ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Αποθήκες Δεδομένων και Εξόρυξη Γνώσης από Δεδομένα

Ανάκτηση Πληροφορίας

Μανώλης Μαραγκουδάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ
INFORMATION RETRIEVAL (IR)

Μανώλης Μαραγκουδάκης

Εισαγωγικά

- ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ: αναπαράσταση, αποθήκευση, οργάνωση και προσπέλαση σε αντικείμενα πληροφορίας
- Επίκεντρο η πληροφοριακή ανάγκη του χρήστη

- Πληροφοριακή ανάγκη του χρήστη:
- Εντόπισε όλα τα κείμενα με πληροφορίες σχετικά με φοιτητές που
  - (1) φοιτούν σε κάποια σχολή πληροφορικής,
  - (2) συμμετέχουν σε κάποιο αθλητικό σύλλογο
- Η ανάγκη του χρήστη μετατρέπεται σε μια ερώτηση (query)

Εισαγωγικά

Η Ανάκτηση Πληροφορίας μελετά προβλήματα που σχετίζονται με

- Αναπαράσταση,
- Αποθήκευση,
- Οργάνωση, και
- Προσπέλαση

σε αντικείμενα πληροφορίας

- Κείμενα, εικόνες, ήχοι, κτλ..

Να βρεθούν όλα τα ξενοδοχεία της Ελλάδας στα οποία η τιμή του δίκλινου δωματίου είναι μικρότερη από 100 € τη βραδιά.

(*σαφές ερώτημα*)

Να βρεθούν κείμενα τα οποία αναφέρονται στον οικολογικό οργανισμό WWF.

(*ασαφές ερώτημα*)

Ανάκτηση Πληροφορίας VS. Ανάκτηση Δεδομένων

Ανάκτηση πληροφορίας

- Η ερώτηση είναι ασαφής
- Η σημασιολογία είναι συχνά ελλιπής
- Μερικά λάθη είναι ανεκτά

Ανάκτηση δεδομένων

- Καλά ορισμένη ερώτηση
- Βρίσκονται αντικείμενα που ταιριάζουν απόλυτα με την ερώτηση
- Ένα μόνο λάθος συνιστά καθολική αποτυχία

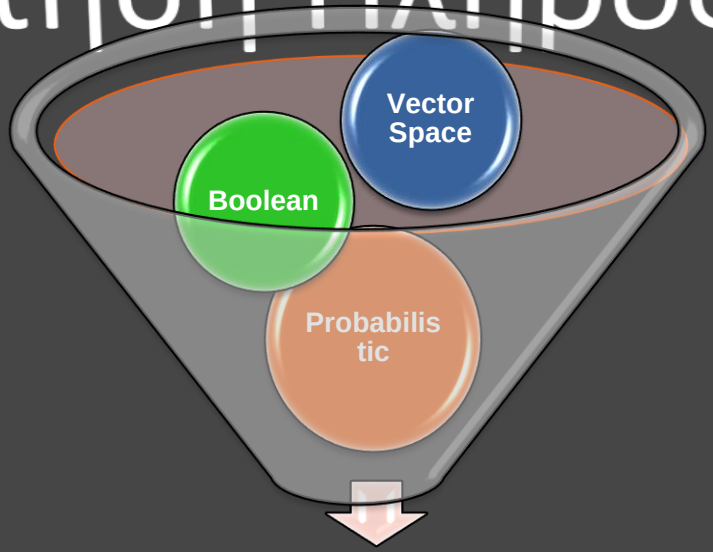
Σύγκριση

	ΑΝΑΚΤΗΣΗ ΔΕΔΟΜΕΝΩΝ	ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ
Ταίριασμα (Matching)	Ακριβής (Exact)	Μερική (partial) Καλύτερη (best)
Ερωτήσεις (Queries)	Σαφείς (Precise)	Ασαφείς (Imprecise)
Πληροφορία (Information)	Δεδομένα, Αριθμητικές τιμές	Φυσική Γλώσσα (Natural Language)

Συστήματα Ανάκτησης Πληροφορίας

- Προσπαθούν να βρουν όλα τα αντικείμενα πληροφορίας που σχετίζονται με την ερώτηση του χρήστη
- Προσπαθούν να μην ανακτήσουν κανένα αντικείμενο πληροφορίας που δε σχετίζεται με την ερώτηση του χρήστη
- Τα αποτελέσματα ταξινομούνται ανάλογα με το ποσοστό συσχέτισης (relevance)
  - *Έννοια σχετικότητας πιο σημαντική από ακριβές ταίριασμα*

Ανάκτηση Πληροφορίας



Μοντελοποίηση

Ενότητα : Μοντελοποίηση

Τυπικός Ορισμός Μοντέλου IR

D (documents)

- σύνολο λογικών όψεων (αναπαραστάσεων) κειμένων

Q (queries)

- σύνολο λογικών όψεων ερωτημάτων

F (framework)

- πλαίσιο μοντελοποίησης κειμένων, ερωτημάτων και συσχετισμών τους

R(q,d) (ranking function)

- συνάρτηση βαθμολόγησης
- Αντιστοιχίζει ένα πραγματικό αριθμό με ένα ερώτημα και ένα κείμενο

Λέξεις κλειδιά (ή όροι δεικτοδότησης-keywords)

- Χρησιμοποιούνται για την αναπαράσταση των κειμένων
- Πρέπει να είναι αντιπροσωπευτικοί για τη σημασιολογία του κειμένου
- Περίληψη των περιεχομένων του κειμένου
- Κυρίως, είναι ουσιαστικά
  - Επίθετα, επιρρήματα κτλ. είναι λιγότερο χρήσιμα

Κείμενο 1

- … η γεωργική επανάσταση

Κείμενο 2

- … η βιομηχανική επανάσταση

Κείμενο 3

- … η οκτωβριανή επανάσταση

Η επιλογή της λέξης **επανάσταση** για τα 3 κείμενα δημιουργεί πρόβλημα. **Γιατί;**

Βάρος λέξης-κλειδιού

- Όλες οι λέξεις κλειδιά δεν έχουν την ίδια βαρύτητα για τις προτιμήσεις των χρηστών.
- Κάποιες λέξεις μπορεί να είναι σημαντικές ενώ κάποιες άλλες λιγότερο σημαντικές.

- Έστω k_i μία λέξη κλειδί και d_j ένα κείμενο.
  - Το βάρος ορίζεται ως $w(k_i,d_j) >= 0$ και δηλώνει το πόσο σημαντική είναι η λέξη κλειδί σε σχέση με το κείμενο.

Ορισμός

- Έστω *t* αριθμός των keywords και $K=\{k_1,\ldots,k_t\}$ το σύνολο των keywords. Εάν το keyword k_i δεν εμφανίζεται στο κείμενο *dj* τότε $w(k_i,d_j)=0$. Διαφορετικά, $w(k_i,d_j) > 0$.

- Άρα σε κάθε κείμενο d_j αντιστοιχεί ένα διάνυσμα βαρών $(w_{1,j}, w_{2,j}, \ldots, w_{t,j})$.

