Ανάκληση Πληροφορίας

Information Retrieval

Διδάσκων –
Δημήτριος Κατσαρός

Διάλεξη 9η

Το μειονέκτημα της Boolean ανάκτησης

- Τα Boolean ερωτήματα δίνουν είτε πολύ λίγα αποτελέσματα (=0) είτε πάρα πολλά (1000s)
  - Query 1: "*standard user dlink 650*" → 200,000 αποτελέσματα
  - Query 2: "*standard user dlink 650 no card found*": 0 αποτελέσματα
- Απαιτείται μεγάλη δεξιότητα και εμπειρία για να φτιάξεις ένα ερώτημα που παράγει διαχειρίσιμο αριθμό αποτελεσμάτων
  - Το AND δίνει λίγα και το OR δίνει πολλά αποτελέσματα

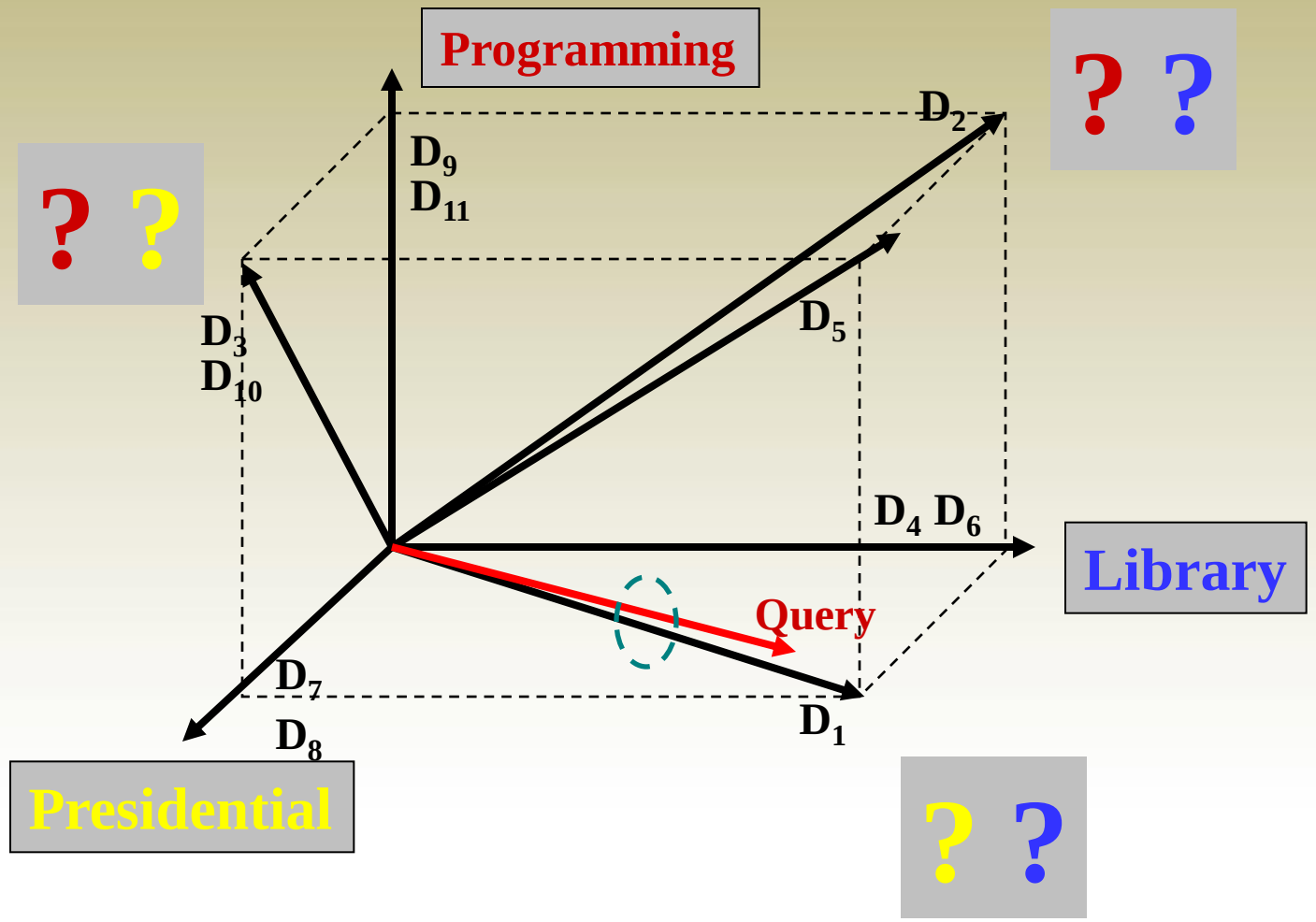
Μοντέλα διαβαθμισμένης ανάκτησης

- Αντί για ένα σύνολο εγγράφων που ικανοποιεί απακριβώς ένα ερώτημα, στα ranked retrieval models, το σύστημα επιστρέφει μια διάταξη (διαβάθμιση) των (top) εγγράφων της συλλογής σε σχέση με το ερώτημα
- Free text queries
  - Αντί για μια γλώσσα ερωτημάτων με τελεστές και εκφράσεις,
  - το ερώτημα του χρήστη αποτελείται απλά μια ή περισσότερες λέξεις κάποιας ανθρώπινης γλώσσας
- Στην πράξη τα ranked retrieval models έχουν συσχετιστεί με τα free text ερωτήματα, και αντιστρόφως

Το σκορ ως βάση της διαβαθμισμένης ανάκτησης

- Θα επιθυμούσαμε να επιστρέψουμε τα έγγραφα σε σειρά που να είναι πιο χρήσιμα στον χρήστη
- Πώς θα μπορούσαμε να το επιτύχουμε;
- Με την ανάθεση ενός score – π.χ., στο διάστημα τιμών [0, 1] – για κάθε έγγραφο
- Αυτό το score μετρά πόσο καλά “ταιριάζει” το έγγραφο με το ερώτημα

Μοντέλο Vector Space Retrieval



Υιοθέτηση του Vector Space μοντέλου

- Το VSM είναι στην πραγματικότητα ένα πλαίσιο (framework) και απαιτείται:
  - Ορισμός των διαστάσεων
  - Απόφαση για το πού θα τοποθετήσουμε τα έγγραφα και τα ερωτήματα ως διανύσματα στον διανυσματικό χώρο (δηλαδή το μήκος τους σε κάθε διάσταση)
  - Ορισμός της ομοιότητας μεταξύ δυο διανυσμάτων

