Relevance Feedback and Query Expansion

Εισαγωγή

- Στις συλλογές εγγράφων, κάποιες έννοιες απαντώνται με διαφορετικούς όρους
- Παράδειγμα: aircraft, plane
- Υπάρχουν συνώνυμες λέξεις
- Προσπαθούμε να χειριστούμε το συγκεκριμένο πρόβλημα με δύο ειδών μεθόδους:
  - Καθολικές (global)
  - Τοπικές (local)

Εισαγωγή

- Οι τοπικές μέθοδοι ανασχηματίζουν τα ερωτήματα σε σχέση με τα έγγραφα που ανακτήθηκαν αρχικά
- Οι τεχνικές αυτές είναι:
  - Ανατροφοδότηση συσχέτισης (relevance feedback)
  - Ψευδο-ανατροφοδότηση συσχέτισης ή τυφλή ανατροφοδότητη συσχέτισης (pseudo relevance feedback or Blind relevance feedback)
  - Καθολική έμμεση ανατροφοδότηση συσχέτισης (global indirect relevance feedback)
- Η πιο συχνά χρησιμοποιούμενη και επιτυχημένη μέθοδος είναι η ανατροφοδότηση συσχέτισης

Εισαγωγή

- Οι καθολικές μέθοδοι ανασχηματίζουν τα ερωτήματα και τους όρους τους
- Αυτές οι τεχνικές είναι:
  - Επέκταση των ερωτημάτων με τη βοήθεια ενός θησαυρού ή του WordNet
  - Επέκταση των ερωτημάτων μέσω αυτόματων θησαυρών
  - Τεχνικές που βασίζονται στο spelling correction

Ανατροφοδότηση Συσχέτισης

- Η ιδέα είναι να εμπλέξουμε το χρήστη στην ανάκτηση των εγγράφων
- Προσπαθούμε να βελτιώσουμε το αποτέλεσμα
- Η βασική διαδικασία έχει ως εξής:
  - Ο χρήστης θέτει ένα ερώτημα
  - Το σύστημα επιστρέφει ένα αρχικό σύνολο εγγράφων
  - Ο χρήστης μαρκάρει κάποια έγγραφα ως σχετικά ή όχι
  - Το σύστημα υπολογίζει μια καλύτερη αναπαράσταση / αποτέλεσμα βασιζόμενο στην πληροφορία που έδωσε ο χρήστης
  - Το σύστημα εμφανίζει το τελικό αποτέλεσμα

Ανατροφοδότηση Συσχέτισης

- Η τεχνική μπορεί να υιοθετήσει περισσότερες από μια επαναλήψεις
- Εστιάζει στο σκεπτικό ότι μπορεί να είναι δύσκολο το να θέσουμε ένα καλό ερώτημα εξ’ αρχής αλλά είναι εύκολο να κρίνουμε τα αποτελέσματα που θα δούμε
- Επίσης, η τεχνική έχει ένα ‘κρυφό’ πλεονέκτημα: ανιχνεύει τις ανάγκες των χρηστών – οι χρήστες κρίνουν τα έγγραφα που βλέπουν

Ανατροφοδότηση Συσχέτισης

- Παράδειγμα:

(a) Query: New space satellite applications

(b)
+ 1. 0.539, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
+ 2. 0.533, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
3. 0.528, 04/04/90, Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
4. 0.526, 09/09/91, A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
5. 0.525, 07/24/90, Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
6. 0.524, 08/22/90, Report Provides Support for the Critics Of Using Big Satellites to Study Climate
7. 0.516, 04/13/87, Arianespace Receives Satellite Launch Pact From Telesat Canada
+ 8. 0.509, 12/02/87, Telecommunications Tale of Two Companies

Ανατροφοδότηση Συσχέτισης

- Παράδειγμα (συνέχεια):

(c)

2.074 new 15.106 space
30.816 satellite 5.660 application
5.991 nasa 5.196 eos
4.196 launch 3.972 aster
3.516 instrument 3.446 arianespace
3.004 bundespost 2.806 ss
2.790 rocket 2.053 scientist
2.003 broadcast 1.172 earth
0.836 oil 0.646 measure

(d)

* 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
* 2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
4. 0.493, 07/31/89, NASA Uses 'Warm' Superconductors For Fast Circuit
* 5. 0.492, 12/02/87, Telecommunications Tale of Two Companies
6. 0.491, 07/09/91, Soviets May Adapt Parts of SS-20 Missile For Commercial Use
7. 0.490, 07/12/88, Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
8. 0.490, 06/14/90, Rescue of Satellite By Space Agency To Cost $90 Million

The Rocchio Algorithm

- Πρόκειται για τον πιο κλασικό αλγόριθμο για την υλοποίηση της ανατροφοδότησης συσχέτισης
- Εμπλέκει το αποτέλεσμα της ανατροφοδότησης στο vector space model
- Θέλουμε να βρούμε ένα διάνυσμα ερωτήματος που μεγιστοποιεί την ομοιότητα με τα σχετικά έγγραφα ενώ ταυτόχρονα ελαχιστοποιεί την ομοιότητα με μη σχετικά έγγραφα

The Rocchio Algorithm

- Έστω $\vec{q}$ το διάνυσμα ερωτήματος, C_r το σύνολο των σχετικών εγγράφων και C_{nr} το σύνολο των μη σχετικών εγγράφων
- Ψάχνουμε να βρούμε το:

$$\vec{q}_{opt} = \arg\max_{\vec{q}} [\text{sim}(\vec{q}, C_r) - \text{sim}(\vec{q}, C_{nr})]$$

- Όπου

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

- Το βέλτιστο διάνυσμα είναι:

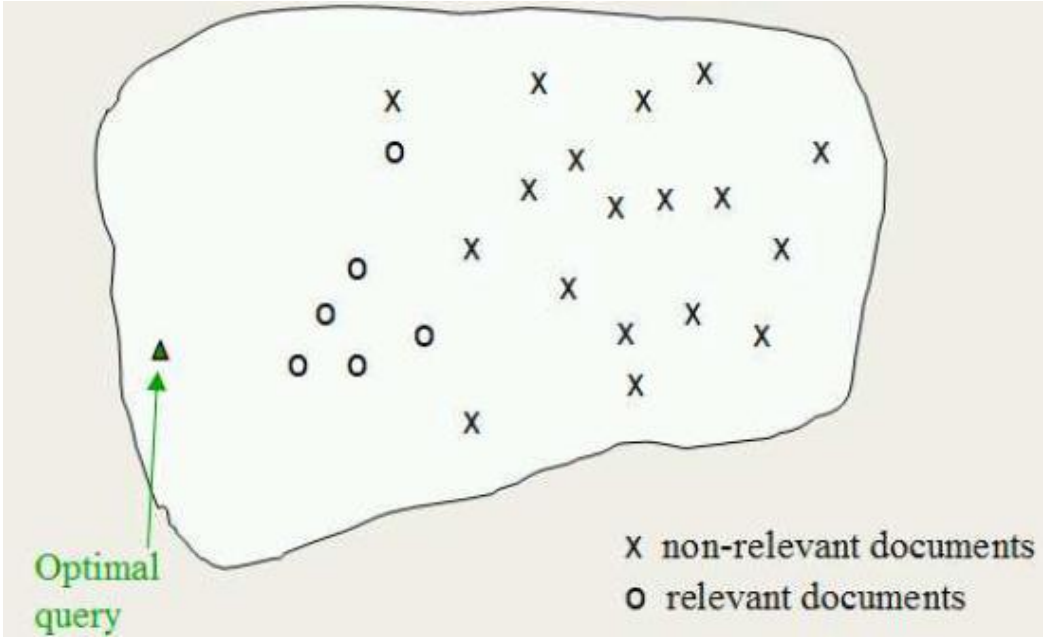
$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{|C_{nr}|} \sum_{\vec{d}_j \in C_{nr}} \vec{d}_j$$

The Rocchio Algorithm

- Στην εξίσωση:

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{|C_{nr}|} \sum_{\vec{d}_j \in C_{nr}} \vec{d}_j$$

