamic programming*
  - O(*mn*) time where *m* and *n* are the lengths of the two strings being compared.

Longest Common Subsequence (LCS)

- Length of the longest subsequence of characters shared by two strings.
- A *subsequence* of a string is obtained by deleting zero or more characters.
- Examples:
  - "misspell" to "mispell" is 7
  - "misspelled" to "misinterpretted" is 7
    "mis…p…e…ed"

More complex patterns: Regular Expressions

- Language for composing complex patterns from simpler ones.
  - An individual character is a regex.
  - Union: If e_1 and e_2 are regexes, then (e_1 | e_2) is a regex that matches whatever either e_1 or e_2 matches.
  - Concatenation: If e_1 and e_2 are regexes, then e_1 e_2 is a regex that matches a string that consists of a substring that matches e_1 immediately followed by a substring that matches e_2
  - Repetition (Kleene closure): If e_1 is a regex, then e_1* is a regex that matches a sequence of zero or more strings that match e_1

Regular Expression Examples

- **(u|e)nabl(e|ing)** matches
  - unable
  - unabling
  - enable
  - enabling
- **(un|en)***able matches
  - able
  - unable
  - unenable
  - enununenable

Enhanced Regex's (Perl)

- Special terms for common sets of characters, such as alphabetic or numeric or general "wildcard".
- Special repetition operator (+) for 1 or more occurrences.
- Special optional operator (?) for 0 or 1 occurrences.
- Special repetition operator for specific range of number of occurrences: {min,max}.
  - A{1,5} One to five A's.
  - A{5,} Five or more A's
  - A{5} Exactly five A's

Perl Regex's

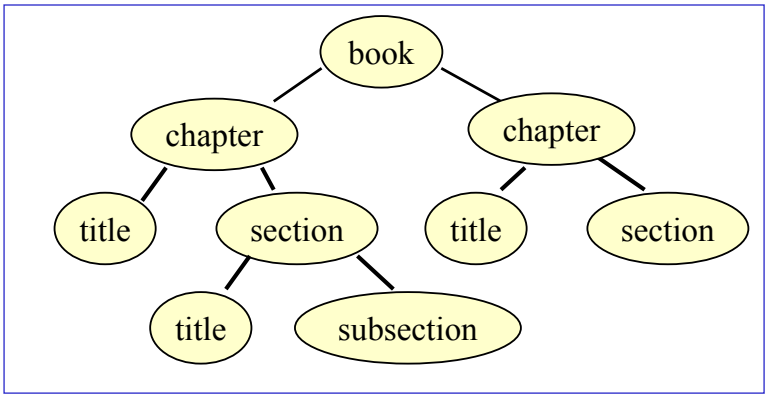
- Character classes:
  - \w (word char) Any alpha-numeric (not: \W)
  - \d (digit char) Any digit (not: \D)
  - \s (space char) Any whitespace (not: \S)
  - . (wildcard) Anything
- Anchor points:
  - \b (boundary) Word boundary
  - ^ Beginning of string
  - $ End of string
- Examples
  - U.S. phone number with optional area code:
    - /\b(\(\d{3}\)\s?)?\d{3}-\d{4}\b/
  - Email address:
    - /\b\S+@\S+(\.com|\.edu|\.gov|\.org|\.net)\b/

Note: Packages available to support Perl regex's in Java

Δομικές Επερωτήσεις (Structural Queries)

- Εδώ τα έγγραφα έχουν **δομή** που μπορεί να αξιοποιηθεί κατά την ανάκτηση

- Η δομή μπορεί να είναι:
  - Ένα προκαθορισμένο σύνολο πεδίων
    - title, author, abstract, etc.
  - Δομή Hypertext
  - Μια ιεραρχική δομή
    - Book, Chapter, Section, etc.