Αναπαράσταση με bit vectors & ομοιότητα

$q = (x_1, \ldots, x_N)$

$d = (y_1, \ldots, y_N)$

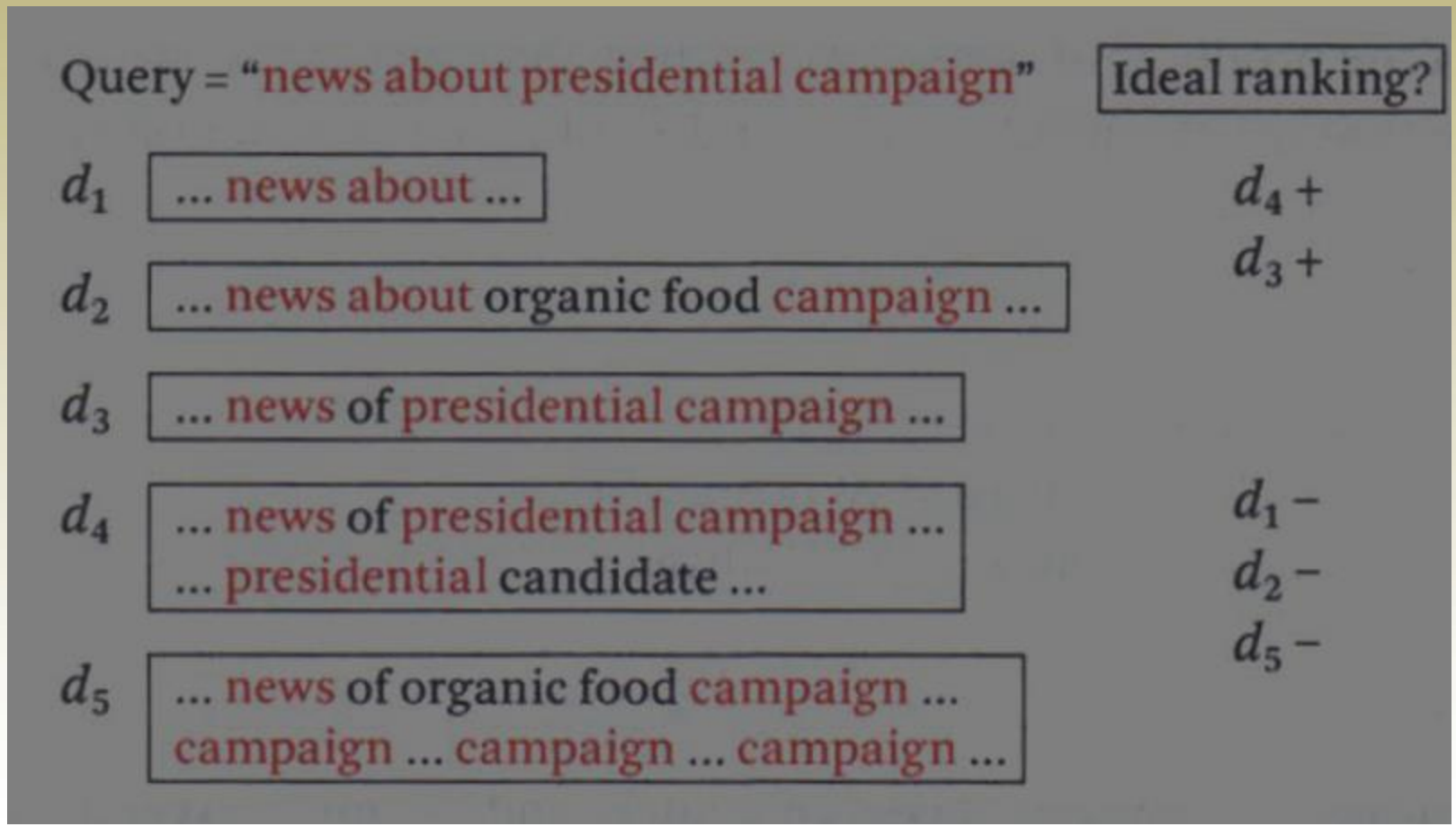
$x_i, y_i \in \{0, 1\}$

1: word W_i is **present**

0: word W_i is **absent**

$$Sim(q, d) = q.d = x_1 y_1 + \cdots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

Παράδειγμα εφαρμογής των bit vectors



Υπολογισμός της ομοιότητας

Query = "news about presidential campaign"

d_1 ... news about ...

d_3 ... news of presidential campaign ...

V =	{news,	about,	presidential,	campaign,	food	...}
q =	(1,	1,	1,	1,	0,	...)
d_1 =	(1,	1,	0,	0,	0,	...)

$f(q, d_1) = 1 * 1 + 1 * 1 + 1 * 0 + 1 * 0 + 0 * 0 + \ldots = 2$

d_3 =	(1,	0,	1,	1,	0,	...)

$f(q, d_3) = 1 * 1 + 1 * 0 + 1 * 1 + 1 * 1 + 0 * 0 + \ldots = 3$

Διαβάθμιση (ranking) των εγγράφων

Query = "news about presidential campaign"

d_1	... news about ...	$f(q, d_1) = 2$
d_2	... news about organic food campaign ...	$f(q, d_2) = 3$
d_3	... news of presidential campaign ...	$f(q, d_3) = 3$
d_4	... news of presidential campaign ... ... presidential candidate ...	$f(q, d_4) = 3$
d_5	... news of organic food campaign ... campaign ... campaign ... campaign ...	$f(q, d_5) = 2$