- Ουσιαστικά υπολογίζουμε τη διαφορά των κεντροϊδών των διανυσμάτων



The Rocchio Algorithm

- Ο αλγόριθμος προτείνει μια βελτιωμένη έκδοση του υπολογισμού του διανύσματος ερωτήματος

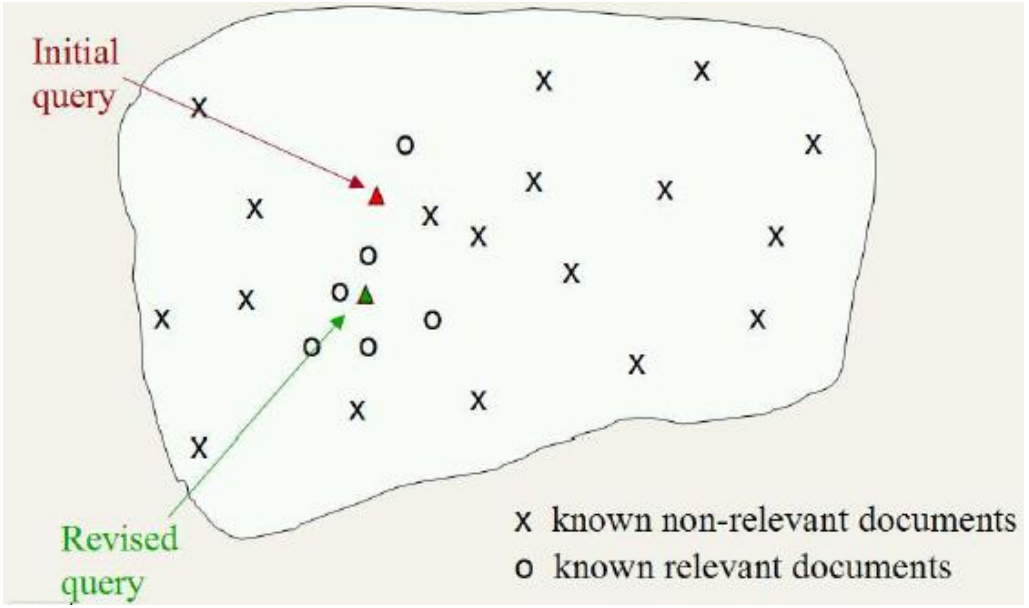
$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

- Το $\vec{q}_0$ είναι το αρχικό ερώτημα, τα D_r & D_{nr} τα σύνολα των σχετικών και μη εγγράφων και τα α, β, γ αποτελούν παραμέτρους του αλγορίθμου
- οι παράμετροι βοηθούν στην εξισορρόπηση των αποτελεσμάτων
- Παράδειγμα:
  - Αν έχουμε πολλά έγγραφα τα οποία τα έχουμε ταξινομήσει ως σχετικά, τότε μπορούμε να αυξήσουμε το β σε σχέση με το γ

The Rocchio Algorithm

- Το νέο ερώτημα ‘μετακινείται’ κατά μια απόσταση προς το centroid σχετικά έγγραφα
- Το νέο ερώτημα ‘απομακρύνεται’ κατά μια απόσταση μακριά από το centroid των μη σχετικων εγγράφων
- Το νέο ερώτημα χρησιμοποείται στο vector space model για την εξαγωγή των τελικών αποτελεσμάτων

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$



The Rocchio Algorithm

- Η τεχνική βελτιώνει και το precision και το recall
- Η θετική ανατροφοδότηση είναι πιο σημαντική από την αρνητική
- Στα περισσότερα συστήματα έχουμε γ<β
- Κάποιες τιμές μπορεί να είναι: α=1, β=0.75, γ=0.15
- Άλλα συστήματα επιτρέπουν μόνο θετικές ανατροφοδοτήσεις (γ=0)

Πιθανοτικοθεωρητική Προσέγγιση

- Αν έχουμε την πληροφόρηση από τους χρήστες για τα σχετικά και μη έγγραφα μπορούμε να χτίσουμε ένα κατηγοριοποιητή (classifier)
- Η πιο απλή τεχνική είναι ένας Naïve Bayes classifier
- Αν R είναι ένας Boolean indicator που δείχνει το αν ένα έγγραφο είναι σχετικό, τότε μπορούμε να εκτιμήσουμε την πιθανότητα $P(x_t=1|R)$
- Η πιθανότητα $P(x_t=1|R)$ μας δείχνει την πιθανότητα ύπαρξης ενός όρου σε ένα έγγραφο δεδομένου ότι είναι σχετικό ή μη

Πιθανοτικοθεωρητική Προσέγγιση

- Η πιθανότητα $P(x_t=1|R)$ υπολογίζεται ως εξής:

$$\hat{P}(x_t = 1|R = 1) = |VR_t|/|VR|$$
$$\hat{P}(x_t = 1|R = 0) = (df_t - |VR_t|)/(N - |VR|)$$

- Ν είναι ο συνολικός αριθμός των εγγράφων, df_t είναι το πλήθος που περιέχουν τον όρο t, VR είναι το σύνολο των σχετικών εγγράφων και VR_t είναι το σύνολο των σχετικών εγγράφων που περιέχουν τον όρο t
- Παράδειγμα:
  - Πλήθος εγγράφων που περιέχουν τον όρο: 100
  - Πλήθος σχετικών εγγράφων: 60, πλήθος σχετικών που περιέχουν τον όρο: 40
  - Συνολικό πλήθος εγγράφων: 200
  - $P(x_t=1|R=1) = 40/60=0.67$
  - $P(x_t=1|R=0) = (100-40)/(200-60)=0.43$

Ψευδο-Ανατροφοδότηση Συσχέτισης

- Η ψευδο-ανατροφοδότηση ή τυφλή ανατροφοδότηση συσχέτισης (pseudo relevance feedback – blind relevance feedback) είναι μια μέθοδος που υιοθετείται για τοπική ανάλυση
- Αυτοματοποιεί το manual μέρος της βασικής τεχνικής
- Ο χρήστης δεν χρειάζεται να έχει μια εκτεταμένη αλληλεπίδραση με το σύστημα
- Εξάγουμε κανονικά τα έγγραφα και στη συνέχεια υποθέτουμε ότι τα top-k είναι σχετικά
- Στη συνέχεια ξαναεφαρμόζουμε την τεχνική πάνω στα top-k έγγραφα

Ψευδο-Ανατροφοδότηση Συσχέτισης

- Η απόδοσή της φαίνεται στον επόμενο πίνακα

	Precision at $k = 50$	
Term weighting	no RF	pseudo RF
lnc.ltc	64.2%	72.7%
Lnu.ltu	74.2%	87.0%

- Όμως έχει τα μειονεκτήματα μιας οποιασδήποτε αυτοματοποιημένης τεχνικής
- Αν δώσουμε ένα ερώτημα ‘καλύτερα εστιατόρια’ και τα top-k έγγραφα αφορούν σε ‘εστιατόρια στη Λαμία’ τότε μπορεί να κατευθυνθούμε προς έγγραφα που σχετίζονται με τη Λαμία

Έμμεση Ανατροφοδότηση Συσχέτισης

- Μπορούμε να βρούμε έμμεσους τρόπους για να αποτιμήσουμε τη συσχέτιση
- Πρόκειται όμως για μια λιγότερο αξιόπιστη τεχνική
- Είναι όμως πιο χρήσιμη από την ψευδο-ανατροφοδότηση
- Μια προσέγγιση είναι να λάβουμε υπόψιν τα κλικ σε συνδέσμους που κάνουν οι χρήστες
- Τα έγγραφα που έχουν πολλές επισκέψεις είναι αυτά που θα μπουν πιο ψηλά στη λίστα
- Η υπόθεση είναι ότι οι συνόψεις των εγγράφων αποτελούν το μέσο για την εξαγωγή της συσχέτισης με ολόκληρο το έγγραφο