Το Διανυσματικό Μοντέλο (Vector Space)

- Η χρήση μόνο δυαδικών βαρών είναι περιοριστική
- Τα βάρη των όρων μπορούν έτσι να χρησιμοποιηθούν για να εκφράσουν ένα *βαθμό ομοιότητας ανάμεσα από κάθε ερώτημα και* κάθε κείμενο

- Τα μη δυαδικά βάρη επιτρέπουν και την αντιμετώπιση ερωτήσεων μερικού ταιριάσματος
- Τα ζυγισμένα σύνολα κειμένων επιτρέπουν καλύτερο/πιο ποιοτικό ταίριασμα

Το Διανυσματικό Μοντέλο

- *t διακριτοί όροι απομένουν μετά την* προ-επεξεργασία των κειμένων
  - Μοναδικοί όροι που σχηματίζουν το ΛΕΞΙΛΟΓΙΟ (vocabulary)
- Αυτοί οι «ορθοκανονικοί» όροι σχηματίζουν ένα χώρο διανυσμάτων
- Διάσταση = t = |*λεξιλόγιο*|
- Σε κάθε όρος *i σε ένα κείμενο ή ερώτηση j* αποδίδεται ένα βάρος w_{ij}.

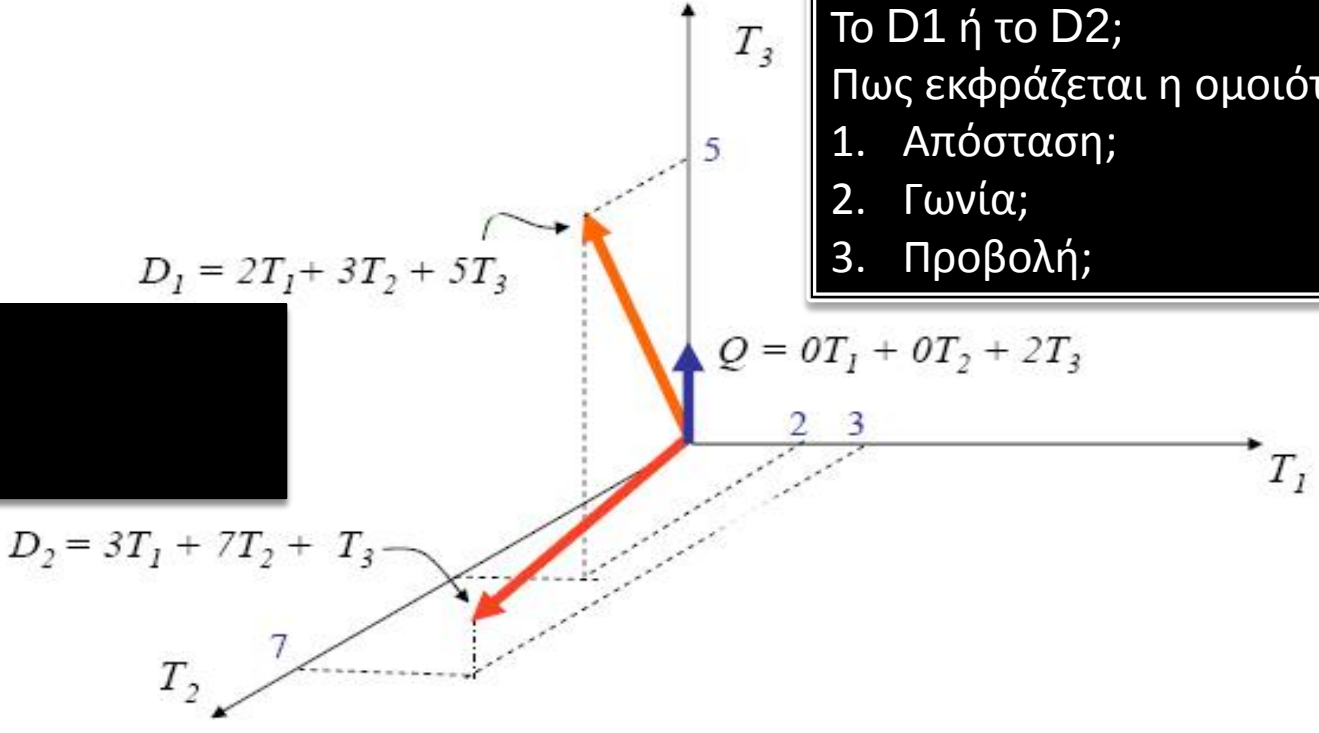
- Και τα κείμενα και οι ερωτήσεις εκφράζονται ως *t-διάστατα διανύσματα:*
  - $dj = (w_{1j}, w_{2j}, \ldots, w_{tj})$
- Θεωρούμε την ερώτηση ως ένα σύντομο κείμενο
  - Επιστρέφουμε τα κείμενα σε κατάταξη ανάλογα με το πόσο κοντά βρίσκονται στην ερώτηση
- Το διανυσματικό μοντέλο αναπτύχθηκε στα πλαίσια του συστήματος SMART (Salton, c. 1970)
- Χρησιμοποιείται ευρέως
  - Από μηχανές αναζήτησης στο Web.

Αναπαράσταση στο χώρο

D=κείμενα
T= λέξη κλειδί
Q=ερώτημα

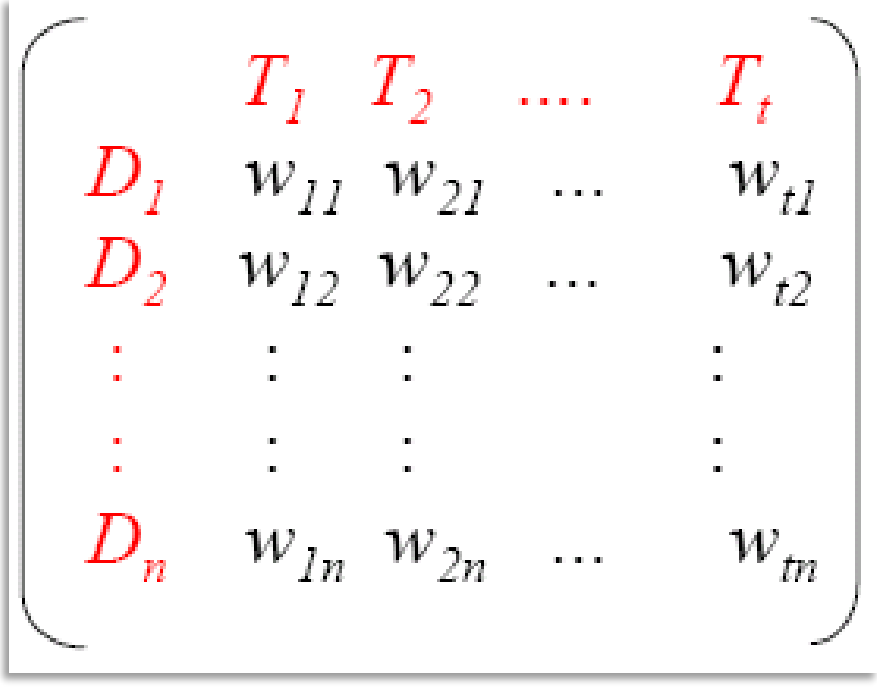
Ποιο είναι πιο όμοιο με το Q;
Το D1 ή το D2;
Πως εκφράζεται η ομοιότητα;
1. Απόσταση;
2. Γωνία;
3. Προβολή;

