Introduction to

# Information Retrieval

ΠΛΕ70: Ανάκτηση Πληροφορίας

*Διδάσκουσα: Ευαγγελία Πιτουρά*

Διάλεξη 6: Βαθμολόγηση. Στάθμιση όρων. Το μοντέλο διανυσματικού χώρου.

1



## Τι (άλλο) θα δούμε σήμερα;

- Βαθμολόγηση και κατάταξη εγγράφων
- Στάθμιση όρων (term weighting)
- Αναπαράσταση εγγράφων και ερωτημάτων ως διανύσματα

2

# Κατάταξη εγγράφων (Ranked retrieval)

- Μέχρι τώρα, τα ερωτήματα που είδαμε ήταν Boolean.
  - Τα έγγραφα ήταν ταίριαζαν, είτε όχι
- Κατάλληλη για ειδικούς με σαφή κατανόηση των αναγκών τους και της συλλογής
  - Επίσης, καλή για εφαρμογές: οι εφαρμογές μπορούν να επεξεργαστούν χιλιάδες αποτελεσμάτων.
- Αλλά, όχι κατάλληλη για την πλειοψηφία των χρηστών
  - Είναι δύσκολο για τους περισσότερους χρήστες να διατυπώσουν Boolean ερωτήματα
  - Οι περισσότεροι χρήστες δεν θέλουν να διαχειριστούν 1000s αποτελεσμάτων.
    - Ιδιαίτερα στην περίπτωση των αναζητήσεων στο web





# Το πρόβλημα της Boolean αναζήτησης: feast or famine

- Τα Boolean ερωτήματα συχνά έχουν είτε πολύ λίγα (=0) είτε πάρα πολλά (1000s) αποτελέσματα.
- Ερώτημα 1: "*standard user dlink 650*" → 200,000 hits
- Ερώτημα 2: "*standard user dlink 650 no card found*": 0 hits
- Χρειάζεται επιδεξιότητα για να διατυπωθεί μια ερώτηση που έχει ως αποτέλεσμα ένα διαχειρίσιμο αριθμό ταιριασμάτων
  - AND πολύ λίγα - OR πάρα πολλά

## Μοντέλα διαβαθμισμένης ανάκτησης

- Αντί ενός **συνόλου** εγγράφων που ικανοποιούν το ερώτημα, η διαβαθμισμένη ανάκτηση (ranked retrieval) επιστρέφει μια **διάταξη** των (κορυφαίων) για την ερώτηση εγγράφων της συλλογής
- Ερωτήματα ελεύθερου κειμένου (Free text queries): Αντί για μια γλώσσα ερωτημάτων με τελεστές και εκφράσεις, συνήθως το ερώτημα είναι μία ή περισσότερες λέξεις σε μια φυσική γλώσσα





## Το πρόβλημα «Feast or famine» δεν υφίσταται πια

- Όταν το σύστημα παράγει ένα διατεταγμένο σύνολο αποτελεσμάτων, τα μεγάλα σύνολα δεν αποτελούν πρόβλημα
  - Δείχνουμε απλώς τα κορυφαία (top) $k$ ( ≈ 10) αποτελέσματα
  - Δεν παραφορτώνουμε το χρήστη

Προϋπόθεση: ο αλγόριθμος διάταξης δουλεύει σωστά

# Βαθμολόγηση ως η βάση της διαβαθμισμένης ανάκτησης

- Θέλουμε να επιστρέψουμε τα αποτελέσματα διατεταγμένα με βάση *το πόσο πιθανό είναι να είναι χρήσιμα στο χρήστη*
- Πως διατάσουμε-διαβαθμίζουμε τα έγγραφα μιας συλλογής με βάση ένα ερώτημα
  - Αναθέτουμε ένα βαθμό (score) – ας πούμε στο [0, 1] – σε κάθε έγγραφο
- Αυτός ο βαθμός μετρά πόσο καλά το έγγραφο "ταιριάζει" (match) με το ερώτημα