Υποθέσεις

- Οι χρήστες πρέπει να έχουν γνώση στο πως να θέσουν καλά ερωτήματα
- Γενικά η τεχνική δεν λειτουργεί σωστά στις ακόλουθες περιπτώσεις:
  - Σε misspellings. Αν ο χρήστης δεν θέσει σωστά τους όρους, η τεχνική δεν μπορεί να είναι αποδοτική
  - Σε cross—language ανακτήσεις. Τα έγγραφα που είναι σε άλλη γλώσσα δεν μπορεί να είναι κοντά στο vector space model
  - Σε ανομοιότητα στο λεξικό που υιοθετούν οι χρήστες και τα έγγραφα. Παράδειγμα: laptop - notebook

Υποθέσεις

- Η τεχνική απαιτεί τα έγγραφα που σχετίζονται να είναι όμοια μεταξύ τους
- Να μπορούμε να δημιουργήσουμε συστάδες
- Ιδεατά, η κατανομή των όρων στα σχετικά έγγραφα πρέπει να είναι ίδια και εντελώς διαφορετική από την κατανομή των όρων από τα έγγραφα που είναι μη σχετικά
- Η τεχνική δεν λειτουργεί αν τα σχετικά έγγραφα δημιουργούν αρκετές συστάδες

Προβλήματα

- Οι χρήστες αρκετές φορές δεν έχουν τη διάθεση να εισάγουν την ανατροφοδότηση
- Δεν επιθυμούν να αυξήσουν το χρόνο αλληλεπίδρασης με το σύστημα
- Πολλές φορές δεν καταλαβαίνουν γιατί κάποια έγγραφα έχουν βρεθεί στα αποτελέσματα
- Μεγάλου μήκους ερωτήματα είναι δύσκολα διαχειρίσιμα

Καθολικές Μέθοδοι

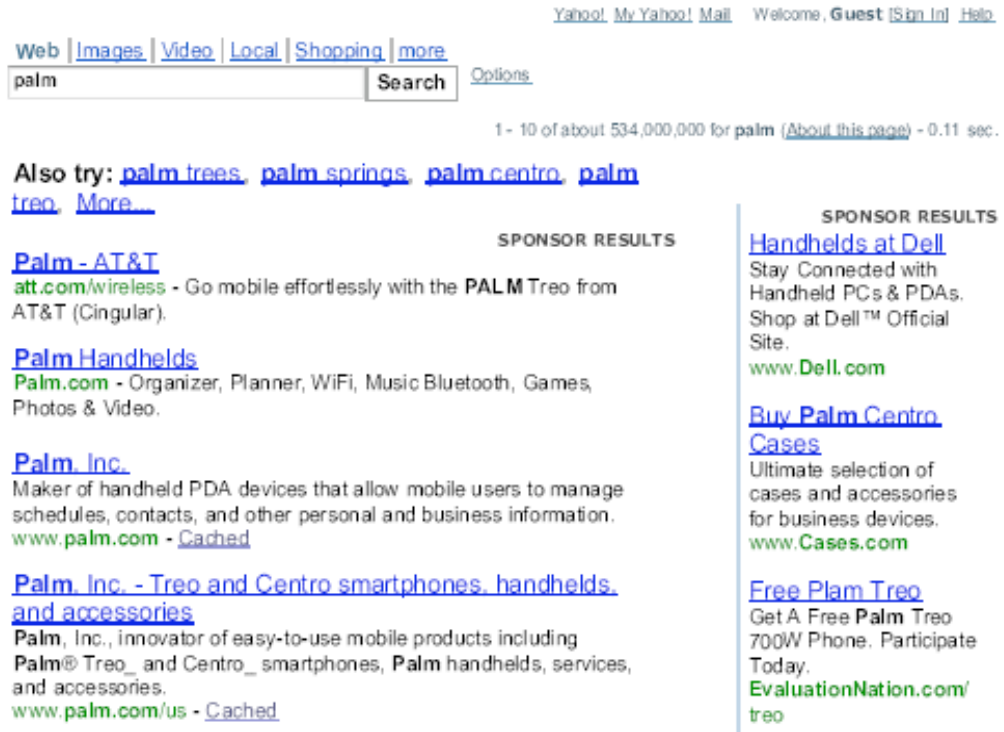
- Θα εξετάσουμε τρεις τεχνικές:
  - Χρήση λεξικού
  - Χρήση manual θησαυρού
  - Χρήση αυτόματου λεξικού

Χρήση Λεξικού

- Υιοθετούμε κάποιο λεξικό με όρους που έχουν απορριφθεί από τα ερωτήματα π.χ. λόγω του ότι ανήκουν σε stop words, λόγω του πλήθους των hits, κ.λπ.
- Μπορεί να επιτραπεί στους χρήστες να επιλέξουν καλούς όρους που υπάρχουν στο inverted index

Επέκταση Ερωτημάτων

- Οι χρήστες δίνουν περισσότερες πληροφορίες για τους όρους ενός ερωτήματος πιθανώς προτείνοντας επιπλέον όρους
- Κάποιες μηχανές αναζήτησης προτείνουν σχετικά ερωτήματα ως απάντηση στο ερώτημα



Επέκταση Ερωτημάτων

- Το κεντρικό ερώτημα είναι το πως δημιουργούμε τα εκτεταμένα ή τα εναλλακτικά ερωτήματα
- Ο πιο κοινός τρόπος είναι η χρήση ενός θησαυρού
- Για κάθε όρο t σε ένα ερώτημα, το ερώτημα μπορεί να επεκταθεί με συνώνυμα του t που θα βρούμε σε ένα θησαυρό
- Οι νέοι όροι μπορεί να λάβουν μικρότερο βάρος σε σχέση με τους αρχικούς όρους

Επέκταση Ερωτημάτων

- Οι μέθοδοι για να χτίσουμε ένα θησαυρό περιλαμβάνουν:
  - Χρήση λεξικού που διατηρείται από ανθρώπους – κάθε έννοια έχει και ένα κανονικό όρο
  - Χρήση manual θησαυρού – έχουμε ανθρώπινη παρέμβαση για την καταγραφή των συνωνύμων χωρίς να έχουμε ένα βασικό κανονικό όρο για κάθε έννοια
  - Χρήση αυτοματοποιημένου θησαυρού – υιοθετούνται στατιστικές που απεικονίζουν τη σύνδεση εμφάνισης μεταξύ των λέξεων
  - Ανασχηματισμός του ερωτήματος – για τους νέους χρήστες υιοθετούμε τους ανασχηματισμούς των ερωτημάτων στους οποίους προχώρησαν προηγούμενοι χρήστες

Επέκταση Ερωτημάτων

- Πλεονεκτήματα:
  - Δεν απαιτείται η εμπλοκή των χρηστών
  - Αυξάνει το recall

Αυτόματη Εξαγωγή του Θησαυρού

- Εξάγουμε το θησαυρό μέσα από ανάλυση των εγγράφων
- Ένας τρόπος είναι να αναλύσουμε τη συνύπαρξη των λέξεων
- Θεωρούμε πως όταν οι λέξεις συνυπάρχουν σε ένα έγγραφο ή μια παράγραφο είναι πιθανό να είναι όμοιες
- Άλλη προσέγγιση είναι να υιοθετήσουμε γραμματική ανάλυση του κειμένου
- Έπειτα βρίσκουμε γραμματικές συσχετίσεις ή εξαρτήσεις

Jaccard Similarity

- Ορίζεται ως το μέγεθος των τομών δύο συνόλων προς το μέγεθος της ένωσής τους
- Αναπαριστούμε και τα ερωτήματα και τα έγγραφα σαν σύνολα στοιχείων / όρων
- Ισχύει ότι:

$$sim_{\text{Jac}}(D, Q) = \frac{|D \cap Q|}{|D \cup Q|}$$

Jaccard Similarity

- Παράδειγμα:
  - A = {0,1,2,5,6}
  - B = {0,2,3,4,5,7,9}
  - $sim_{Jacc}(A,B) = |A \cap B| / |A \cup B| = |\{0,2,5\}| / |\{0,1,2,3,4,5,6,7,9\}| = 3/9 = 0.33.$

Dice Similarity

- Κατά αναλογία με την ομοιότητα Jaccard, η ομοιότητα Dice ορίζεται ως εξής:

$$sim_{\text{Dice}}(D, Q) = \frac{2|D \cap Q|}{|D| + |Q|}$$

- Παράδειγμα:
  - A={ni,ig,gh,ht}
  - B={na,ac,ch,ht}
  - sim_{Dice} = (2 · 1) / (4 + 4) = 0.25