D1=2T1+3T2+5T3
D2=3T1+7T2+T3
Q=0T1+0T2+2T3



Αναπαράσταση Συλλογής Κειμένων

- Μία συλλογή από *n κείμενα μπορεί να αναπαρασταθεί* ως ένας πίνακας όρων-κειμένων *nxt (term-document* matrix)
- Κάθε στοιχείο του πίνακα αντιστοιχεί στο βάρος ενός όρου σε ένα κείμενο. Μηδέν βάρος σημαίνει ότι ο όρος δεν εμφανίζεται στο κείμενο



Βάρη Όρων: Συχνότητα Όρων

- Όσο πιο συχνός είναι ένας όρος σε ένα κείμενο τόσο πιο σημαντικός είναι
- f_{ij} = *συχνότητα του όρου i στο κείμενο j*
- Η κανονικοποίηση της συχνότητας όρων (*term frequency, tf) μπορεί να γίνει ως εξής:*
  - $tf_{ij} = f_{ij} / max\{f_{ij}\}$

Βάρη Όρων: Αντίστροφη Συχνότητα Κειμένου

- Οι όροι που εμφανίζονται σε πολλά διαφορετικά κείμενα είναι λιγότερο ενδεικτικοί του θέματος
  - df_i = *συχνότητα κειμένου του όρου* i = αριθμός κειμένων που περιλαμβάνουν τον όρο i
  - idf_i = *αντίστροφη συχνότητα κειμένου του όρου* i = $\log_2 (N/ df_i)$
    - (*N: συνολικός αριθμός κειμένων)*

- Αντίστροφη συχνότητα κειμένου (Inverted document frequency)
  - Παρέχει μία ένδειξη της διακριτικής ικανότητας ενός όρου
  - Ο λογάριθμος χρησιμοποιείται για να μετριαστεί η επίδραση στην συχνότητα όρου *tf*

Σχήμα TF-IDF

- Το σχήμα στάθμισης όρων *tf-idf έχει ως εξής:*

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2(N/df_i)$$

- Αποδίδεται μεγαλύτερο βάρος σε έναν όρο που εμφανίζεται συχνά σε ένα κείμενο αλλά σπάνια στα υπόλοιπα κείμενα της συλλογής
- Πειραματικά, το σχήμα tf-idf έχει βρεθεί ότι δουλεύει καλά
- Υπάρχει και θεωρητική απόδειξη (Papineni, 2001).

Παράδειγμα

- Ένα κείμενο περιέχει τους ακόλουθους όρους με συχνότητα:

  A(3), B(2), C(1)

- Υποθέτουμε ότι η συλλογή περιλαμβάνει 10.000 κείμενα και ότι οι συχνότητες κειμένων αυτών των όρων είναι:

  A(50), B(1300), C(250)

- Τότε:

  A: tf = 3/3, idf = log(10000/50) = 5.3, tf-idf = 5.3

  B: tf = 2/3, idf = log(10000/1300) = 2.0, tf-idf = 1.3

  C: tf = 1/3, idf = log(10000/250) = 3.7, tf-idf = 1.2

Διάνυσμα Ερώτησης-Μέτρο Ομοιότητας (Similarity Measure)

- Το διάνυσμα της ερώτησης εκφράζεται τυπικά ως ένα κείμενο σταθμισμένο κατά tf-idf
- Εναλλακτικά, ο χρήστης μπορεί να παρέχει βάρη στους όρους της ερώτησης
  - Δίχως βάρη:
    - Q = < database; text; information >
  - Με χρήση βαρών:
    - Q = < database 0.5; text 0.8; information 0.2 >

- Πώς υπολογίζουμε την ομοιότητα μεταξύ των διανυσμάτων των κειμένων της συλλογής και των διανυσμάτων της ερώτησης;
- Ένα μέτρο ομοιότητας είναι μία συνάρτηση που υπολογίζει τον βαθμό ομοιότητας μεταξύ δυο διανυσμάτων
- Με βάση ενός μέτρου ομοιότητας μπορούμε να:
  - Κατατάξουμε τα ανακτημένα κείμενα ανάλογα με την (υποτιθέμενη) σχετικότητα
  - Καθορίσουμε ένα συγκεκριμένο κατώφλι ώστε να ελεγχθεί το πλήθος των ανακτημένων κειμένων

Επιθυμητές Ιδιότητες Ομοιότητας

- Αν το d1 είναι κοντά στο d2, τότε το d2 είναι κοντά στο d1
- Αν το d1 είναι κοντά στο d2, και το d2 είναι κοντά στο d3, τότε το d1 δεν είναι μακριά από το d3
- Κανένα κείμενο δεν είναι πιο κοντά στο d από το ίδιο το d
  - Μερικές φορές είναι καλή ιδέα να καθορίσουμε ως μέγιστη δυνατή ομοιότητα την «απόσταση» μεταξύ ενός κειμένου d και του εαυτού του

- Ευκλείδεια απόσταση:
  - Η απόσταση μεταξύ των διανυσμάτων *d1 και d2 είναι το μήκος του διανύσματος:* $|d1 - d2|$
- Δεν έχουμε ασχοληθεί ακόμα με το θέμα της κανονικοποίησης ως προς το μήκος κειμένου
- Μεγάλα κείμενα θα ήταν πιο όμοια μεταξύ τους λόγω μήκους και όχι λόγω θέματος

Εσωτερικό Γινόμενο (Inner Product)

- Η ομοιότητα μεταξύ των διανυσμάτων ενός κειμένου d_i ***και μιας ερώτησης q μπορεί να*** υπολογιστεί ως το εσωτερικό τους γινόμενο:

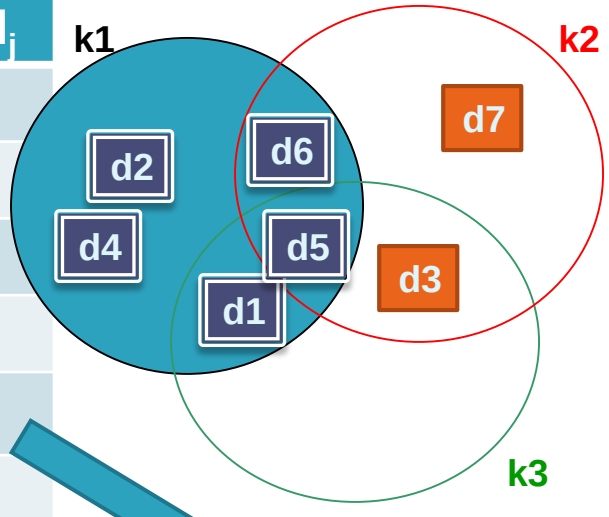
$$\text{sim}(d_j, q) = d_j \bullet q = \sum_{i=1}^{t} w_{ij} \cdot w_{iq}$$