# Βαθμός ταιριάσματος ερωτήματος-εγγράφου

- Χρειαζόμαστε ένα τρόπο για να αναθέσουμε ένα βαθμό σε κάθε ζεύγος ερωτήματος(q)/εγγράφου(d)

  **score(d, q)**

- Αν ο όρος του ερωτήματος δεν εμφανίζεται στο έγγραφο, τότε ο βαθμός θα πρέπει να είναι 0
- Όσο πιο συχνά εμφανίζεται ο όρος του ερωτήματος σε ένα έγγραφο, τόσο μεγαλύτερος θα πρέπει να είναι ο βαθμός
- Θα εξετάσουμε κάποιες εναλλακτικές για αυτό

# Προσπάθεια 1: Συντελεστής Jaccard

Υπενθύμιση: συνηθισμένη μέτρηση της επικάλυψης δύο συνόλων $A$ και $B$

$$\text{jaccard}(A,B) = |A \cap B| \,/\, |A \cup B|$$

- jaccard$(A,A) = 1$
- jaccard$(A,B) = 0$ if $A \cap B = 0$

- *Τα A* και *B* δεν έχουν απαραίτητα το ίδιο μέγεθος
- Πάντα αναθέτει έναν αριθμό μεταξύ του 0 και του 1





# Συντελεστής Jaccard: Παράδειγμα βαθμολόγησης

- Ποιος είναι ο βαθμός ταιριάσματος ερωτήματος-εγγράφου με βάση το συντελεστή Jaccard για τα παρακάτω;
  - Ερώτημα (q): *ides of march*
  - Έγγραφο 1 (d1): *caesar died in march*
  - Έγγραφο 2 (d2): *the long march*

## Προβλήματα με τη βαθμολογία με Jaccard

- Δεν λαμβάνει υπ' όψιν την *συχνότητα όρου* (*term frequency*): πόσες φορές εμφανίζεται ο όρος στο έγγραφο
- Αγνοεί το γεγονός πως οι *σπάνιοι όροι* περιέχουν περισσότερη πληροφορία από ό,τι οι συχνοί.

- Ποιο πλήρη τρόπο κανονικοποιήσης του μήκους:

$$|A \cap B| / \sqrt{|A \cup B|}$$





## Δυαδική μήτρα σύμπτωσης (binary term-document incidence matrix)

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 1 | 1 | 0 | 0 | 0 | 1 |
| Brutus | 1 | 1 | 0 | 1 | 0 | 0 |
| Caesar | 1 | 1 | 0 | 1 | 1 | 1 |
| Calpurnia | 0 | 1 | 0 | 0 | 0 | 0 |
| Cleopatra | 1 | 0 | 0 | 0 | 0 | 0 |
| mercy | 1 | 0 | 1 | 1 | 1 | 1 |
| worser | 1 | 0 | 1 | 1 | 1 | 0 |

Κάθε έγγραφο αναπαρίσταται ως ένα δυαδικό διάνυσμα $\in \{0,1\}^{|V|}$(την αντίστοιχη στήλη)

# Ο πίνακας με μετρητές

- Θεωρούμε τον αριθμό (πλήθος) των εμφανίσεων ενός όρου σε ένα έγγραφο:
  - Κάθε έγγραφο είναι ένα διάνυσμα μετρητών στο $\mathbb{N}^{|v|}$: μια στήλη παρακάτω

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 157 | 73 | 0 | 0 | 0 | 0 |
| Brutus | 4 | 157 | 0 | 1 | 0 | 0 |
| Caesar | 232 | 227 | 0 | 2 | 1 | 1 |
| Calpurnia | 0 | 10 | 0 | 0 | 0 | 0 |
| Cleopatra | 57 | 0 | 0 | 0 | 0 | 0 |
| mercy | 2 | 0 | 3 | 5 | 5 | 1 |
| worser | 2 | 0 | 1 | 1 | 1 | 0 |





# *Bag of words* model

- Η διανυσματική αναπαράσταση δεν εξετάζει τη διάταξη των λέξεων σε ένα έγγραφο
  - *John is quicker than Mary* και
  - *Mary is quicker than John*

  Έχουν τα ίδια διανύσματα

- Αυτό λέγεται μοντέλου σάκου λέξεων (bag of words model).