Scoring and Term Weighting

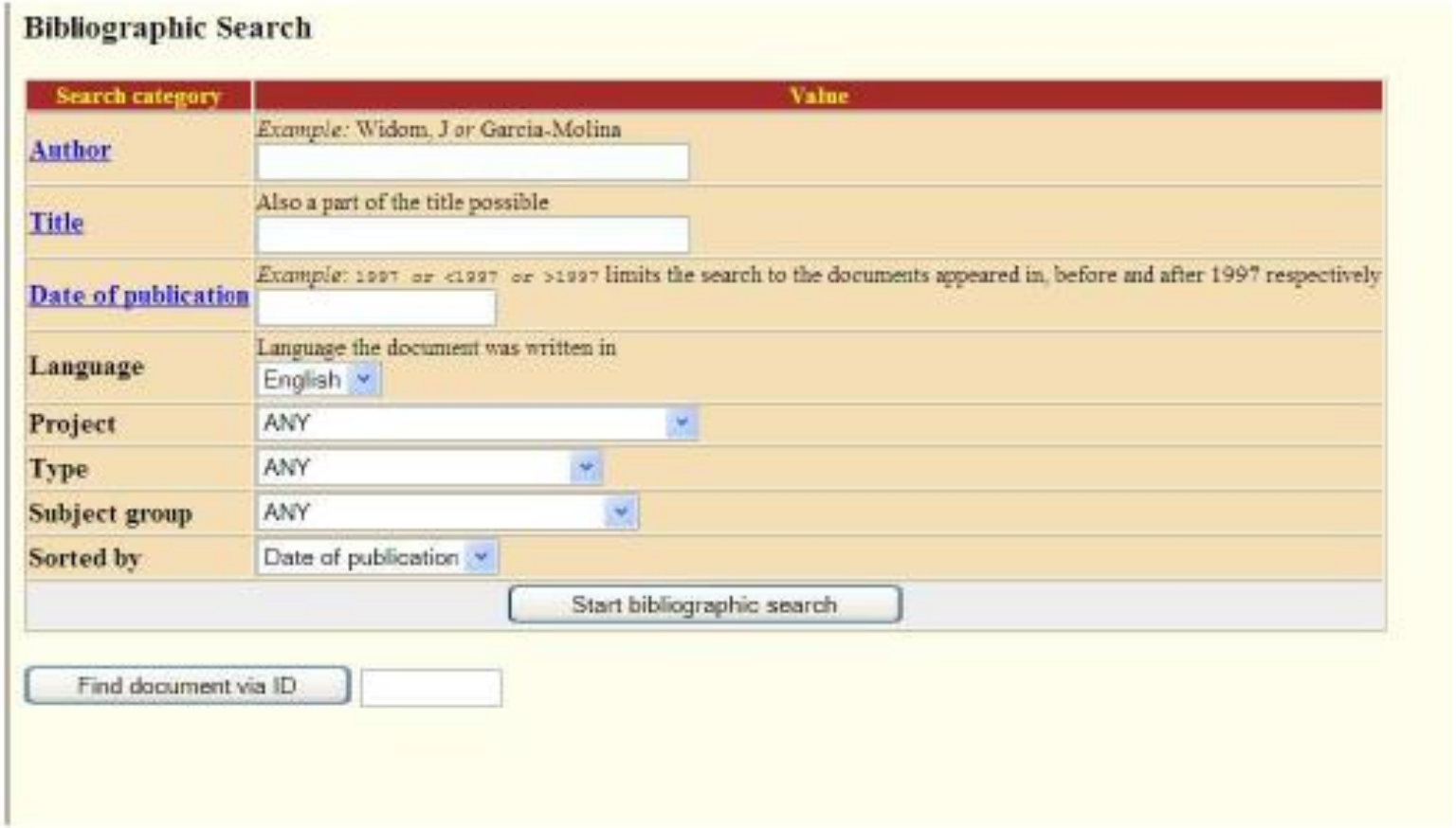
Parametric and Zone Indexes

- Στην πραγματικότητα τα έγγραφα δεν αποτελούν απλές ακολουθίες όρων
- Περιλαμβάνουν κάποια μεταδεδομένα (metadata)
- Παραδείγματα:
  - Συγγραφέας
  - Ημερομηνία δημοσίευσης
  - Κ.λπ.
- Κάθε μεταδεδομένο ονομάζεται πεδίο (field)

Parametric and Zone Indexes

- Πέρα από τη συγχώνευση των postings lists θα πρέπει να συγχωνεύσουμε και τις **παραμέτρους (parametric indexes)**
- Υπάρχει ένα ευρετήριο για κάθε πεδίο
- Παράδειγμα:
  - Ημερομηνία δημιουργίας
  - Θα πρέπει να συγχωνεύσουμε λίστες με έγγραφα που έχουν δημιουργηθεί μια συγκεκριμένη ημερομηνία
- Τα ευρετήρια αυτά μας επιτρέπουν να επιλέξουμε έγγραφα που ικανοποιούν κάποιες συγκεκριμένες παραμέτρους

Parametric and Zone Indexes



Bibliographic Search

Search category	Value
Author	*Example:* Widom, J or Garcia-Molina
Title	Also a part of the title possible
Date of publication	*Example:* `1997` or `<1997` or `>1997` limits the search to the documents appeared in, before and after 1997 respectively
Language	Language the document was written in: English
Project	ANY
Type	ANY
Subject group	ANY
Sorted by	Date of publication

Start bibliographic search

Find document via ID

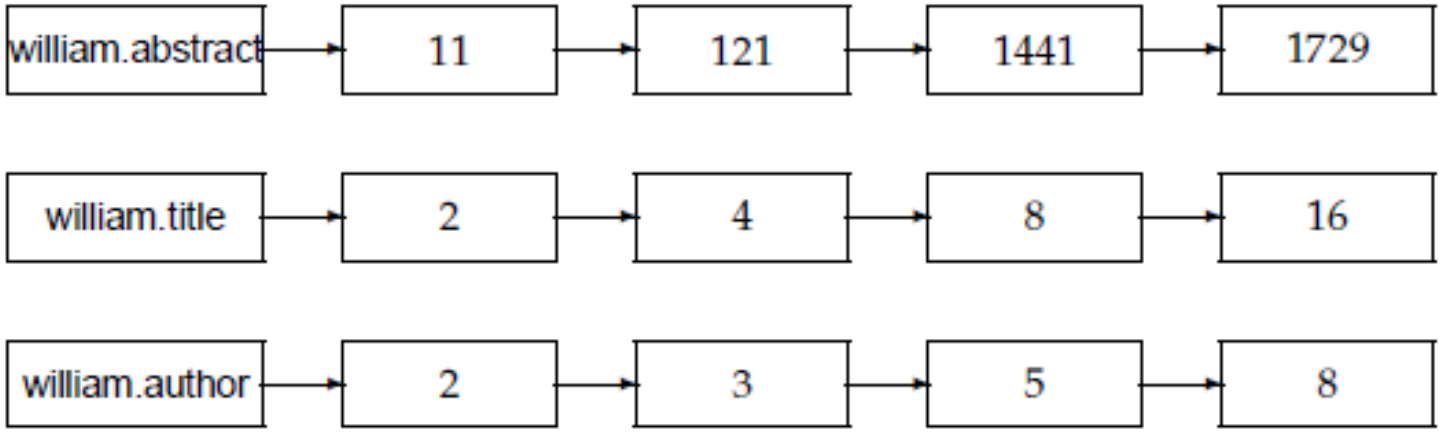
Parametric and Zone Indexes

- Οι **ζώνες (zones)** είναι παραπλήσιες με τα πεδία μόνο που μπορεί να είναι ελεύθερο κείμενο
- Ένα πεδίο μπορεί να λάβει μικρό αριθμό τιμών
- Μια ζώνη δεν έχει όριο τιμών
- Παράδειγμα:
  - Τίτλοι εγγράφων
  - Περιλήψεις εγγράφων
- Μπορούμε να χτίσουμε ένα ξεχωριστό inverted index για κάθε ζώνη ενός εγγράφου
- Με αυτό τον τρόπο υποστηρίζουμε ερωτήματα της μορφής:

find documents with merchant in the title and william in the author list and the phrase gentle rain in the body