- Όπου
  - *w_{ij} είναι το βάρος του όρου i στο κείμενο j και w_{iq} είναι το βάρος του όρου i στην ερώτηση*
- Για δυαδικά διανύσματα, το εσωτερικό γινόμενο είναι ο αριθμός των όρων της ερώτησης που ταιριάζουν με το κείμενο
- Για σταθμισμένα διανύσματα όρων, είναι το άθροισμα των γινομένων των βαρών των όρων που ταιριάζουν

Ιδιότητες Εσωτερικού Γινομένου

- Λειτουργεί υπέρ μεγάλων κειμένων με μεγάλο αριθμό μοναδικών όρων
  - Ξανά, το θέμα της κανονικοποίησης
- Μετρά πόσοι όροι ταιριάζουν αλλά όχι και πόσοι όροι δεν ταιριάζουν
- Παράδειγμα:

	k1	k2	k3	$q \bullet d_j$
d1	1	0	1	2
d2	1	0	0	1
d3	0	1	1	2
d4	1	0	0	1
d5	1	1	1	3
d6	1	1	0	2
d7	0	1	0	1
q	1	1	1	

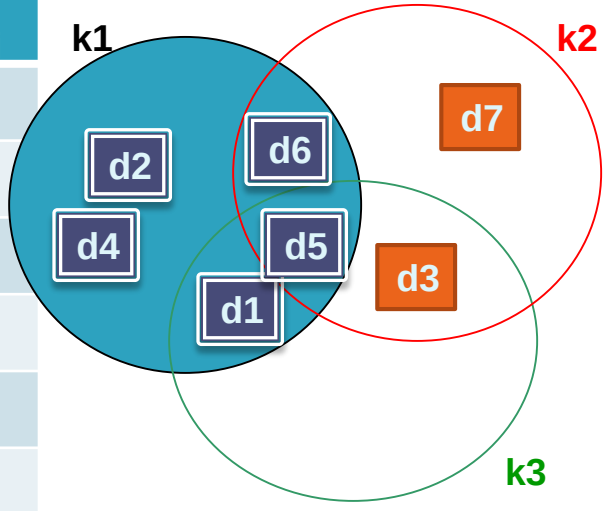


Το κείμενο d5 είναι όντως αυτό που περιέχει και τα 3 keywords k1, k2, k3.

Άσκηση

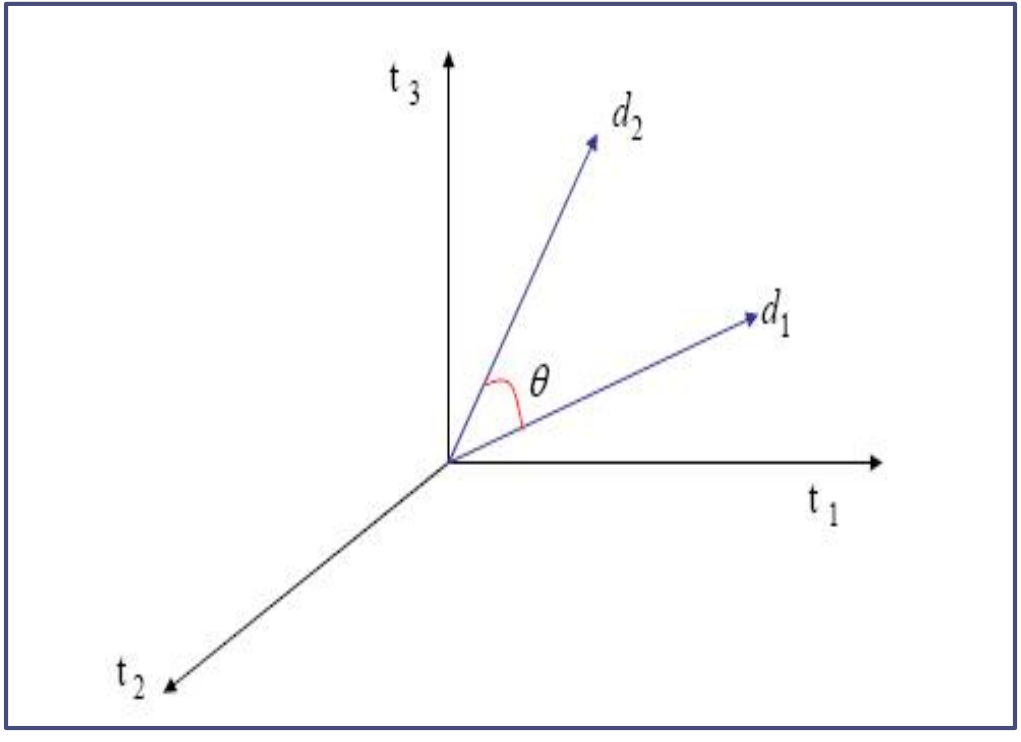
- Με βάση το εσωτερικό γινόμενο, ποιο κείμενο είναι πιο κοντά στην ερώτηση q;

	k1	k2	k3	$q \cdot d_j$
d1	1	0	1	?
d2	1	0	0	?
d3	0	1	1	?
d4	1	0	0	?
d5	1	1	1	?
d6	1	1	0	?
d7	0	1	0	?
q	1	2	3	



Ομοιότητα Συνημίτονου (Cosine Similarity)

- Η απόσταση μεταξύ των διανυσμάτων *d1 και d2* μπορεί να εκφραστεί από το συνημίτονο της γωνίας *ϑ ανάμεσά τους*
- Προσοχή: πρόκειται για ομοιότητα όχι για απόσταση



Ομοιότητα Συνημίτονου

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

- Ο παρονομαστής εμπλέκει τα μήκη των διανυσμάτων

$$\text{Length} \left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,j}^2}$$

- Το μέτρο συνημίτονου είναι επίσης γνωστό ως *κανονικοποιημένο εσωτερικό γινόμενο*
- *Για κανονικοποιημένα διανύσματα ισχύει ο τύπος του εσωτερικού γινόμενου*

$$\cos(\vec{d}_j, \vec{d}_k) = \vec{d}_j \cdot \vec{d}_k$$

Jaccard Similarity και Dice Similarity

•**Jaccard**: $SC(Q,D_i)=\frac{\sum_{j=1}^{t} w_{qj} d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 + \sum_{j=1}^{t} (w_{qj})^2 - \sum_{j=1}^{t} w_{qj} d_{ij}}$

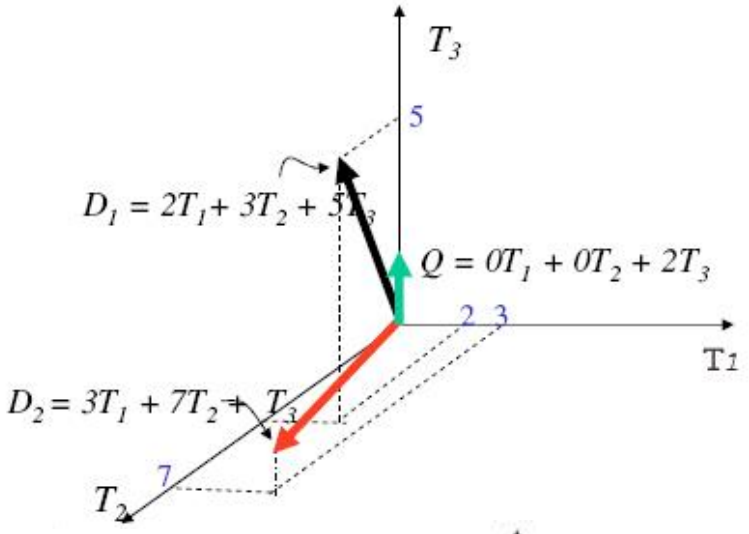
•**Dice**: $SC(Q,D_i)=\frac{2\sum_{j=1}^{t} w_{qj} d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}$