## Συχνότητα όρου - Term frequency (tf)

- Η συχνότητα όρου $tf_{t,d}$ του όρου $t$ σε ένα έγγραφο $d$ ορίζεται ως αριθμός των φορών που το $t$ εμφανίζεται στο $d$.

- Θέλουμε να χρησιμοποιήσουμε το tf όταν υπολογίζουμε το βαθμό ταιριάσματος ερωτήματος-εγγράφου. Αλλά πως;
- Φτάνει μόνο η συχνότητα ..
  - Ένα έγγραφο με 10 εμφανίσεις ενός όρου είναι πιο σχετικό από ένα έγγραφο με 1 εμφάνιση του όρου .. Αλλά είναι 10 φορές πιο σχετικό;
- Η σχετικότητα (Relevance) δεν αυξάνει αναλογικά με τη συχνότητα όρου





## Στάθμιση με Log-συχνότητας

- Η στάθμιση με χρήση του λογάριθμου της συχνότητα (log frequency weight) του όρου $t$ στο $d$ είναι

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

  - 0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, etc.
- Ο βαθμός για ένα ζεύγος εγγράφου-ερωτήματος: άθροισμα όλων των κοινών όρων :

$$score = \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

  - Ο βαθμός είναι 0 όταν κανένας από τους όρους του ερωτήματος δεν εμφανίζεται στο έγγραφο

# Παράδειγμα

Ποιο είναι ο βαθμός για τα παρακάτω ζεύγη χρησιμοποιώντας jaccard και tf;

q: [information on cars]

d: "all you've ever wanted to know about cars"

q: [information on cars]

d: "information on trucks, information on planes, information on trains"

q: [red cars and red trucks]

d: "cops stop red cars more often"





# Συχνότητα εγγράφων (Document frequency)

Οι σπάνιοι όροι δίνουν περισσότερη πληροφορία από τους συχνούς όρους

- Θυμηθείτε τα stop words (διακοπτόμενες λέξεις)

- Θεωρείστε έναν όρο σε μια ερώτηση που είναι σπάνιος στη συλλογή (π.χ., *arachnocentric*)
- Το έγγραφο που περιέχει αυτόν τον όρο είναι πιο πιθανό να είναι πιο σχετικό με το ερώτημα από ένα έγγραφο που περιέχει ένα λιγότερο σπάνιο όρο του ερωτήματος

→ Θέλουμε να δώσουμε μεγαλύτερο βάρος στους σπάνιους όρους – αλλά πως; **df**

## Βάρος idf

- $df_t$ είναι η συχνότητα εγγράφων του $t$: ο αριθμός (πλήθος) των εγγράφων της συλλογής που περιέχουν το $t$
  - $df_t$ είναι η αντίστροφη μέτρηση της πληροφορίας που παρέχει ο όρος $t$
  - $df_t \leq N$
- Ορίζουμε την αντίστροφη συχνότητα εγγράφων idf (inverse document frequency) του $t$ ως

$$idf_t = \log_{10}(N/df_t)$$

  - Χρησιμοποιούμε log $(N/df_t)$ αντί για $N/df_t$ για να «ομαλοποιήσουμε» την επίδραση του idf.

19



## Παράδειγμα idf, έστω $N$ = 1 εκατομμύριο

| term | $df_t$ | $idf_t$ |
|---|---|---|
| calpurnia | 1 | 6 |
| animal | 100 | 4 |
| sunday | 1,000 | 3 |
| fly | 10,000 | 2 |
| under | 100,000 | 1 |
| the | 1,000,000 | 0 |

$$idf_t = \log_{10}(N/df_t)$$

✓ Κάθε όρος στη συλλογή έχει μια τιμή idf

20

## Η επίδραση του idf στη διάταξη

- Το idf δεν επηρεάζει τη διάταξη ερωτημάτων με ένα όρο, όπως
  - iPhone
- Το idf επηρεάζει μόνο τη διάταξη εγγράφων με τουλάχιστον δύο όρους
  - Για το ερώτημα capricious person, η idf στάθμιση έχει ως αποτέλεσμα οι εμφανίσεις του capricious να μετράνε περισσότερο στην τελική διάταξη των εγγράφων από ότι οι εμφανίσεις του person.