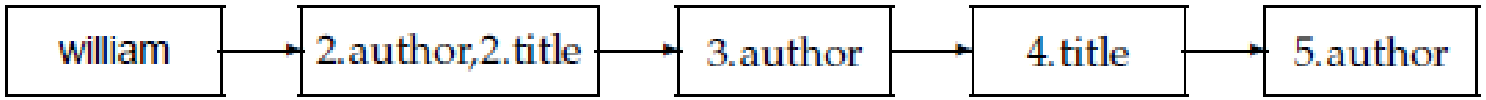
Parametric and Zone Indexes

- Το λεξικό για τις ζώνες πρέπει να απεικονίσει ότι περιλαμβάνεται στο κείμενο αυτής της ζώνης



Parametric and Zone Indexes

- Μπορούμε να μειώσουμε το μέγεθος του λεξικού κωδικοποιώντας κάθε ζώνη με κάθε όρο που απαντάται σε κάποιο έγγραφο
- Ο τρόπος αυτός μας βοηθάει πάρα πολύ στην επόμενη τεχνική που ονομάζεται **weighted zone scoring**



Weighted Zone Scoring

- Έστω ότι έχουμε να απαντήσουμε στο ερώτημα q και έχουμε ένα έγγραφο d
- Η τεχνική αναθέτει στο ζεύγος (q,d) μια τιμή (score) στο διάστημα [0,1]

- Κάθε ζώνη του εγγράφου συνεισφέρει με μια λογική τιμή

- Ο στόχος είναι να ταξινομήσουμε με κάποιο τρόπο τα αποτελέσματα
- Ονομάζεται αλλιώς ως ranked Boolean retrieval

Weighted Zone Scoring

- Έστω ότι έχουμε ένα σύνολο εγγράφων με l ζώνες το καθένα
- Έστω ότι $g_1, g_2, \ldots, g_l$ στο διάστημα [0,1] με το άθροισμα τους να ισούται με 1
- Για 1<=i<=l, έστω ότι το s_i είναι το Boolean σκορ που απεικονίζει ένα ταίριασμα (ή απουσία ταιριάσματος) ανάμεσα στο q και στη ζώνη i
- Προφανώς, το σκορ είναι 1 όταν ο όρος του ερωτήματος υπάρχει στη ζώνη και 0 διαφορετικά
- Μπορούμε να χρησιμοποιήσουμε οποιαδήποτε Boolean συνάρτηση που απεικονίζει στις τιμές 0 ή 1
- Το τελικό αποτέλεσμα είναι:
- $\sum_{i=1}^{l} g_i s_i$

Weighted Zone Scoring

- Αριθμητικό παράδειγμα:
  - Έστω 3 ζώνες: author, title, body
  - Θα έχουμε 1 αν ο όρος Δήμος θα υπάρχει στη ζώνη
  - Έστω ότι $g_1=0.6$, $g_2=0.3$, $g_3=0.1$
  - Πιο σημαντική είναι η $1^η$ ζώνη
  - Αν ο όρος Δήμος υπάρχει στα author & body η τελική τιμή είναι 0.7
  - Αν ο όρος Δήμος υπάρχει στο author η τελική τιμή είναι 0.6
  - **Ποια η τελική τιμή αν ο όρος Δήμος υπάρχει και στις τρεις ζώνες;**

Weighted Zone Scoring

- Μια προσέγγιση για να υπολογίσουμε την τελική τιμή για κάθε έγγραφο προσθέτωντας τα αποτελέσματα για όλες τις ζώνες
- Όμως μπορούμε να κάνουμε τον υπολογισμό απ’ ευθείας στο inverted index

Weighted Zone Scoring

- Ο αλγόριθμος υποθέτει πως το ερώτημα αποτελείται από 2 όρους
- Επίσης, υποθέτει πως εφαρμόζουμε το λογικό τελεστή AND

```
ZONESCORE(q1, q2)
 1  float scores[N] = [0]
 2  constant g[ℓ]
 3  p1 ← postings(q1)
 4  p2 ← postings(q2)
 5  // scores[] is an array with a score entry for each document, initialized to zero.
 6  // p1 and p2 are initialized to point to the beginning of their respective postings.
 7  //Assume g[] is initialized to the respective zone weights.
 8  while p1 ≠ NIL and p2 ≠ NIL
 9  do if docID(p1) = docID(p2)
10        then scores[docID(p1)] ← WEIGHTEDZONE(p1, p2, g)
11             p1 ← next(p1)
12             p2 ← next(p2)
13        else if docID(p1) < docID(p2)
14                then p1 ← next(p1)
15                else p2 ← next(p2)
16  return scores
```

Εκμάθηση των Βαρών

- Ένα μεγάλο ερώτημα είναι το πως θα καθοριστούν τα βάρη g_i
- Ένας τρόπος είναι να καθοριστούν από experts
- Όμως μπορούμε να υιοθετήσουμε και τεχνικές μηχανικής μάθησης ώστε να υπολογιστούν μέσα από samples

Εκμάθηση των Βαρών

- Πρέπει να καθορίσουμε ένα σύνολο από training examples
- Το καθένα θα αποτυπώνει το συνδυασμό ενός ερωτήματος q και ενός εγγράφου d μαζί με ένα χαρακτηριστικό
- Συνήθως ο χαρακτηρισμός μπορεί να είναι Σχετικό – Μη Σχετικό
- Τα βάρη καθορίζονται μέσα από αυτό το training dataset ώστε η τιμή εκμάθησης να βοηθά στην προσέγγιση των χαρακτηρισμών
- Το κύριο πρόβλημα είναι το πως θα προσεγγίσουμε τους χαρακτηρισμούς ιδιαίτερα όταν στο Web τα έγγραφα και τα ερωτήματα αλλάζουν συνεχώς

Εκμάθηση των Βαρών

- Ας υποθέσουμε πως έχουμε έγγραφα τα οποία χαρακτηρίζονται από μια ζώνη τίτλου και μια ζώνη του κυρίως σώματος του εγγράφου
- Δοσμένου ενός ερωτήματος q και ενός εγγράφου d υπολογίζουμε τις μεταβλητές $s_T(d,q)$ & $s_B(d,q)$ με βάση μια Boolean συνάρτηση
- Προφανώς τα αποτελέσματα εξαρτώνται από το αν το ερώτημα περιλαμβάνεται στον τίτλο ή στο κυρίως σώμα του εγγράφου
- Μπορούμε να υπολογίσουμε το τελικό σκορ ως εξής:

$$score(d,q) = g \cdot s_T(d,q) + (1-g) \cdot s_B(d,q)$$

- Ανάλογα με την τιμή του g δίνουμε βάρος στον τίτλο ή στο κυρίως σώμα του εγγράφου

Εκμάθηση των Βαρών

- Έστω ότι θέλουμε να υπολογίσουμε το βάρος g
- Δεχόμαστε το training set στη μορφή πλειάδων όπως $\Phi_j = (d_j, q_j, r(d_j, q_j))$
- Για κάθε Φ_j έχουμε τις Boolean τιμές που υιοθετούμε για τον υπολογισμό του τελικού σκορ

$$score(d_j, q_j) = g \cdot s_T(d_j, q_j) + (1 - g) \cdot s_B(d_j, q_j)$$

Example	DocID	Query	s_T	s_B	Judgment
Φ_1	37	linux	1	1	Relevant
Φ_2	37	penguin	0	1	Non-relevant
Φ_3	238	system	0	1	Relevant
Φ_4	238	penguin	0	0	Non-relevant
Φ_5	1741	kernel	1	1	Relevant
Φ_6	2094	driver	0	1	Relevant
Φ_7	3191	driver	1	0	Non-relevant