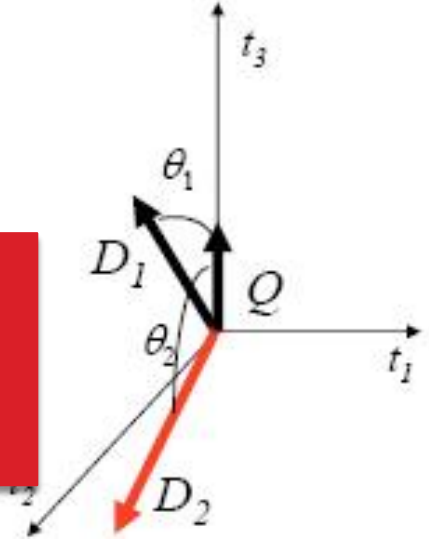
Άσκηση

- *Να κατατάξετε τα ακόλουθα σε φθίνουσα σειρά ομοιότητας συνημίτονου:*
  - Δύο κείμενα που έχουν κοινές μόνο συχνές λέξεις (π.χ. *το, η, οι, ο, και)*
  - Δύο κείμενα που δεν έχουν κοινές λέξεις
  - Δύο κείμενα που έχουν πολλές σπάνιες λέξεις κοινές (π.χ. *αστροφυσική, διαστημόπλοιο)*

Σύγκριση Εσωτερικού Γινομένου-Συνημίτονου

- D1=2T1+3T2+5T3
- D2=3T1+7T2+T3
- Q=0T1+0T2+2T3
  - Sim(D1,Q)=2*0+3*0+5*2=10
  - Sim(D2,Q)=3*0+7*0+1*2=2
  - CosSim(D1,Q)=(2+3+5)/√(4+9+25)*(0+0+4)=0.81
  - CosSim(D2,Q)=(3+7+1)/√(9+49+1)*(0+0+4)=0.13





Με εσωτερικό γινόμενο:
Το D1 είναι πιο κοντά προς την ερώτηση Q από το D2: 5 φορές
Με συνημίτονο:
Το D1 είναι πιο κοντά προς την ερώτηση Q από το D2: 6.23 φορές!

Σύνοψη Διανυσματικού Μοντέλου

Πλεονεκτήματα

- Απλή, αλγεβρική προσέγγιση
- Βασίζεται και σε τοπικές (*tf) και σε καθολικές* (*idf) συχνότητες εμφάνισης λέξεων*
- Παρέχει δυνατότητα μερικού ταιριάσματος και αποτελέσματα σε κατάταξη ως προς τη σχετικότητά τους
- Τείνει να δουλεύει πολύ καλά στην πράξη
- Επιτρέπει αποδοτική υλοποίηση για μεγάλες συλλογές κειμένων

Μειονεκτήματα

- Χάνεται σημασιολογική πληροφορία (π.χ. έννοιες λέξεων)
- Χάνεται συντακτική πληροφορία (π.χ. δομή φράσεων, σειρά λέξεων, εγγύτητα λέξεων)
- Υπόθεση ανεξαρτησίας όρων (π.χ. αγνοεί συνώνυμους όρους)
- Έλλειψη ελέγχου που παρέχει το μοντέλο Boolean (π.χ. που απαιτεί να εμφανίζεται ένας όρος σε ένα κείμενο)
  - Με βάση μία ερώτηση δύο όρων “A B”, μπορεί να προτιμηθεί ένα κείμενο που περιέχει το A πολλές φορές και καθόλου το B από ένα κείμενο που περιλαμβάνει και τους δύο όρους

Περίληψη Διανυσματικού Μοντέλου

- K={k1,…,kt} το σύνολο των όρων της ευρετηρίασης
- Κάθε έγγραφο dj αναπαρίσταται σαν ένα διάνυσμα $d_j = (w_{1j}, ..., w_{tj})$

όπου

$$w_{ij} = tf_{ij} idf_i = tf_{ij} \log_2(N / df_i)$$

- Μια ερώτηση q είναι επίσης ένα διάνυσμα $q = (w_{1q}, ..., w_{tq})$

όπου $w_{iq} = tf_{iq} idf_i = tf_{iq} \log_2(N / df_i)$

- Υπολογισμός συνημίτονου ομοιότητας

$$CosSim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \sum_{i=1}^{t} w_{iq}^2}}$$

Απλοϊκή Υλοποίηση

1. Φτιάξε το tf-idf διάνυσμα για κάθε έγγραφο dj της συλλογής (έστω V το λεξιλόγιο)
2. Φτιάξε το td-idf διάνυσμα q της επερώτησης
3. Για κάθε έγγραφο dj της συλλογής
   a. Υπολόγισε το σκορ cosSim(dj,q)
4. Κατάταξε τα έγγραφα σε φθίνουσα σειρά
5. Παρουσίασε τα έγγραφα στο χρήστη

Χρονική πολυπλοκότητα

$$O(|D||V|)$$

Κακό για μεγάλο V & D !

- Π.χ.

|V|=10.000

|D|=100.000

|V|*|D|=1.000.000!

Ευφυέστερη Υλοποίηση

- Ένας όρος που δεν εμφανίζεται και στην ερώτηση και στο έγγραφο δεν επηρεάζει το βαθμό ομοιότητας συνημίτονου
  - Το γινόμενο των βαρών είναι 0→δεν συνεισφέρει στο εσωτερικό γινόμενο
- Συνήθως ή ερώτηση είναι μικρή, άρα το διάνυσμα της είναι εξαιρετικά αραιό

- Μπορούμε να χρησιμοποιήσουμε ένα ευρετήριο ώστε να υπολογίσουμε το βαθμό ομοιότητας μόνο εκείνων των εγγράφων που περιέχουν τουλάχιστον έναν όρο της ερώτησης

Νέο

3. Για κάθε έγγραφο dj της συλλογής
   a. Υπολόγισε το σκορ

3. Για κάθε έγγραφο dj που περιέχει τουλάχιστον έναν όρο της ερώτησης
   a. Υπολόγισε το σκορ cosSim(dj,q)

Ανάκτηση Πληροφορίας

Ενότητα : Αξιολόγηση Ανάκτησης

Ρόλος της Αξιολόγησης

- Επίδοση συστήματος
  - Χρόνος απόκρισης, αποθηκευτικός χώρος, συμπίεση δεδομένων
- Κόστος
  - Κόστος ανάπτυξης
    - Σχεδιασμού, δοκιμών, αξιολόγησης
  - Λειτουργικά έξοδα
    - Εξοπλισμού, προσωπικό, κτλ