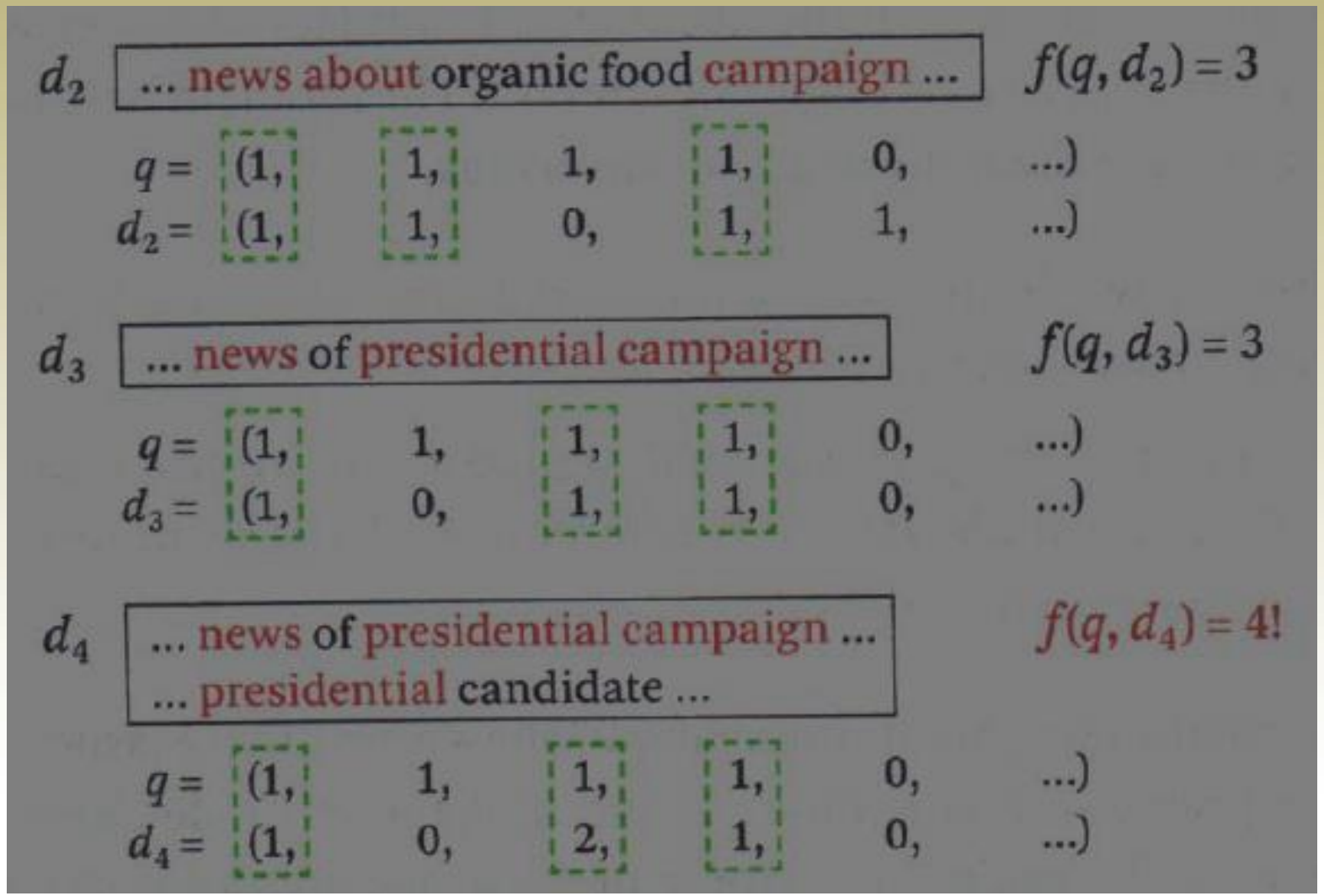
Αναπαράσταση με διάνυσμα συχνότητας

$q = (x_1, \ldots, x_N)$ x_i = count of word W_i in query

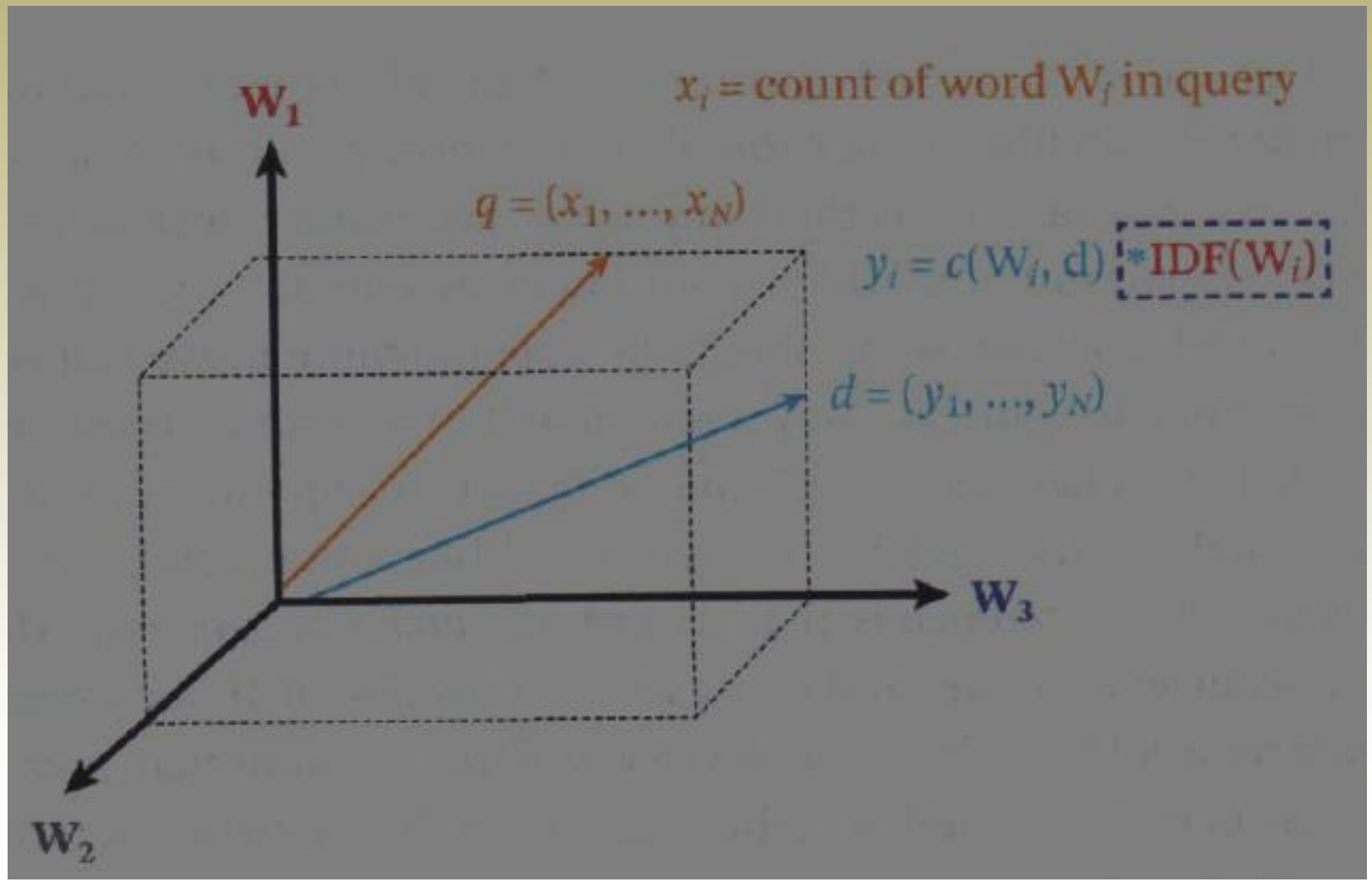
$d = (y_1, \ldots, y_N)$ y_i = count of word W_i in doc