Εκμάθηση των Βαρών

- Στη συνέχεια συγκρίνουμε το τελικό σκορ με το αποτέλεσμα που δίνουν οι experts
- Αντιστοιχούμε το Relevant στο 1 και το Non-Relevant στο 0
- Προσπαθούμε να αποτυπώσουμε το σφάλμα σε περίπτωση διαφοράς με το αποτέλεσμα που πρέπει να πάρουμε
- Έστω ότι χρησιμοποιούμε τη συνάρτηση:

  $\varepsilon(g,\Phi_j) = (r(d_j, q_j) - score(d_j, q_j))^2$
- Το συνολικό σφάλμα θα είναι: $\sum_j \varepsilon(g, \Phi_j)$
- Ο στόχος μας πλέον είναι να επιλέξουμε το g ώστε να ελαχιστοποιηθεί το συνολικό σφάλμα

Εκμάθηση των Βαρών

- Ο ακόλουθος πίνακας μας δείχνει όλους τους πιθανούς συνδυασμούς βαρών

s_T	s_B	Score
0	0	0
0	1	$1-g$
1	0	g
1	1	1

- Έστω ότι n_{01r} και n_{01n} είναι το πλήθος των training tuples όπου $s_T(d_j, q_j) = 0$ and $s_B(d_j, q_j) = 1$ και ο χαρακτηρισμός είναι Relevant ή Non-Relevant αντίστοιχα
- Η συνεισφορά στο συνολικό σφάλμα είναι: $[1 - (1-g)]^2 n_{01r} + [0 - (1-g)]^2 n_{01n}$

Εκμάθηση των Βαρών

- Με το ίδιο σκεπτικό αποτυπώνουμε το σφάλμα και για τους άλλους τρεις συνδυασμούς
- Μετά από υπολογισμούς το συνολικό σφάλμα θα είναι:

  $(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1 - g)^2 + n_{00r} + n_{11n}$
- Πρόκειται για συνάρτηση του g την οποία παραγωγίζουμε για να πάρουμε τη βέλτιστη τιμή του
- Το τελικό αποτέλεσμα είναι:

$$g = \frac{n_{10r} + n_{01n}}{n_{10r} + n_{10n} + n_{01r} + n_{01n}}$$

Συχνότητα Όρων

- Μέχρι στιγμής έχουμε εστιάσει στο αν κάποια ζώνη του εγγράφου περιλαμβάνει κάποιους όρους του ερωτήματος
- Η επόμενη φάση είναι να δούμε το ποια η συχνότητα εμφάνισης ώστε να δώσουμε μεγαλύτερο βάρος στο αντίστοιχο έγγραφο
- Σε κάθε όρο αναθέτουμε ένα βάρος που εξαρτάται από το πλήθος των εμφανίσεων του όρου
- Η πιο απλή προσέγγιση είναι να αναθέσουμε το βάρος ίσο με το πλήθος των εμφανίσεων
- Αυτό το σχήμα ονομάζεται term frequency και συμβολίζεται με $tf_{t,d}$

Συχνότητα Όρων

- Για ένα έγγραφο το σύνολο των βαρών που καθορίζονται με το tf μπορεί να θεωρηθεί σαν μια περίληψη / σύνοψη του
- Η προσέγγιση είναι γνωστή ως bag of words model
- Το ranking των όρων το παραλείπουμε
- Η μόνη πληροφορία που κρατάμε είναι αυτή που σχετίζεται με τη συχνότητα εμφάνισης
- Προφανώς, με αυτή την προσέγγιση όμοια bags of words θα οδηγήσουν στο συμπέρασμα για όμοια έγγραφα

Συχνότητα Όρων

- Η προηγούμενη τεχνική είναι απλή και δεν λαμβάνει υπόψιν της το γεγονός πως κάποιοι όροι είναι πιο σημαντικοί για ένα έγγραφο σε σχέση με κάποιους άλλους
- Παράδειγμα:
  - Μια συλλογή εγγράφων που σχετίζεται με την αυτοκινητοβιομηχανία θα έχει τους όρους car, auto σε σχεδόν όλα τα έγγραφα
- Έχει προταθεί μηχανισμός που λαμβάνει υπόψιν του την επίπτωση των όρων που απαντώνται πολύ συχνά
- Μπορούμε να μειώσουμε το βάρος των όρων που έχουν υψηλή συχνότητα μέσα σε μια συλλογή εγγράφων
- Παράδειγμα: μείωση κατά ένα παράγοντα που μεγαλώνει μαζί με τη συχνότητα εμφάνισης

Συχνότητα Εγγράφων

- Υιοθετούμε το document frequency – df_t
- Ορίζεται ως το πλήθος των εγγράφων που περιέχουν τον όρο t
- Η μετρική προτιμάται σε σχέση με το collection frequency – cf
- Παράδειγμα:

Word	cf	df
try	10422	8760
insurance	10440	3997

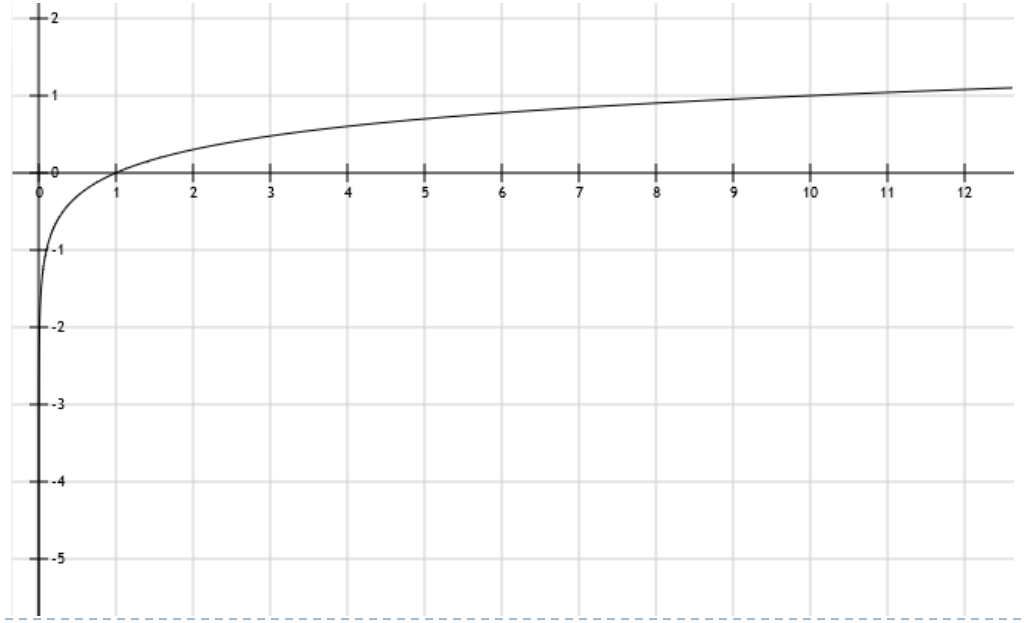
  - Οι τιμές για τα try & insurance για το cf είναι περίπου ίδιες
  - Οι df τιμές τους όμως είναι εντελώς διαφορετικές

Συχνότητα Εγγράφων

- Έστω ότι το συνολικό πλήθος των εγγράφων σε μια συλλογή είναι N
- Ορίζουμε την inverse document frequency (idf) ενός όρου t ως ακολούθως:

$$idf_t = log \frac{N}{df_t}$$

- Με αυτό τον τρόπο το idf ενός σπάνιου όρου θα είναι υψηλό
- Το idf ενός συχνού όρου θα είναι χαμηλό



Παραδείγματα

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Συνδυασμός

- Συνδυάζουμε τα tf & idf για να πάρουμε καλύτερα αποτελέσματα
- Το σχήμα tf-idf αναθέτει στο όρο t ένα βάρος για ένα έγγραφο d μέσω του:

  $\mathbf{tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t}$
- Το $\text{tf-idf}_{t,d}$ αναθέτει ένα βάρος:
  - Που είναι μεγαλύτερο όταν ο όρος απαντάται πολλές φορές σε ένα μικρό αριθμό εγγράφων
  - Που είναι μικρότερο όταν ο όρος απαντάται σε πολλά έγγραφα
  - Που είναι το μικρότερο όταν ο όρος απαντάται σε όλα τα έγγραφα