(ένα έγγραφο που περιέχει μόνο το capricious είναι πιο σημαντικό από ένα που περιέχει μόνο το person)





## Συχνότητα συλλογής και εγγράφων

- Η συχνότητα συλλογής ενός όρου *t* είναι ο αριθμός των εμφανίσεων του *t* στη συλλογή, μετρώντας και τις πολλαπλές εμφανίσεις

Παράδειγμα:

| Word | Collection frequency | Document frequency |
|---|---|---|
| *insurance* | 10440 | 3997 |
| *try* | 10422 | 8760 |

- Ποια λέξη είναι καλύτερος όρος αναζήτησης (και πρέπει να έχει μεγαλύτερο βάρος)?

## Στάθμιση tf-idf

- Το tf-idf βάρος ενός όρου είναι το γινόμενο του βάρους tf και του βάρους idf.

$$w_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log_{10}(N / \mathrm{df}_t)$$

- Το *πιο γνωστό* σχήμα διαβάθμισης στην ανάκτηση πληροφορίας
  - Εναλλακτικά ονόματα: tf.idf, tf x idf
- Αυξάνει με τον αριθμό εμφανίσεων του όρου στο έγγραφο
- Αυξάνει με τη σπανιότητα του όρου στη συλλογή





## Βαθμός εγγράφου και ερώτησης

$$\mathrm{Score}(q,d) = \sum_{t \in q \cap d} \mathrm{tf.idf}_{t,d}$$

Υπάρχουν πολλές άλλες παραλλαγές

- Πως υπολογίζεται το “tf” (με ή χωρίς log)
- Αν δίνεται βάρος και στους όρους του ερωτήματος
- …

# Δυαδική μήτρα σύμπτωσης

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 1 | 1 | 0 | 0 | 0 | 1 |
| Brutus | 1 | 1 | 0 | 1 | 0 | 0 |
| Caesar | 1 | 1 | 0 | 1 | 1 | 1 |
| Calpurnia | 0 | 1 | 0 | 0 | 0 | 0 |
| Cleopatra | 1 | 0 | 0 | 0 | 0 | 0 |
| mercy | 1 | 0 | 1 | 1 | 1 | 1 |
| worser | 1 | 0 | 1 | 1 | 1 | 0 |

Κάθε έγγραφο αναπαρίσταται ως ένα δυαδικό διάνυσμα $\in \{0,1\}^{|V|}$ (την αντίστοιχη στήλη)





# Ο πίνακας με μετρητές

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 157 | 73 | 0 | 0 | 0 | 0 |
| Brutus | 4 | 157 | 0 | 1 | 0 | 0 |
| Caesar | 232 | 227 | 0 | 2 | 1 | 1 |
| Calpurnia | 0 | 10 | 0 | 0 | 0 | 0 |
| Cleopatra | 57 | 0 | 0 | 0 | 0 | 0 |
| mercy | 2 | 0 | 3 | 5 | 5 | 1 |
| worser | 2 | 0 | 1 | 1 | 1 | 0 |

Θεωρούμε τον αριθμό των εμφανίσεων ενός όρου σε ένα έγγραφο:

- Κάθε έγγραφο είναι ένα διάνυσμα μετρητών στο $\mathbb{N}^{|v|}$:

# Ο πίνακας με βάρη

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 5.25 | 3.18 | 0 | 0 | 0 | 0.35 |
| Brutus | 1.21 | 6.1 | 0 | 1 | 0 | 0 |
| Caesar | 8.59 | 2.54 | 0 | 1.51 | 0.25 | 0 |
| Calpurnia | 0 | 1.54 | 0 | 0 | 0 | 0 |
| Cleopatra | 2.85 | 0 | 0 | 0 | 0 | 0 |
| mercy | 1.51 | 0 | 1.9 | 0.12 | 5.25 | 0.88 |
| worser | 1.37 | 0 | 0.11 | 4.15 | 0.25 | 1.95 |