$$Sim(q, d) = q.d = x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

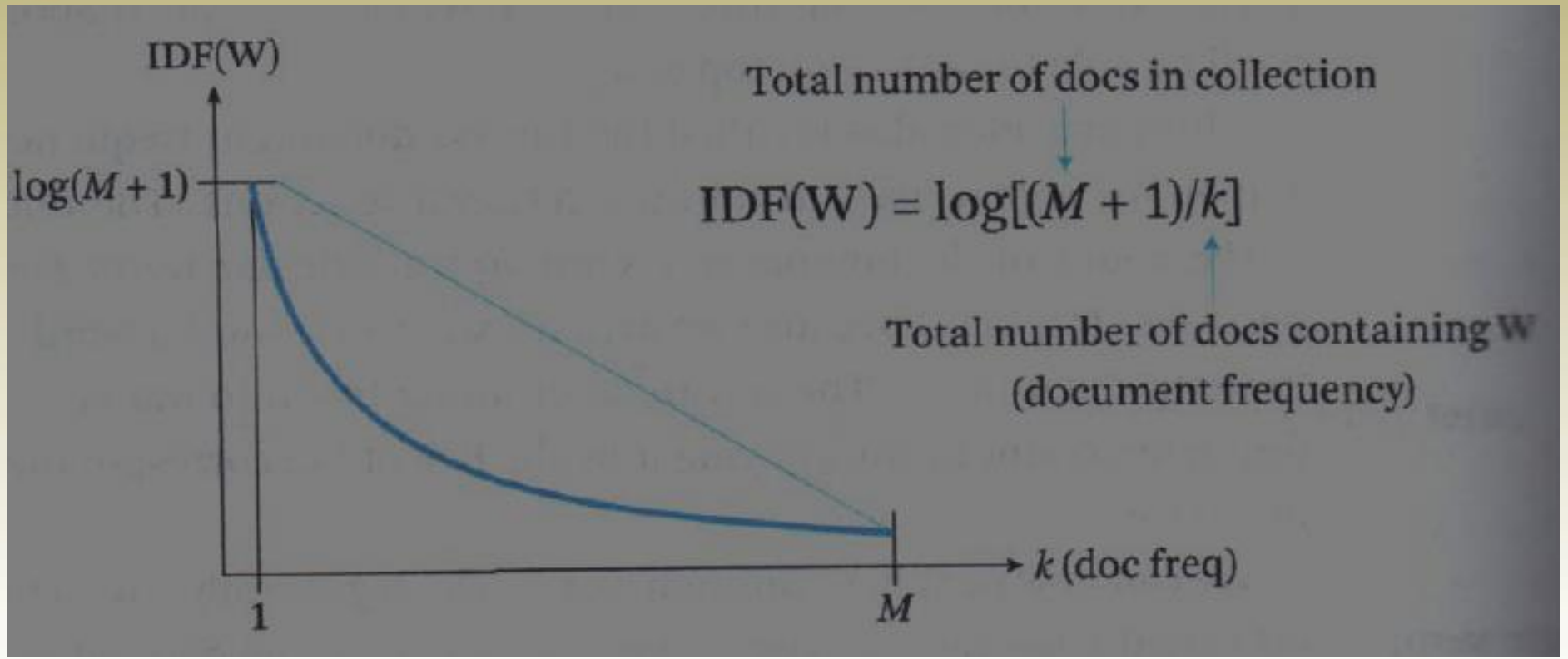
Υπολογισμός της ομοιότητας



Αναπαράσταση με TF-IDF ζύγισμα



IDF ως συνάρτηση της συχνότητας στα έγγραφα



Η επίδραση του IDF στην διαβάθμιση

d_2 ... news about organic food campaign ...

d_3 ... news of presidential campaign ...

V =	{news,	about,	presidential,	campaign,	food	...}
IDF(W) =	**1.5**	**1.0**	**2.5**	**3.1**	**1.8**	
q =	(1,	1,	1,	1,	0,	...)
d_2 =	(1 * 1.5,	1 * 1.0,	0,	1 * 3.1,	0,	...)
q =	(1,	1,	1,	1,	0,	...)
d_3 =	(1 * 1.5,	0,	1 * 2.5,	1 * 3.1,	0,	...)

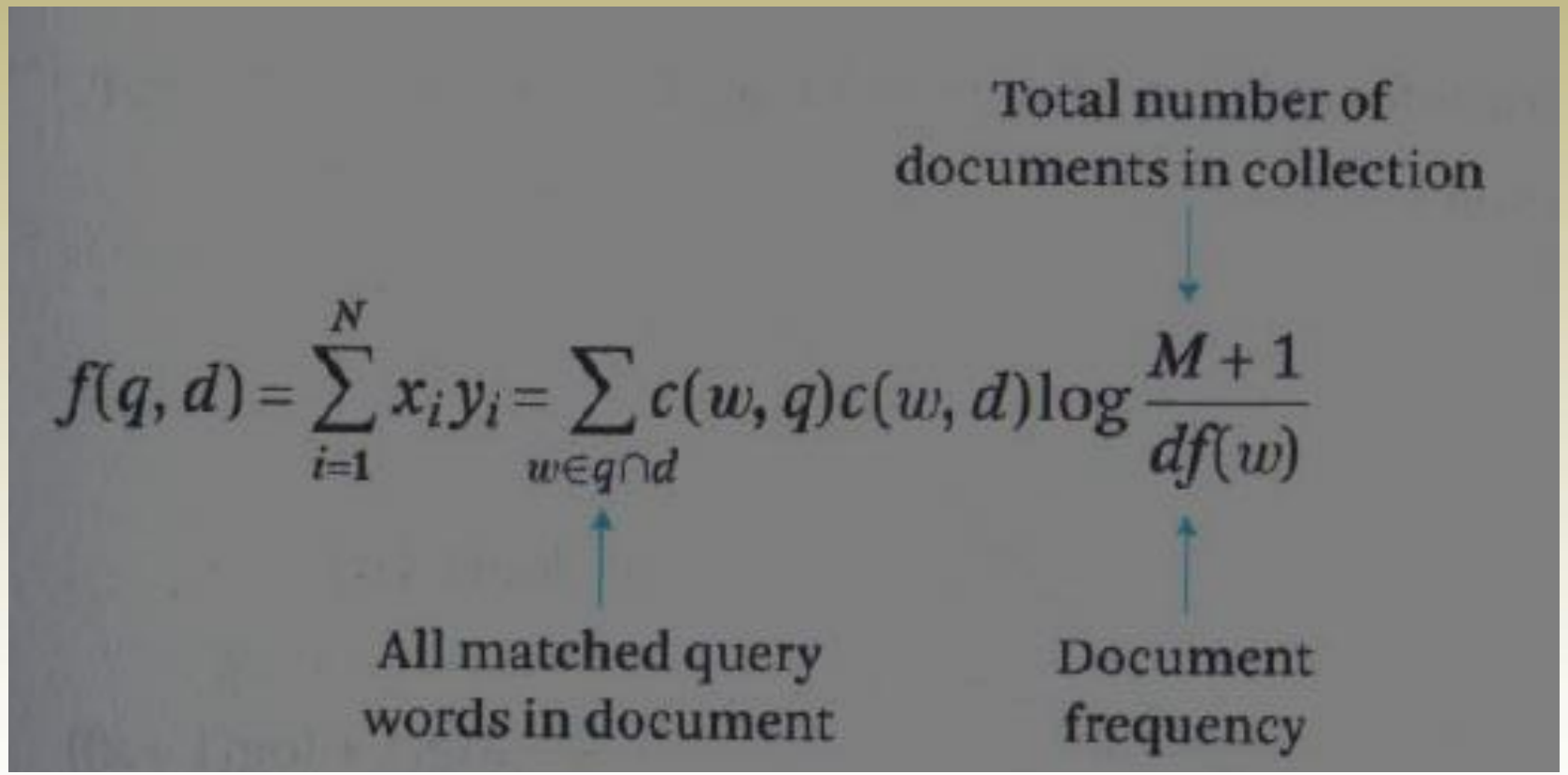
$f(q, d_2) = 5.6 \quad < \quad f(q, d_3) = 7.1$