- ***Θα τις μελετήσουμε αναλυτικά σε μια άλλη διάλεξη***

Query Protocols

- They are not intended for final users.
- They are query languages that are used automatically by software applications to query text databases. Some of them are proposed as standard for querying CD-ROMs or as intermediate languages to query library systems
- Query Protocols
  - Z39.50
    - 1995 standard ANSI, NISO
    - bibliographical information
    - **SRW (Search and Retrieve Web Service): Extension of Z39.50 using Web Technologies. Queries in CQL**
  - WAIS (Wide Area Information Service)
    - used before the Web
  - Dienst Protocol
  - For CD-ROMS
    - CCL (Common Command Language)
      - 19 commands. Based on Z39.50
    - CD-RDx (Compact Disk Read only Data Exchange)
    - SFQL (Structured Full-text Query Language)

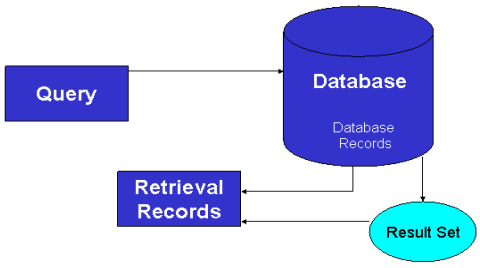
SFQL

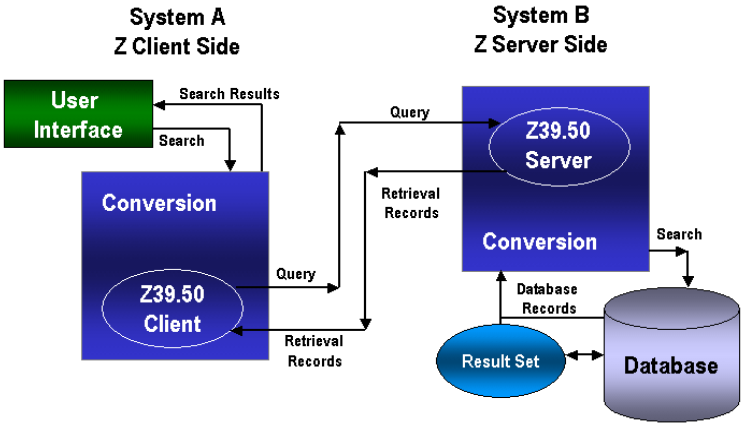
- **SFQL** (Structured Full-text Query Language)
  - Relational database query language SQL enhanced with “full text” search.
  - Παράδειγμα:

```
Select abstract
from journal.papers
where   author contains “Teller” and
        title contains “nuclear fusion” and
        date < 1/1/1950
```

- Supports Boolean operators, thesaurus, proximity operations, wild cards, repetitions.

Z39.50





CQL (Common Query Language)

- A formal language for representing queries to information retrieval systems
- Human-readable
- Search clause
  - Always includes a term
    - simple terms consist of one or more words
  - May include index name (i.e. field name)
    - To limit search to a particular field/element
    - Index name includes base name and may include prefix
      - title, subject
      - dc.title, dc.subject
    - Several index sets have been defined (called Context Sets in SRW)
      - dc
      - bath
      - srw
    - Context set defines the available indexes for a particular application

CQL (Common Query Language) (II)

- Relation
  - <, >, <=, >=, =, <>
  - exact used for string matching
  - all when term is list of words to indicate all words must be found
  - any when term is list of words to indicate any words must be found
- Boolean operators: and, or, not
- Proximity (prox operator)
  - relation (<, >, <=, >=, =, <>)
  - distance (integer)
  - unit (word, sentence, paragraph, element)
  - ordering (ordered or unordered)
- Masking rules and special characters
  - single asterisk (*) to mask zero or more characters
  - single question mark (?) to mask a single character
  - carat/hat (^) to indicate anchoring, left or right

CQL Examples

- **Simple queries:**
  - dinosaur
  - "the complete dinosaur"
- **Boolean**
  - dinosaur and bird or dinobird
  - "feathered dinosaur" and (yixian or jehol)
- **Proximity**
  - foo prox bar
  - foo prox/>/4/word/ordered bar
- **Indexes**
  - title = dinosaur
  - bath.title="the complete dinosaur"
  - srw.serverChoice=dinosaur
- **Relations**
  - year > 1998
  - title all "complete dinosaur"
  - title any "dinosaur bird reptile"
  - title exact "the complete dinosaur"