Θεωρούμε το tf-idf βάρος του όρου:

- Κάθε έγγραφο είναι ένα διάνυσμα tf-idf βαρών στο $R^{|v|}$

27



# Τα έγγραφα ως διανύσματα

Έχουμε ένα |V|-διάστατο διανυσματικό χώρο

- Οι όροι είναι οι άξονες αυτού του χώρου
- Τα έγγραφα είναι σημεία ή διανύσματα σε αυτόν τον χώρο

- Πολύ μεγάλη διάσταση: δεκάδες εκατομμύρια διαστάσεις στην περίπτωση της αναζήτησης στο web
- Πολύ αραιά διανύσματα – οι περισσότεροι όροι είναι 0

28

## Τα ερωτήματα ως διανύσματα

- Βασική ιδέα 1: Εφαρμόζουμε το ίδιο και για τα ερωτήματα, δηλαδή, *αναπαριστούμε και τα ερωτήματα ως διανύσματα στον ίδιο χώρο*
- Βασική ιδέα 2: Διαβάθμιση των εγγράφων με βάση το πόσο κοντά είναι στην ερώτηση σε αυτό το χώρο
  - Κοντινά = ομοιότητα διανυσμάτων
  - Ομοιότητα ≈ αντίθετο της απόστασης





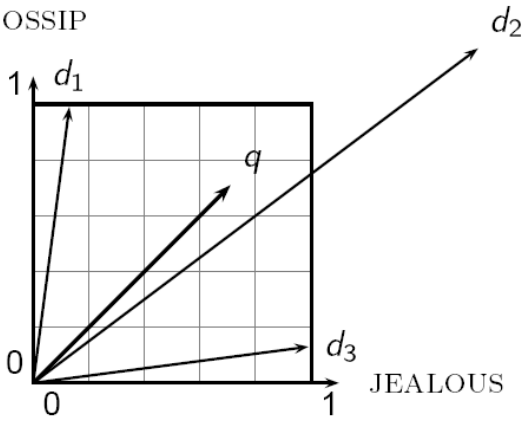
## Ομοιότητα διανυσμάτων

Πρώτη προσέγγιση: απόσταση μεταξύ δυο διανυσμάτων

- Ευκλείδεια απόσταση;
  - Δεν είναι καλή ιδέα – είναι μεγάλη για διανύσματα διαφορετικού μήκους

# Γιατί η απόσταση δεν είναι καλή ιδέα

Η Ευκλείδεια απόσταση μεταξύ του $\vec{q}$ και του $\vec{d_2}$ είναι μεγάλη αν και η κατανομή των όρων είναι παρόμοια








# Χρήση της γωνίας αντί της απόστασης

- Έστω ένα έγγραφο d. Ως άσκηση, υποθέστε ότι κάνουμε append το d στον εαυτό του και έστω d' το κείμενο που προκύπτει.
- "Σημασιολογικά" το d και το d' έχουν το ίδιο περιεχόμενο
- Η Ευκλείδεια απόσταση μεταξύ τους μπορεί να είναι πολύ μεγάλη
- Η γωνία όμως είναι 0 (αντιστοιχεί στη μεγαλύτερη ομοιότητα) => χρήση της γωνίας