Scores με το TF-IDF ζύγισμα

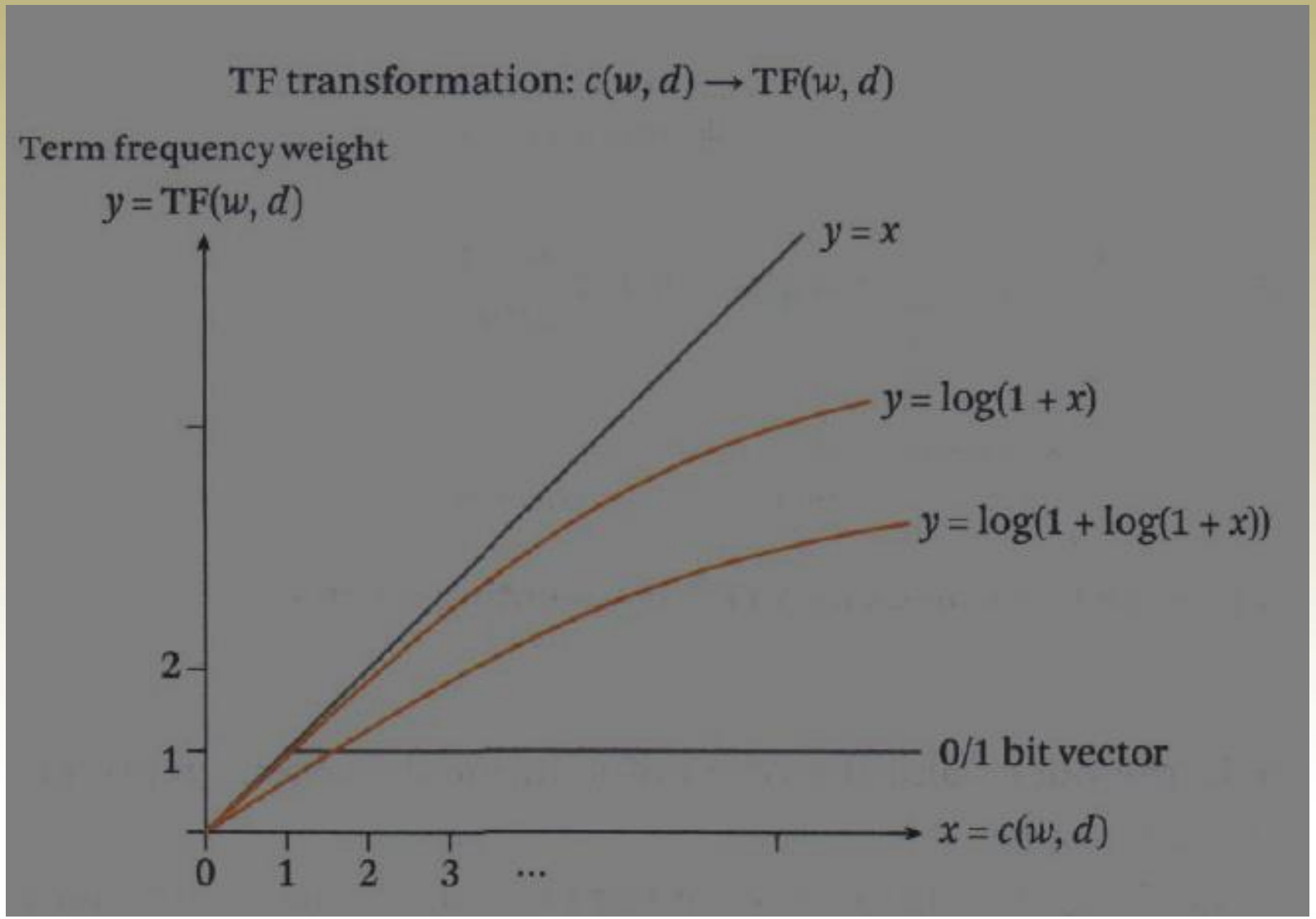
Query = "news about presidential campaign"

d_1	... news about ...	$f(q, d_1) = 2.5$
d_2	... news about organic food campaign ...	$f(q, d_2) = 5.6$
d_3	... news of presidential campaign ...	$f(q, d_3) = 7.1$
d_4	... news of presidential campaign ... ... presidential candidate ...	$f(q, d_4) = 9.6$
d_5	... news of organic food campaign ... campaign ... campaign ... campaign ...	$f(q, d_5) = 13.9!$