Συνδυασμός

- Μπορούμε να δούμε ένα έγγραφο σαν ένα διάνυσμα με κάθε στοιχείο να αντιστοιχεί σε κάθε όρο που υπάρχει στο λεξικό
- Επίσης, μαζί τοποθετούμε και το βάρος του κάθε όρου
- Για όρους που δεν υπάρχουν σε κάποιο έγγραφο, το βάρος είναι ίσο με το 0
- Ορίζουμε το overlap score measure

$$\text{Score}(q, d) = \sum_{t \in q} \text{tf-idf}_{t,d}$$

Ασκήσεις

- Έστω οι όροι και οι συχνότητες που απεικονίζονται στο ακόλουθο πίνακα. Να υπολογιστούν τα tf-idf βάρη για τους όρους αυτούς για κάθε έγγραφο με βάση τα idf βάρη του δεύτερου πίνακα

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Ασκήσεις

- Λύση

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

	Doc1	Doc2	Doc3
car	44.55	6.6	39.6
Auto	6.24	68.64	0
Insurance	0	53.46	46.98
Best	21	0	25.5

Ασκήσεις

- Αποδείξτε πως η βάση του λογαρίθμου στον υπολογισμό του idf $idf_t = log \frac{N}{df_t}$ δεν διαδραματίζει ουσιαστικό ρόλο στον υπολογισμό.

Ασκήσεις

- Λύση
  - Έστω ότι για τον υπολογισμό υποθέτουμε βάση z>0
  - Έχουμε:

    $$idf = log_z(\frac{N}{df})$$

  - Προχωρούμε σε αλλαγή βάσης στο λογάριθμο και παίρνουμε:

    $$idf = log_z\left(\frac{N}{df}\right) = \frac{log_{10}(\frac{N}{df})}{log_{10}(z)}$$

  - Όπου το $\frac{1}{log_{10}(z)}$ είναι ένας σταθερός αριθμός που δεν επηρεάζει τη σχετική θέση κάθε όρου για ένα ερώτημα

The Vector Space Model

- Αναπαριστούμε με $\vec{V}(d)$ το διάνυσμα που εξάγεται από το έγγραφο d
- Κάθε στοιχείο του διανύσματος είναι και ένας όρος του λεξικού
- Τα στοιχεία μπορεί να έχουν υπολογιστεί με τη μέθοδο **tf-idf**
- Το σύνολο των εγγράφων μπορεί να θεωρηθεί ως ένα σύνολο διανυσμάτων στο χώρο διανυσμάτων με μια διάσταση για κάθε όρο του λεξικού
- Προφανώς αυτή η αναπαράσταση δεν λογίζει τη σχετική σειρά των όρων σε κάθε έγγραφο
- Το ερώτημα είναι το πως θα υπολογίσουμε την ομοιότητα των εγγράφων σε αυτό το χώρο διανυσμάτων;

The Vector Space Model

- Μια πρώτη προσπάθεια αφορά στον υπολογισμό της διαφοράς των διανυσμάτων
- Όμως παρά το γεγονός ότι δύο έγγραφα μπορεί να είναι τα ίδια, το αποτέλεσμα μπορεί να μην είναι ικανοποιητικό λόγω του μεγέθους των διανυσμάτων
- Η πιο διαδεδομένη τεχνική είναι η cosine similarity
- Αυτή ορίζεται ως εξής:

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

The Vector Space Model

- Στη μετρική cosine έχουμε:

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

  - Ο αριθμητής είναι το εσωτερικό γινόμενο των διανυσμάτων:

    $\sum_{i=1}^{M} x_i y_i$

  - Ο παρονομαστής είναι το γινόμενο των Euclidean lengths των δύο εγγράφων

    $\sqrt{\sum_{i=1}^{M} \vec{V}_i^2(d)}$

  - Ο παρονομαστής στοχεύει στο να κανονικοποιήσει τα διανύσματα σε unit vectors

$$\vec{v}(d_1) = \vec{V}(d_1) / |\vec{V}(d_1)|$$

$$\vec{v}(d_2) = \vec{V}(d_2) / |\vec{V}(d_2)|$$

The Vector Space Model

- Οπότε μπορούμε να ξαναγράψουμε την εξίσωση ως εξής:

$$\text{sim}(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2)$$

- Η απόσταση cosine μπορεί να θεωρηθεί ώς το γινόμενο των κανονικοποιημένων διανυσμάτων
- Η χρήση της μετρικής αφορά στην εξαγωγή όμοιων εγγράφων
- Το πρόβλημα μπορεί να το δούμε ως τον υπολογισμό του μεγαλύτερου γινομένου $\vec{v}(d) \cdot \vec{v}(d_i)$

The Vector Space Model

- Παράδειγμα:

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
document1	5	0	3	0	2	0	0	2	0	0
document2	3	0	2	0	1	1	0	1	0	1

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$ $d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 1 + 2 \times 1 + 0 \times 0 + 0 \times 1 = 25$

$$\| d_1 \| = \sqrt{5\times5+0\times0+3\times3+0\times0+2\times2+0\times0+0\times0+2\times2+0\times0+0\times0} = 6.481$$

$$\| d_2 \| = \sqrt{3\times3+0\times0+2\times2+0\times0+1\times1+1\times1+0\times0+1\times1+0\times0+1\times1} = 4.12$$

- Τελικό αποτέλεσμα: 25/(6.481X4.12) = 0.94

The Vector Space Model

- Υπολογίστε το cosine similarity για τα ακόλουθα έγγραφα:

Έγγραφο	Δήμος	Λαμία	Αττική	Πανεπιστήμιο	Πληροφορική
D1	3	1	2	0	10
D2	0	4	2	1	2
D3	4	2	2	1	1

The Vector Space Model

- Λύση:

Έγγραφο	Δήμος	Λαμία	Αττική	Πανεπιστήμιο	Πληροφορική
D1	3	1	2	0	10
D2	0	4	2	1	2
D3	4	2	2	1	1

- Sim(D1,D2)= 28/(10.67708X5) = 0.52
- Sim(D1,D3)= 28/(10.67708X5.09902) = 0.51
- Sim(D2,D3)= 15/(5X5.09902) = 0.59

The Vector Space Model

- Ο λόγος που αναπαριστούμε τα έγγραφα σαν διανύσματα είναι ότι μπορούμε με τον ίδιο τρόπο να αναπαραστήσουμε και τα ερωτήματα
- Οπότε πρέπει να υπολογίσουμε το γινόμενο:

  $\vec{v}(q) \cdot \vec{v}(d)$

- Χειριζόμαστε τα ερωτήματα σαν σύντομα έγγραφα
- Υπολογίζουμε τις αποστάσεις από τα έγγραφα και παίρνουμε τις υψηλότερες τιμές
- Ένα έγγραφο μπορεί να έχει υψηλό σκορ ακόμα και αν δεν περιέχει όλους τους όρους του ερωτήματος

The Vector Space Model

- Ο υπολογισμός της ομοιότητας με ένα πολύ μεγάλο σύνολο εγγράφων μπορεί να μην είναι αποδοτικός
- Ένα σύνολο τεχνικών υιοθετούνται για να εξάγουν γρήγορα το αποτέλεσμα

The Vector Space Model

- Σε ένα τυπικό setup όπου έχουμε μια συλλογή εγγράφων αναπαριστούμε το ερώτημα σαν ένα διάνυσμα επίσης και λαμβάνουμε υπόψιν μας μια τιμή K>0
- Αναζητούμε τα K έγγραφα στη συλλογή με το υψηλότερο σκορ
- Προφανώς, η ταξινόμηση θα είναι σε φθίνουσα σειρά

The Vector Space Model

- Έχουν προταθεί και άλλες τεχνικές για τον υπολογισμό του βάρους ενός όρου
- Ο λόγος είναι πως είναι απίθανο ένας όρος που απαντάται 20 φορές να είναι 20 φορές σημαντικός για ένα έγγραφο σε σχέση με μια απλή εμφάνιση
- Έχουν προταθεί αλλαγές στην τεχνική tf που εξετάζουν και άλλα στοιχεία πέρα από τη συχνότητα εμφάνισης ενός όρου