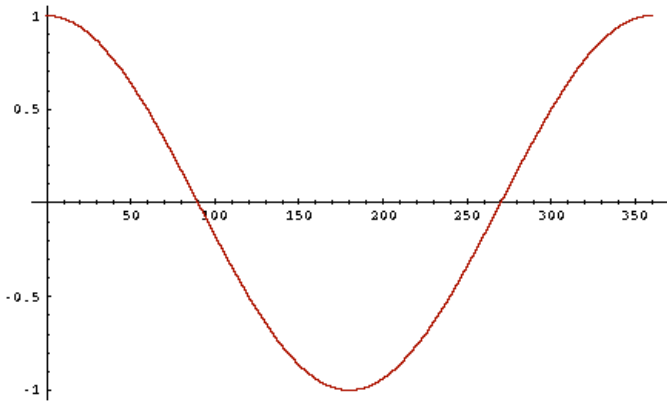
# Από γωνίες σε συνημίτονα

- Οι παρακάτω έννοιες είναι ισοδύναμες:
  - Διαβάθμιση των εγγράφων σε φθίνουσα διάταξη με βάση τη γωνία μεταξύ του εγγράφου και του ερωτήματος
  - Διαβάθμιση των εγγράφων σε αύξουσα διάταξη με βάση το συνημίτονο της γωνίας μεταξύ του εγγράφου και του ερωτήματος
- Συνημίτονο φθίνουσα συνάρτηση στο διάστημα [$0^o$, $180^o$]





# Από γωνίες σε συνιμήτονα



## cosine(query,document)

Dot product Unit vectors

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\bullet\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

$q_i$ είναι το tf-idf βάρος του όρου $i$ στην ερώτηση
$d_i$ είναι το tf-idf βάρος του όρου $i$ στο έγγραφο

$\cos(\vec{q},\vec{d})$ is the cosine similarity of $\vec{q}$ and $\vec{d}$ … or, equivalently, the cosine of the angle between $\vec{q}$ and $\vec{d}$.

35



## Κανονικοποίηση του μήκους

- Ένα διάνυσμα μπορεί να κανονικοποιηθεί διαιρώντας τα στοιχεία του α με το μήκος του, με χρήση της $L_2$ νόρμας:

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$$

- Διαιρώντας ένα διάνυσμα με την $L_2$ νόρμα το κάνει μοναδιαίο
- Για παράδειγμα το d and d’ (d και μετά d) έχουν τα ίδια διανύσματα μετά την κανονικοποίηση μήκους
  - *Ως αποτέλεσμα, μικρά και μεγάλα έγγραφα έχουν συγκρίσιμα βάρη*

36

## Συνημίτονο για κανονικοποιημένα διανύσματα

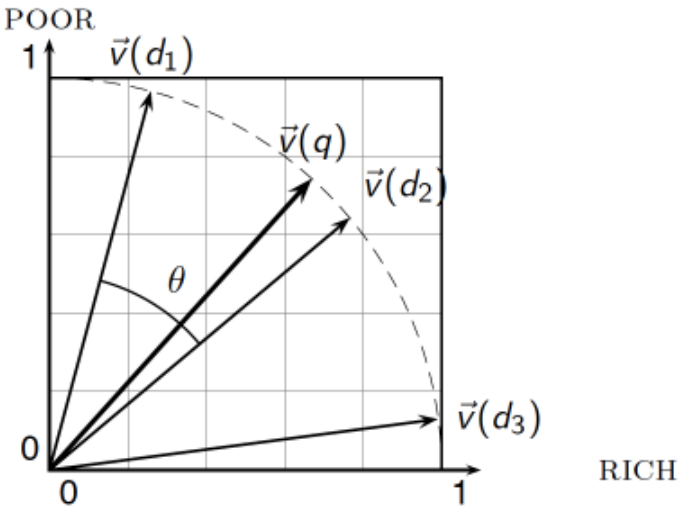
- Για διανύσματα που έχουμε κανονικοποιήσει το μήκος τους (length-normalized vectors) το συνημίτονο είναι απλώς το εσωτερικό γινόμενο (dot or scalar product):

$$\cos(\vec{q},\vec{d}) = \vec{q} \bullet \vec{d} = \sum_{i=1}^{|V|} q_i d_i$$

37



## Ομοιότητα συνημιτόνου




38

## Παράδειγμα

Ποια είναι οι ομοιότητα μεταξύ των έργων

SaS: *Sense and Sensibility*

PaP: *Pride and Prejudice*, and

WH: *Wuthering Heights*?

| term | SaS | PaP | WH |
|---|---|---|---|
| affection | 115 | 58 | 20 |
| jealous | 10 | 7 | 11 |
| gossip | 2 | 0 | 6 |
| wuthering | 0 | 0 | 38 |