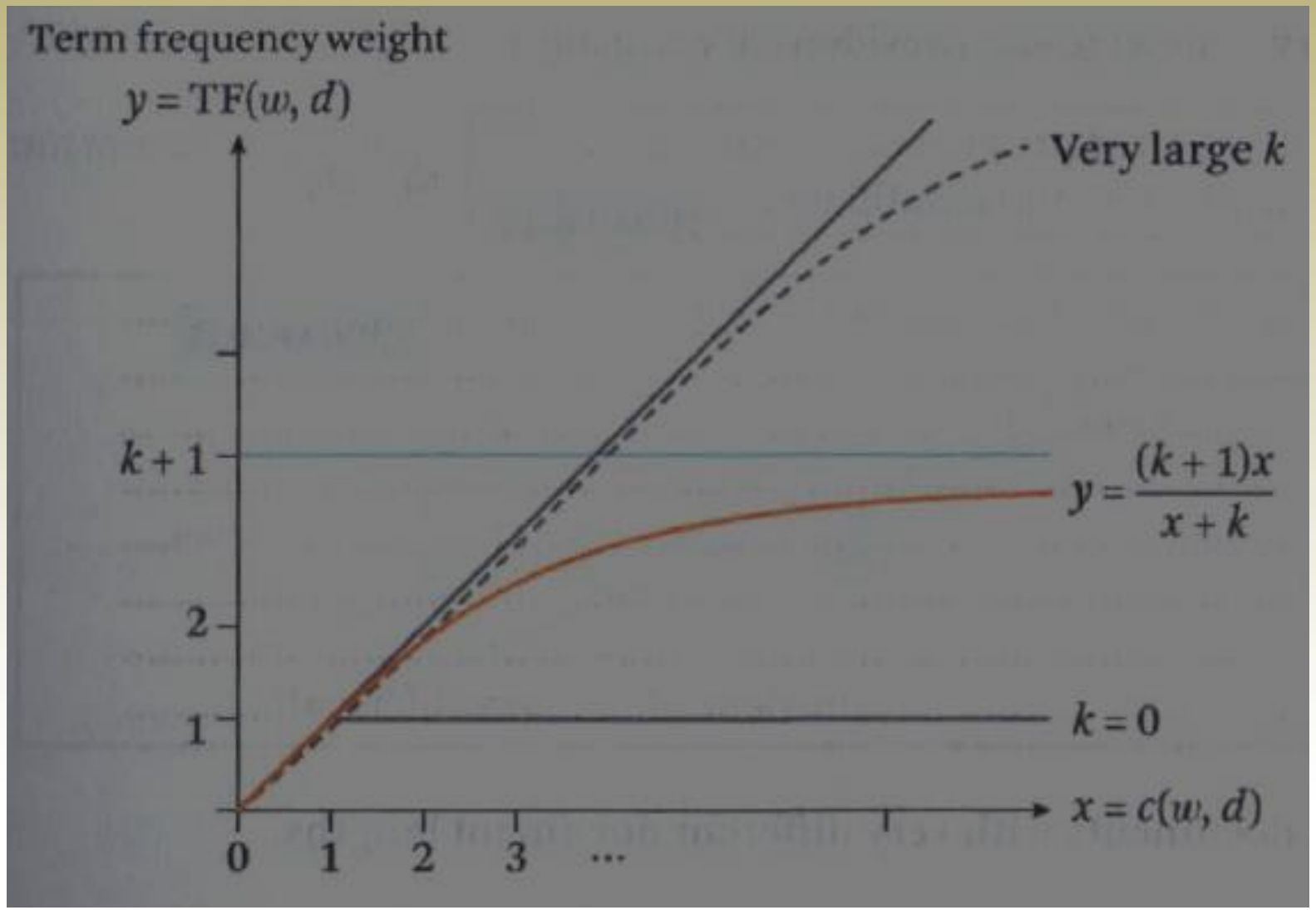
Συνάρτηση διαβάθμισης (ranking) μέχρι στιγμής