- Αξιολόγηση αποτελεσματικότητας
  - Ποιος είναι ο καλύτερος τρόπος για:
    - Ανάκτηση
    - Κατάταξη
    - Προσδιορισμό βαρών
  - Πόσα έγγραφα από την απάντηση ενός συστήματος θα πρέπει να εξετάσει ο χρήστης για να βρει μερικά/όλα τα συναφή;

Αξιολόγηση μέσω χειρωνακτικά επισημειωμένων συλλογών

- Βήματα
  1. Ξεκίνα από μια συλλογή εγγράφων
  2. Συνέλεξε ένα σύνολο ερωτημάτων για τη συλλογή αυτή
  3. Βρες έναν ή περισσότερους ειδήμονες για να επισημειώσουν τα συναφή έγγραφα για κάθε ερώτημα
  4. Χρησιμοποίησε την κρίση τους για την αξιολόγηση συστημάτων

Μειονέκτημα

- Απαιτεί μεγάλη και χρονοβόρα προσπάθεια σε μεγάλες συλλογές

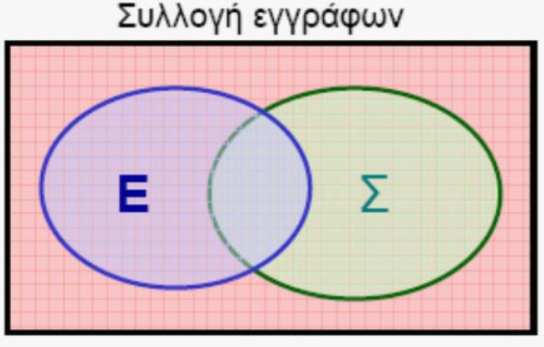
Μέτρα Αξιολόγησης

- Ακρίβεια **(Precision)**:
  - Διαισθητικά: Η ικανότητα ανάκτησης μόνο συναφών εγγράφων

- Ανάκληση **(Recall):**
  - Διαισθητικά: Η ικανότητα εύρεσης όλων των συναφών εγγράφων της συλλογής

Ακρίβεια και Ανάκληση

Έστω ένα ερώτημα q



Ε: Ευρεθέντα (από το ΣΑΠ)

Σ: Συναφή (με το ερώτημα q)

$$\text{Ακρίβεια} = \frac{|E \cap \Sigma|}{|E|}$$

P(recision)

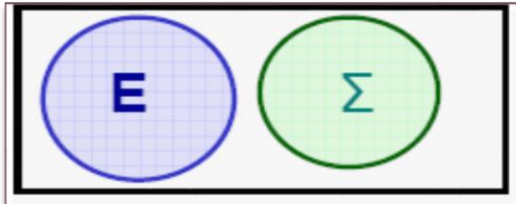
$$\text{Ανάκληση} = \frac{|E \cap \Sigma|}{|\Sigma|}$$

R(ecall)

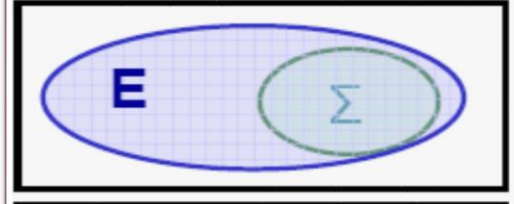
$$\text{ανάκληση} = \frac{\text{Αριθμός σχετικών κειμένων που ανακτήθηκαν}}{\text{Συνολικός αριθμός σχετικών κειμένων}}$$

$$\text{ακρίβεια} = \frac{\text{Αριθμός σχετικών κειμένων που ανακτήθηκαν}}{\text{Συνολικός αριθμός κειμένων που ανακτήθηκαν}}$$

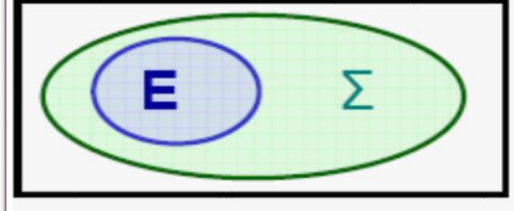
Περιπτώσεις



P=0, R=0 (χειρότερη περίπτωση)



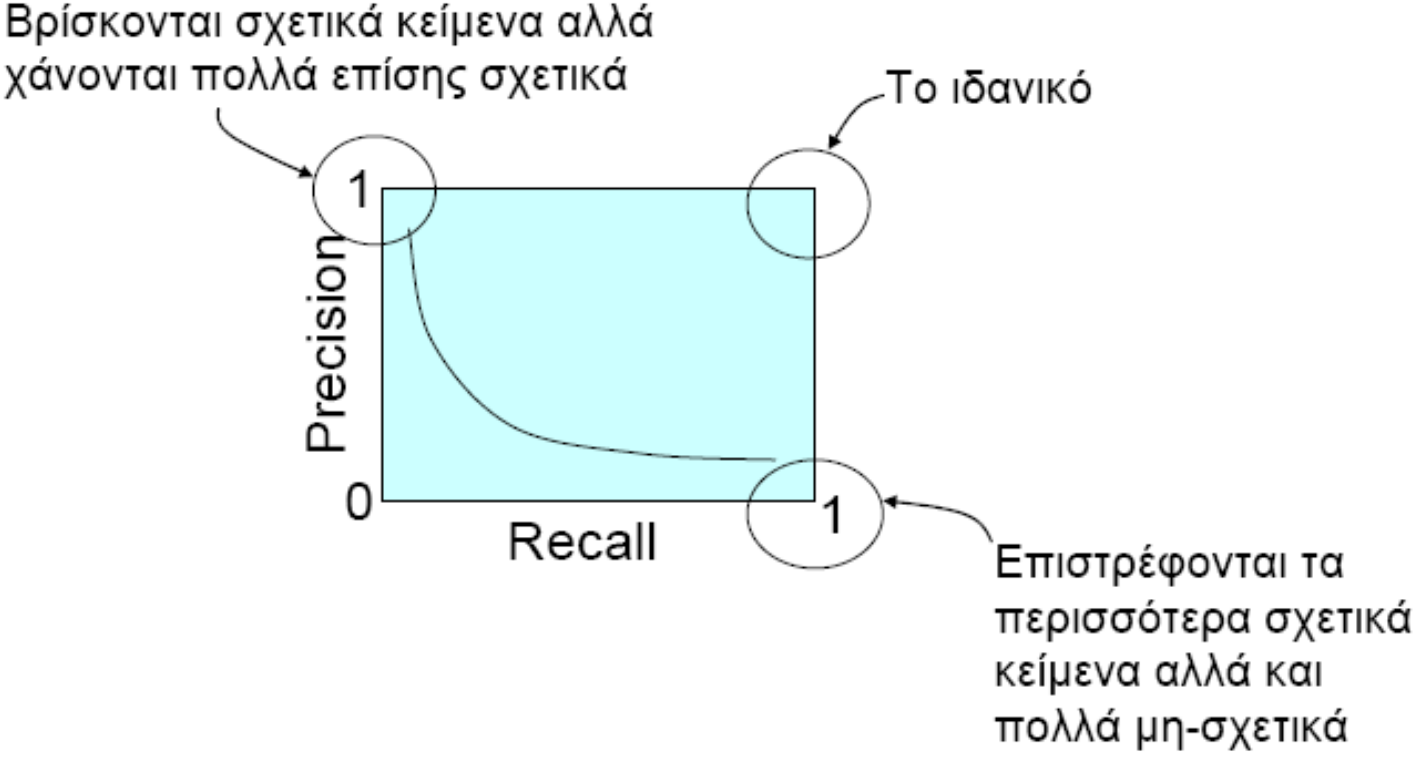
P=low, R=1 (η επίτευξη R=1 είναι ευκολότατη)



P=1, R:low

P=1, R:1 (ιδανική περίπτωση)

Ακρίβεια vs. Ανάκλησης



Κίνητρο για την ύπαρξη καμπυλών και σημείων ακρίβειας-ανάκλησης (Precision-Recall)

- Ο χρήστης δεν **καταναλώνει** την απάντηση μονομιάς
  - Αρχίζει από την κορυφή της λίστας των αποτελεσμάτων
- Αυτό δεν λαμβάνεται υπόψη από την ακρίβεια και την ανάκληση!!!