Συχνότητα όρων (μετρητές)

Για απλοποίηση δε θα χρησιμοποιήσουμε τα idf βάρη





## Παράδειγμα (συνέχεια)

**Log frequency weighting**

| term | SaS | PaP | WH |
|---|---|---|---|
| affection | 3.06 | 2.76 | 2.30 |
| jealous | 2.00 | 1.85 | 2.04 |
| gossip | 1.30 | 0 | 1.78 |
| wuthering | 0 | 0 | 2.58 |

**After length normalization**

| term | SaS | PaP | WH |
|---|---|---|---|
| affection | 0.789 | 0.832 | 0.524 |
| jealous | 0.515 | 0.555 | 0.465 |
| gossip | 0.335 | 0 | 0.405 |
| wuthering | 0 | 0 | 0.588 |

cos(SaS,PaP) ≈
0.789 × 0.832 + 0.515 × 0.555 + 0.335 × 0.0 + 0.0 × 0.0
≈ 0.94
cos(SaS,WH) ≈ 0.79
cos(PaP,WH) ≈ 0.69

Why do we have cos(SaS,PaP) > cos(SaS,WH)?

# Computing cosine scores

```
CosineScore(q)
 1  float Scores[N] = 0
 2  float Length[N]
 3  for each query term t
 4  do calculate w_{t,q} and fetch postings list for t
 5     for each pair(d, tf_{t,d}) in postings list
 6     do Scores[d]+ = w_{t,d} × w_{t,q}
 7  Read the array Length
 8  for each d
 9  do Scores[d] = Scores[d]/Length[d]
10  return Top K components of Scores[]
```



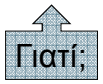


# Παραλλαγές της tf-idf στάθμισης

| 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) | $\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$ | | | | |

## Στάθμιση ερωτημάτων και εγγράφων

- Πολλές μηχανές αναζήτησης σταθμίζουνε διαφορετικά τις ερωτήσεις από τα έγγραφα
- Συμβολισμό: *ddd.qqq,* με χρήση των ακρονύμων του πίνακα
- Συχνό σχήμα : lnc.ltc
- Έγγραφο: logarithmic tf (l as first character), no idf, cosine normalization




- Ερώτημα: logarithmic tf (l in leftmost column), idf (t στη δεύτερη στήλη), no normalization …





## Παράδειγμα: lnc.ltc

Έγγραφο: *car insurance auto insurance*
Ερώτημα: *best car insurance*

| Term | Query | | | | | | Document | | | | Prod |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | tf-raw | tf-wt | df | idf | wt | n'lize | tf-raw | tf-wt | wt | n'lize | |
| auto | 0 | 0 | 5000 | 2.3 | 0 | 0 | 1 | 1 | 1 | 0.52 | 0 |
| best | 1 | 1 | 50000 | 1.3 | 1.3 | 0.34 | 0 | 0 | 0 | 0 | 0 |
| car | 1 | 1 | 10000 | 2.0 | 2.0 | 0.52 | 1 | 1 | 1 | 0.52 | 0.27 |
| insurance | 1 | 1 | 1000 | 3.0 | 3.0 | 0.78 | 2 | 1.3 | 1.3 | 0.68 | 0.53 |

Doc length $= \sqrt{1^2 + 0^2 + 1^2 + 1.3^2} \approx 1.92$

Score = 0+0+0.27+0.53 = 0.8

# Περίληψη βαθμολόγησης στο διανυσματικό χώρο

- Αναπαράσταση του ερωτήματος ως ένα διαβαθμισμένο tf-idf διάνυσμα
- Αναπαράσταση κάθε εγγράφου ως ένα διαβαθμισμένο tf-idf διάνυσμα
- Υπολόγισε το συνημίτονο για κάθε ζεύγος ερωτήματος, εγγράφου
- Διάταξε τα έγγραφα με βάση αυτό το βαθμό
- Επέστρεψε τα κορυφαία K (π.χ., K =10) έγγραφα στο χρήστη





ΤΕΛΟΣ 5ου 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*