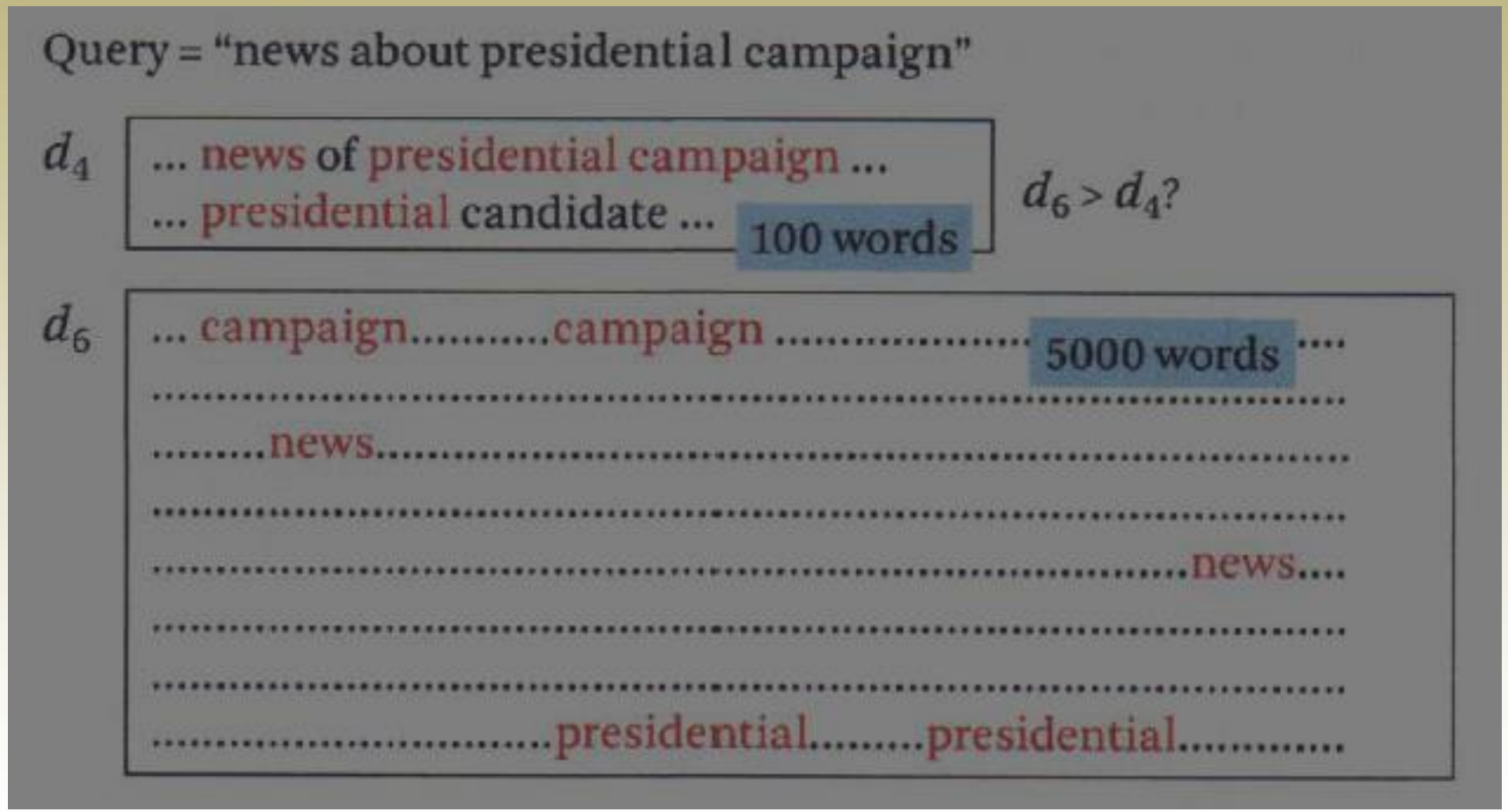
Διαφορετικοί τρόποι ορισμού του TF



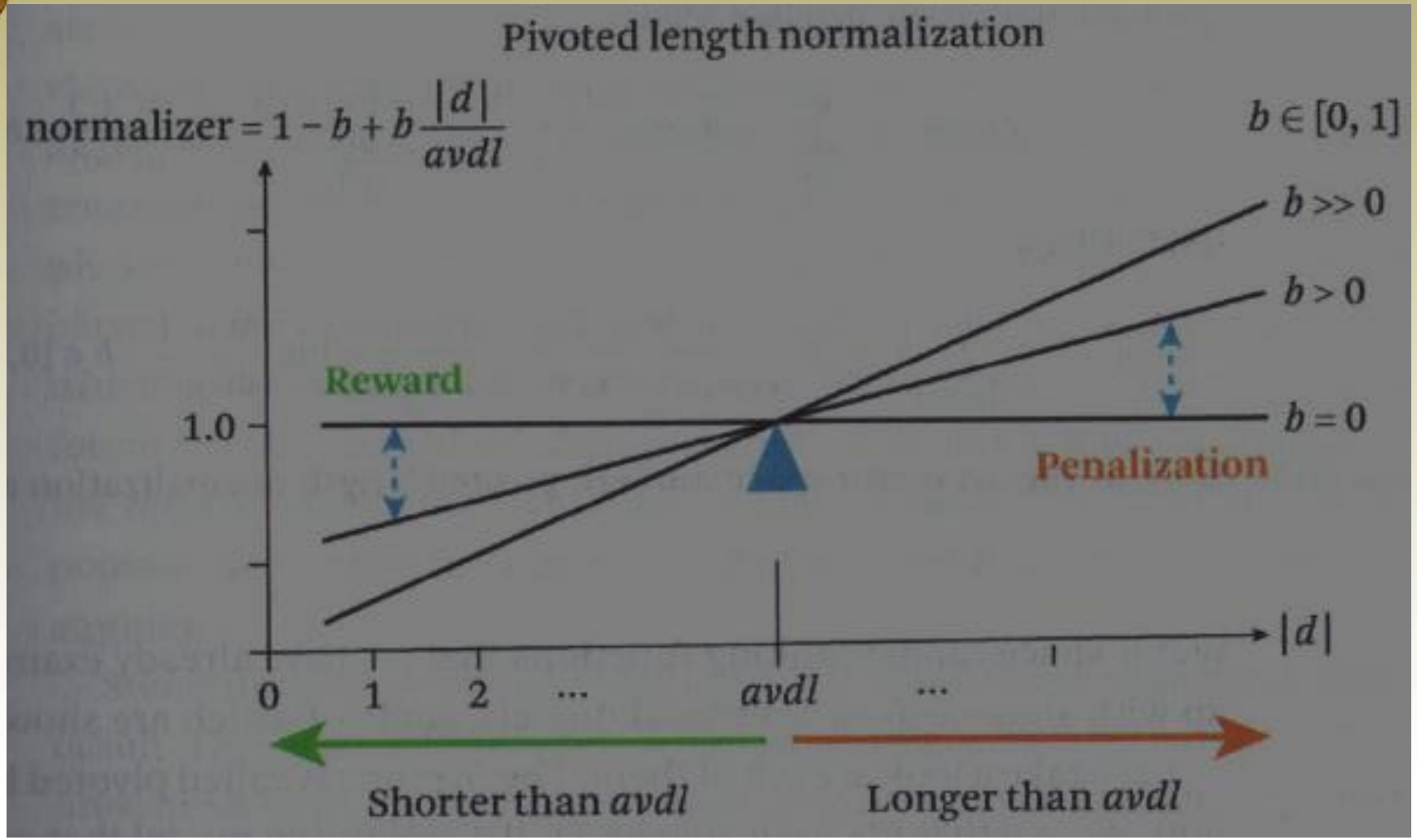
Ορισμός του BM25 ως TF



Έγγραφα με διαφορετικά μήκη



Pivoted document length κανονικοποίηση



State-of-the-art VSR μοντέλα

Pivoted length normalization VSM

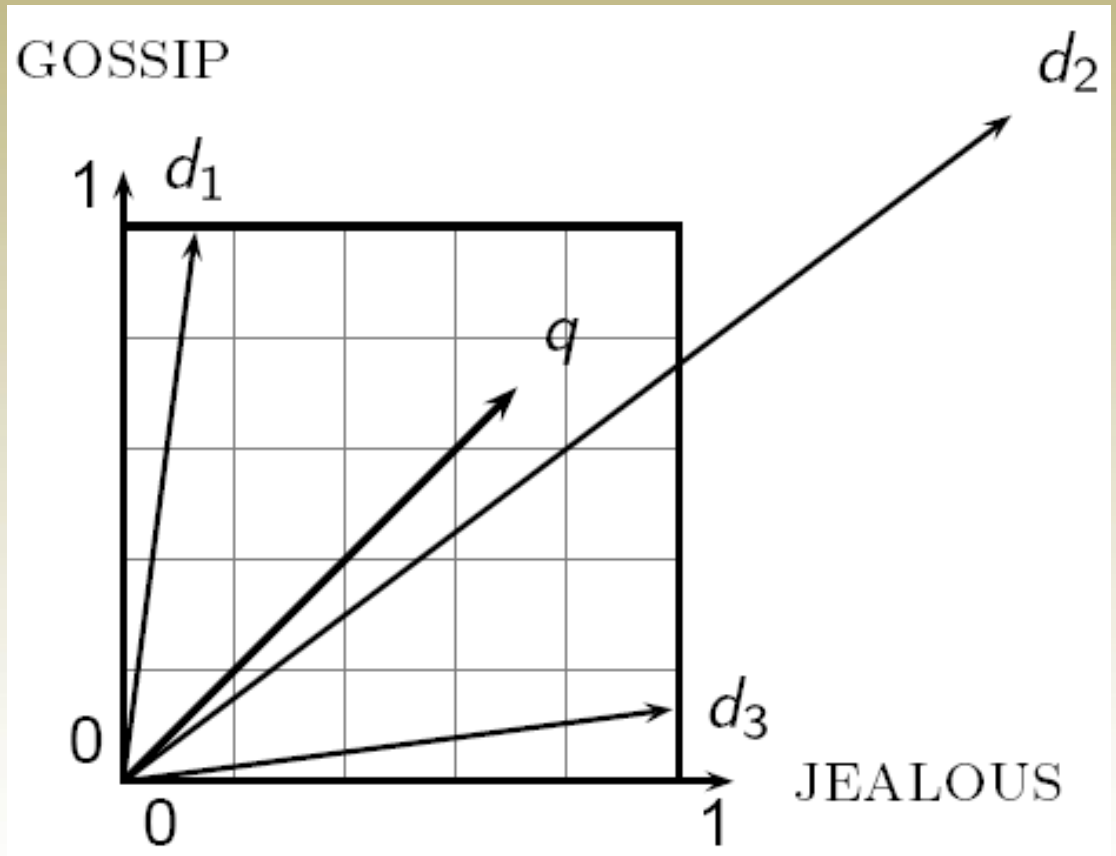
$$f(q,d)=\sum_{w\in q\cap d} c(w,q)\frac{\ln(1+\ln(1+c(w,d)))}{1-b+b\frac{|d|}{avdl}}\log\frac{M+1}{\mathrm{df}(w)} \qquad b\in[0,1]$$