The Vector Space Model

- Μια παραλλαγή είναι να υιοθετήσουμε το λογάριθμο της συχνότητας του όρου

$$\mathrm{wf}_{t,d} = \begin{cases} 1 + \log \mathrm{tf}_{t,d} & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Οπότε αντικαθιστούμε τη μετρική tf στον υπολογισμό του tf-idf

$$\text{wf-idf}_{t,d} = \mathrm{wf}_{t,d} \times \mathrm{idf}_t$$

The Vector Space Model

- Άλλη παραλλαγή είναι να κανονικοποιήσουμε το tf με το μέγιστο tf σε ένα έγγραφο
- Έστω ότι

$$\mathrm{tf}_{\max}(d) = \max_{\tau \in d} \mathrm{tf}_{\tau,d}$$

- Το τ αναπαριστά όλους τους όρους σε ένα έγγραφο
- Η κανονικοποιημένη συχνότητα ενός όρου υπολογίζεται ως εξής:

$$\mathrm{ntf}_{t,d} = a + (1-a)\frac{\mathrm{tf}_{t,d}}{\mathrm{tf}_{\max}(d)}$$

- Το α είναι στο διάστημα [0,1] – έχει προταθεί να είναι ίσο με 0.4 ή 0.5
- Το α είναι ένας παράγοντας εξισορρόπησης που στοχεύει στο να ‘περιορίσει’ τη σημασία του δεύτερου όρου

The Vector Space Model

- Η βασική ιδέα της κανονικοποίησης με τη μέγιστη συχνότητα είναι να μετριάσουμε την επιρροή των όρων σε μεγάλα έγγραφα
- Παρατηρούμε μεγαλύτερες συχνότητες σε μεγάλα έγγραφα αφού οι όροι τείνουν να επαναλαμβάνονται

- Παράδειγμα:
  - Για ένα έγγραφο Ε φτιάχνουμε το Ε’ το οποίο είναι το Ε δύο φορές
  - Το Ε’ θα πάρει διπλό σκορ

The Vector Space Model

- Μεγάλα έγγραφα περιλαμβάνουν μεγαλύτερες συχνότητες όρων και άρα μεγαλύτερες tf τιμές
- Μεγάλα έγγραφα περιλαμβάνουν περισσότερους διακριτούς όρους
- Αυτοί οι παράγοντες μπορεί να αυξήσουν τα βάρη των όρων

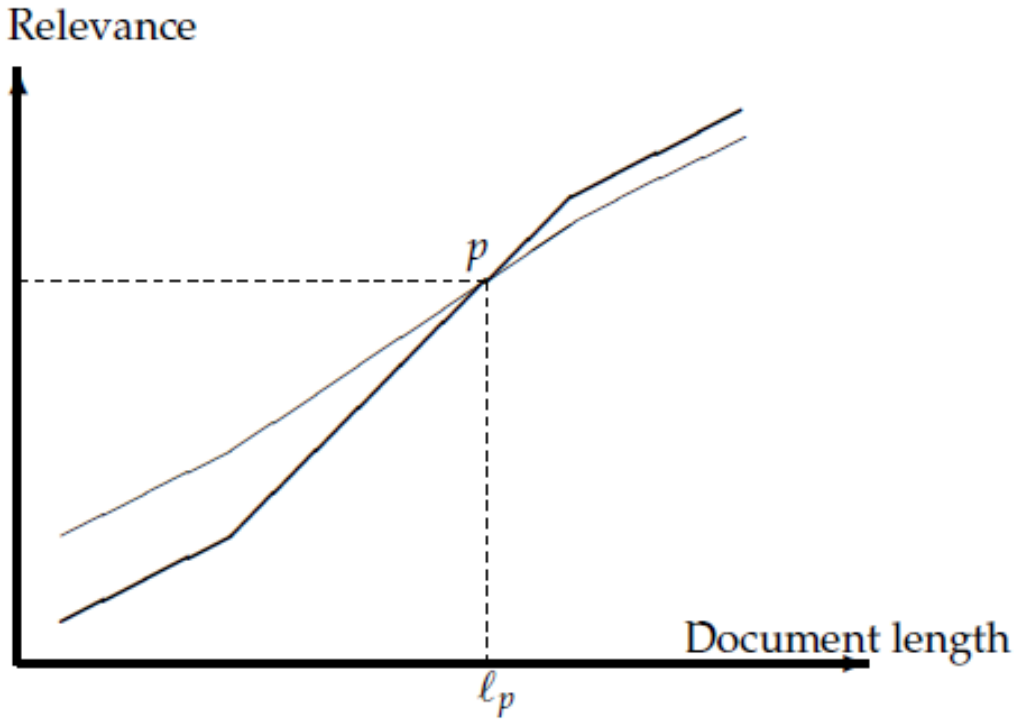
- Τα μεγάλα έγγραφα μπορεί να ανήκουν στις επόμενες κατηγορίες:
  - Verbose – επαναλαμβάνουν το ίδιο περιεχόμενο συνεχώς
  - Έγγραφα που καλύπτουν πολλαπλά διαφορετικά αντικείμενα – οι όροι που αναζητούμε ταιριάζουν με μικρά τμήματα

The Vector Space Model

- Η τεχνική pivoted document length normalization υιοθετείται ώστε να λάβουμε υπόψιν μας κανονικοποιημένα έγγραφα
- Έστω μια συλλογή εγγράφων και μια συλλογή ερωτημάτων για αυτά τα έγγραφα
- Βασιζόμενοι στη συσχέτιση ερωτημάτων – εγγράφων μπορούμε να εξάγουμε την πιθανότητα της συσχέτισης (probability of relevance) ως συνάρτηση του μήκους των εγγράφων και ως μέσος όρος για όλα τα ερωτήματα
- Τοποθετούμε τα έγγραφα σε buckets ως προς το μήκος τους και υπολογίζουμε το ποσοστό των σχετικών εγγράφων σε κάθε bucket
- Στη συνέχεια απεικονίζουμε το ποσοστό αυτό ως προς τη διάμεσο του κάθε bucket

The Vector Space Model

- Το μήκος του κάθε εγγράφου απεικονίζεται με τη μεταβλητή l_p και ονομάζεται pivot length



Έντονη γραμμή: probability of relevance

Αχνή γραμμή: relevance based on cosine similarity

The Vector Space Model

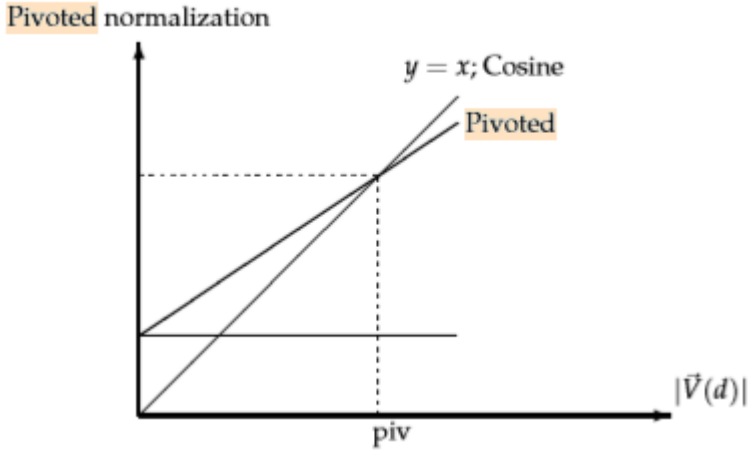
- Η pivoted length normalization τιμή υπολογίζεται ως εξής:

$$a|\vec{V}(d)| + (1-a)\text{piv}$$

- piv είναι η τιμή στην οποία οι δύο ευθείες τέμνονται
- Γενικά, η τελική εξίσωση είναι:

$$au_d + (1-a)\text{piv}$$

- Το u_d είναι το πλήθος των μοναδικών όρων στο έγγραφο



The Vector Space Model

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log\frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0,\log\frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases}1 & \text{if } \mathrm{tf}_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$