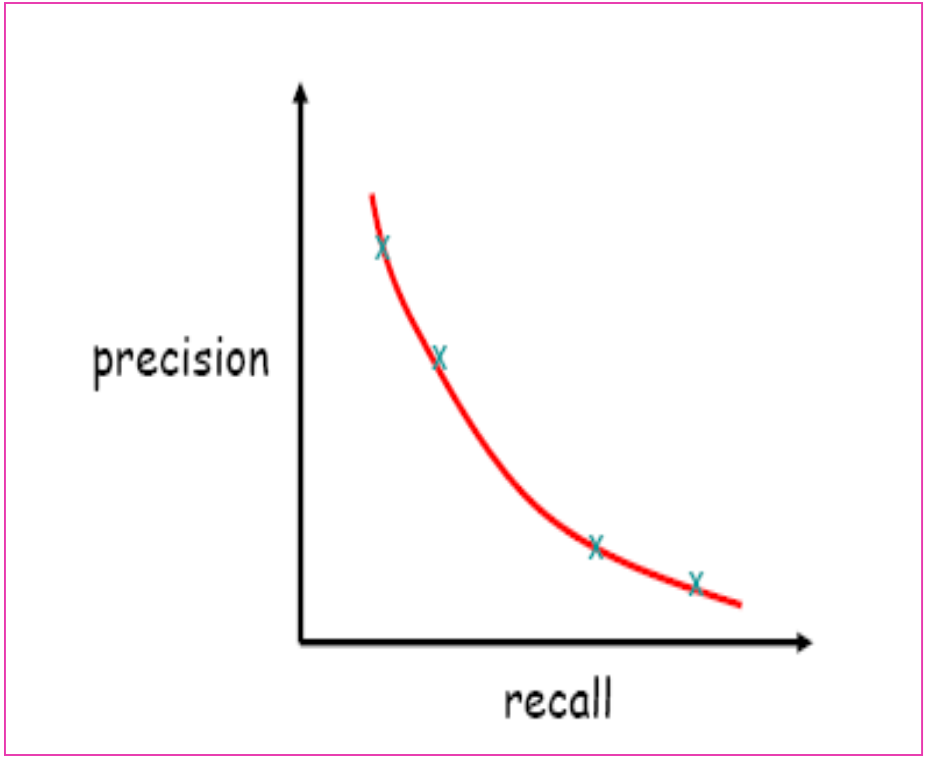
- Παράδειγμα:
  - Έστω 2 διαφορετικά συστήματα ΑΠ:
    - Answer(System1,q) = N N N N N N N R R R
    - Answer(System2,q) = R R R N N N N N N N
- Η ανάκληση & ακρίβεια είναι ίδια ☺

Λύση του προβλήματος

- Χρήση καμπύλης ακρίβειας/ανάκλησης
- Τρόπος υπολογισμού
  - Για κάθε ερώτημα παίρνουμε τη λίστα των απαντήσεων
  - Σημειώνουμε κάθε συναφές έγγραφο
  - Υπολογίζουμε το ζεύγος ακρίβειας-ανάκλησης για κάθε θέση της λίστας που έχει συναφές έγγραφο.

Καμπύλη Ακρίβειας-Ανάκλησης

- Η ανάκληση και η ακρίβεια είναι αντιστρόφως ανάλογες.
- Μετρούμε την ακρίβεια σε διαφορετικά επίπεδα ανάκλησης.



Καμπύλη Ακρίβειας-Ανάκλησης: Παράδειγμα

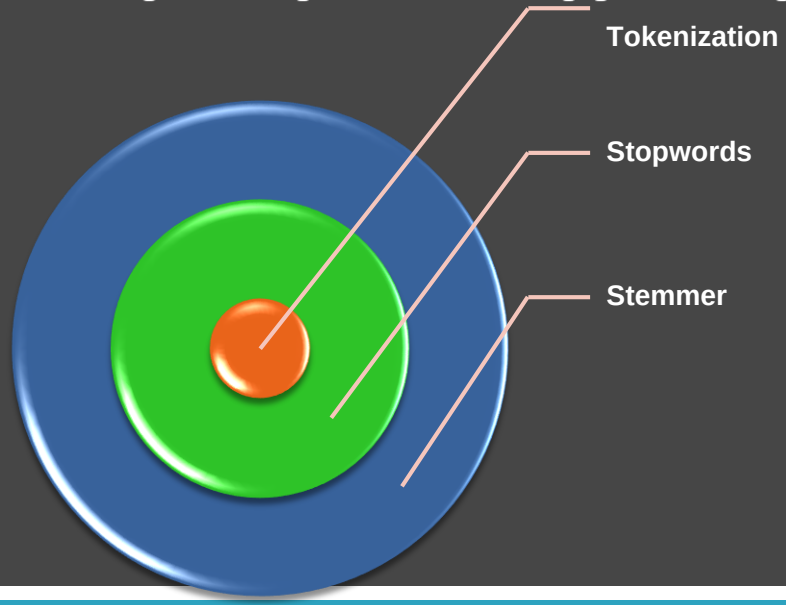
- Έστω τα συνολικά συναφή έγγραφα=6
- Έλεγχος στα σημεία με συναφή έγγραφα

n	doc #	συναφές
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

Ανάκληση	Ακρίβεια
R=1/6=0.176	P=1/1=1
R=2/6=0.333	P=2/2=1
R=3/6=0.5	P=3/4=0.75
R=4/6=0.667	P=4/6=0.667
R=5/6=0.844	P=5/13=0.38

Παρατήρηση!
Ελλείψει ενός συναφούς εγγράφου → η ανάκληση δεν γίνεται **ποτέ** 100%

Ανάκτηση Πληροφορίας



Ενότητα : Προ-επεξεργασία Κειμένου

Εισαγωγικές έννοιες

- Προεπεξεργασία κειμένου
  - Πριν από τη δεικτοδότηση των κειμένων προηγούνται μερικές βασικές διαδικασίες οι οποίες χρησιμοποιούνται για την απλοποίηση των κειμένων. Το σύνολο των διεργασιών αυτών καλείται Προεπεξεργασία Κειμένου.

- Λεκτική ανάλυση (lexical analysis)
- Απαλοιφή stopwords
- Stemming
- Επιλογή index terms
- Δημιουργία δομών κατηγοριών

Δειγματοποίηση - Tokenization

- Μετασχηματισμός των κειμένων σε μία ακολουθία από διακριτά αλφαριθμητικά (λέξεις?) που καλούμε tokens
- Μερικές φορές τα σημεία στίξης (e-mail), οι αριθμοί (1999), και η χρήση κεφαλαίων (Republican vs. republican) μπορεί να αποτελούν σημαντικό τμήμα ενός token
  - Όμως, πολύ συχνά αυτό δεν ισχύει
- Η πιο απλή προσέγγιση:
  - Τα σημεία στίξης αγνοούνται
  - Τα tokens αποτελούνται από ακολουθίες αλφαβητικών χαρακτήρων

- Πιο προσεκτική προσέγγιση:
  - Διαχωρισμός των ? ! ; : “ ‘ [ ] ( ) < >
  - Προσοχή με την τελεία **. (συντομογραφίες, ακρωνύμια)**
- Κατηγορίες που πρέπει να εξετάζονται:
  - Αριθμητικά ψηφία
  - Συλλαβισμός
  - Σύμβολα Στίξης
  - Μικρά και Κεφαλαία Γράμματα

Tokenization

- Αριθμητικά ψηφία:
  - Οι αριθμοί δεν θεωρούνται καλές περιπτώσεις index terms διότι χωρίς τα συμφραζόμενα το νόημά τους είναι αρκετά ασαφές.
  - Γενικά, τα συστήματα IR δεν περιλαμβάνουν τους αριθμούς στη λίστα των index terms.
  - Ωστόσο, υπάρχουν περιπτώσεις στις οποίες απαιτείται ιδιαίτερη προσοχή. Για παράδειγμα, κείμενα τα οποία περιέχουν *αριθμούς πιστωτικών καρτών*.

- Συλλαβισμός
  - Συνήθως η απαλοιφή του συμβόλου συλλαβισμού (‘-’) δε δημιουργεί προβλήματα στην ανάκτηση πληροφορίας (π.χ. State-of-the-art -> state of the art)
  - Ωστόσο απαιτείται προσοχή, διότι υπάρχουν λέξεις στις οποίες το σύμβολο ‘-’ παίζει σημαντικό ρόλο (B-52)

Tokenization

- Σύμβολα στίξης
  - Συνήθως τα σύμβολα στίξης αφαιρούνται εντελώς κατά τη φάση της λεκτικής ανάλυσης κειμένων και ερωτήσεων

  (I.K.A -> IKA, D.N.A. -> DNA)
  - Υπάρχουν ειδικές περιπτώσεις οι οποίες πρέπει να προσεχθούν ιδιαίτερα. Για παράδειγμα, σε ένα σύστημα IR το οποίο διαχειρίζεται κώδικα γραμμένο σε C/C++, υπάρχει διαφορά ανάμεσα στις εκφράσεις **x.id** και **xid**.

- Πεζά-κεφαλαία γράμματα
  - Κατά τη φάση της λεκτικής ανάλυσης όλα τα γράμματα μετατρέπονται σε μικρά ή σε κεφαλαία. (HORSE, Horse, horse)
  - Ειδικές περιπτώσεις πρέπει να αντιμετωπίζονται ξεχωριστά. Για παράδειγμα, κατά την αναζήτηση κειμένων που σχετίζονται με το λειτουργικό σύστημα Unix, η σημασία των εντολών ls –l και ls –L είναι διαφορετική.
- HTML
  - Το κείμενο εντός εντολών σε HTML που δεν είναι ορατό στο χρήστη πρέπει να συμπεριληφθεί στα tokens
  - Λέξεις που εμφανίζονται σε URLs
  - Λέξεις που εμφανίζονται σε “meta text” εικόνων.

StopWords

- Λέξεις οι οποίες εμφανίζονται στην πλειοψηφία των κειμένων δεν είναι καλές για index terms.
- Αυτές οι λέξεις καλούνται stopwords.
- Άρθρα, προθέσεις, σύνδεσμοι
  - “a”, “the”, “in”, “to” , “I”, “he”, “she”, “it”
- Η λίστα με τα stopwords είναι language-dependent

- Η απαλοιφή των stopwords μειώνει σημαντικά το μέγεθος ενός κειμένου.
- Ωστόσο, η απαλοιφή των stopwords μπορεί να μειώσει το recall. Για παράδειγμα αναζητώντας τη φράση “to be or not to be” ο χρήστης θα αντιμετωπίσει πρόβλημα. Για το λόγο αυτό πολλές μηχανές αναζήτησης στο WEB χρησιμοποιούν όλες τις λέξεις των κειμένων.
- Πώς καθορίζεται η λίστα με τα stopwords?
  - SMART’s commonword list (~ 400)
  - WordNet stopword list

Λημματοποίηση-Lemmatizer

- Εύρεση του λήμματος κλιτών λέξεων
  - *am, are, is → be*
  - *car, cars, car's, cars' → car*
- Άμεση επίδραση στο μέγεθος του λεξιλογίου
  - *the boy's cars are different colors → the boy car be different color*

- Πώς μπορεί να γίνει;
  - Χρειάζεται μία λίστα από γραμματικούς κανόνες και μία λίστα από ανώμαλες λέξεις
  - Children → child, spoken → speak …
  - Πρακτική υλοποίηση: χρήση της συνάρτησης morphstr() του WordNet

Stemming

- Ο μετασχηματισμός του token στη «ρίζα» του μπορεί να μειώσει δραστικά το μέγεθος του λεξιλογίου
  - “computer”, “computational”, “computation” → “compute”
- Η σωστή μορφολογική ανάλυση είναι language- specific και μπορεί να είναι αρκετά πολύπλοκη
- Με το stemming απλά κόβουμε γνωστά επιθέματα (προθέματα και καταλήξεις) με μία επαναληπτική διαδικασία

for example compressed and compression are both accepted as equivalent to compress.

for exampl compres and compres are both accept as equival to compres.

Ο αλγόριθμος του Porter

- Απλή διαδικασία απομάκρυνσης γνωστών επιθεμάτων για τα Αγγλικά χωρίς χρήση λεξικού
- Μπορεί να παράγει περίεργες «ρίζες» που δεν αντιστοιχούν σε πραγματικές λέξεις
  - "computer", "computational", "computation"→ "comput"
- Μπορεί να μετασχηματίσει στο ίδιο token λέξεις που δεν σχετίζονται
- Δεν αναγνωρίζει όλες τις μορφολογικές παραγωγές

- *Μερικοί τυπικοί κανόνες*
  - *sses → ss*
  - *es → i*
  - *ational → ate*
  - *tional → tion*
- Λάθος μετατροπή σε κοινή ρίζα:
  - organization, organ → organ
  - police, policy → polic
  - arm, army → arm

POS tagging

- Αναγνώριση μέρους του λόγου
- Συνήθως χρησιμοποιείται το Penn Treebank σύνολο ετικετών

- Π.χ.

NN	noun, common, singular or mass
RB	adverb
VB	verb, base form
JJ	adjective or numeral, ordinal