BM25/Okapi

$$f(q,d)=\sum_{w\in q\cap d} c(w,q)\frac{(k+1)c(w,d)}{c(w,d)+k(1-b+b\frac{|d|}{avdl})}\log\frac{M+1}{\mathrm{df}(w)} \qquad b\in[0,1],\ k\in[0,+\infty)$$

Γιατί η απόσταση είναι κακή ιδέα

Η Ευκλείδια απόσταση μεταξύ του $\vec{q}$ και του $\vec{d_2}$ είναι μεγάλη, παρόλο που η κατανομή των όρων στο ερώτημα $\vec{q}$ και η κατανομή των όρων στο έγγραφο $\vec{d_2}$ είναι πολύ όμοια



Χρήση της γωνίας αντί της απόστασης

- Πείραμα σκέψης: πάρτε ένα έγγραφο d και append στον εαυτό του. Καλέστε αυτό το νέο έγγραφο ως d'
- "Σημασιολογικά" τα d και d′ έχουν το ίδιο περιεχόμενο
- Η Ευκλείδια απόσταση μεταξύ των δυο εγγράφων (μπορεί) να είναι αρκετά μεγάλη
- Η γωνία όμως μεταξύ των δυο εγγράφων είναι 0, δηλαδή αντιστοιχεί στην μέγιστη ομοιότητα

Από τις γωνίες στα συνημίτονα

- Οι επόμενες έννοιες είναι ισοδύναμες:
  - Διάταξη των εγγράφων σε φθίνουσα γωνία μεταξύ του ερωτήματος και του εγγράφου
  - Διάταξη των εγγράφων σε αύξουσα τιμή του συνημιτόνου cosine(query, document)

- Θυμηθείτε ότι το συνημίτονα είναι γνησίως φθίνουσα συνάρτηση στο διάστημα [0^o, 180^o]

Κανονικοποίηση μήκους

- Ένα διάνυσμα μπορεί να κανονικοποιηθεί (ως προς το μήκος) διαιρώντας κάθε μια από τις συνιστώσες σου με το μήκος του (ως μήκος τους χρησιμοποιούμε την L_2 νόρμα):

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

- Αυτή η διαίρεση οδηγεί σε μοναδιαίο διάνυσμα

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* στο ερώτημα
d_i είναι το tf-idf βάρος του όρου *i* στο έγγραφο

$\cos(\vec{q},\vec{d})$ είναι η cosine ομοιότητα των $\vec{q}$ και $\vec{d}$ … ή, ισοδύναμα, το cosine της γωνίας μεταξύ των $\vec{q}$ και $\vec{d}$

Cosine για length-normalized διανύσματα

- Για length-normalized διανύσματα, η ομοιότητα cosine είναι απλά το εσωτερικό γινόμενο (dot product) ή βαθμωτό γινόμενο (scalar product):

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

για q, d length-normalized.

28

Υπολογισμός των 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[]
```

Το σχήμα tf-idf έχει πολλές παραλλαγές

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

Οι στήλες με τίτλο ‘n’ είναι ακρωνύμια για σχήματα βάρους

Στενωποί

- Η κύρια υπολογιστική στενωπός στον υπολογισμό των scores: cosine computation
- Μπορουμε ν' αποφύγουμε αυτόν τον υπολογισμό;
- Ναι, αλλά μερικές φορές σφάλουμε:
  - Ένα doc που *δεν είναι* στα top K μπορεί να εισδύσει στην λίστα των K output docs
  - Προκαλεί αυτό δυσχέρειες;

Η ομοιότητα cosine είναι απλά μια προσέγγιση

- Ο κάθε χρήστης έχει μια δραστηριότητα και μια διατύπωση ερωτήματος
- Το cosine ταιριάζει docs στο ερώτημα
- Συνεπώς, το cosine είναι απλά μια προσέγγιση στην ικανοποίση που λαμβάνει ο χρήστης
- Εάν τού επιστραφεί (του χρήστη) μια λίστα με K docs "κοντά" στα πραγματικά top K (τα οποία θα επέστρεφε η μετρική cosine), τότε δεν θα υπάρχει σημαντικό πρόβλημα

Η γενική προσέγγιση

- Εύρεση ενός συνόλου A αποτελούμενο από *contenders*, με $K < |A| << N$
  - Το A δεν περιέχει αναγκαστικά τα top K, αλλά περιέχει πολλά docs από αυτά του top K
  - Επιστροφή των top K docs του A
- Σκεφτείτε το A ως pruning των non-contenders
- Η ίδια προσέγγιση μπορεί να χρησιμοποιηθεί και για άλλες (non-cosine) συναρτήσεις score
- Θα εξετάσουμε διάφορα σχήματα τα οποία ακολουθούν αυτήν την προσέγγιση

Μόνο οι query terms με υψηλό idf

- Για ένα ερώτημα όπως: *catcher in the rye*
- Συσσωρεύουμε μόνο τα score από τα *catcher* και *rye*
- Διαίσθηση: οι *in* και *the* συνεισφέρουν λίγο στα scores και έτσι δεν μεταβάλλουν την διάταξη σημαντικά
- Ώφελος:
  - Οι postings των of low-idf όρων έχουν πολλά → αυτά τα (πολλά) docs απαλοίφονται από το σύνολο *A* των contenders

Docs που περιέχουν πολλά query terms

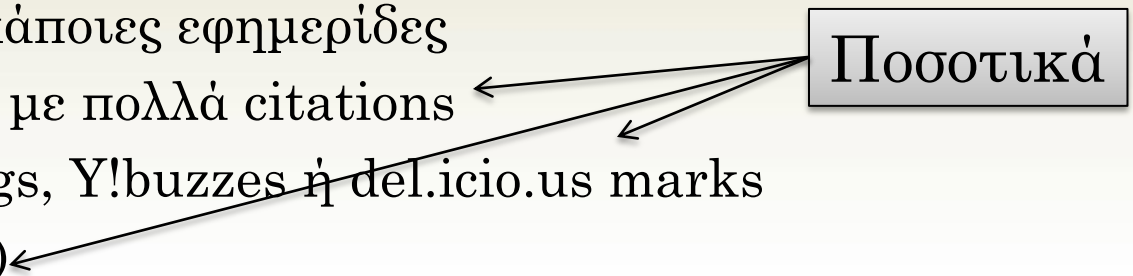
- Κάθε doc με τουλάχιστον ένα query term είναι υποψήφιο για την top K output list
- Σε ερωτήματα με πολλαπλούς όρους, υπολογίζουμε τα scores των docs τα οποιά περιέχουν "αρκετούς" από τουςquery terms
  - Π.χ., τουλάχιστον 3 από τους 4
  - Επιβάλλει ένα "soft conjunction" σε ερωτήματα που υποβάλλονται στις Web search engines
- Εύκολο στην υλοποίηση κατά την διάσχιση των postings

Champion lists

- Προϋπολογίζουμε για κάθε dictionary term *t*, τα *r* docs με το υψηλότερο στην postings list του *t*
  - Αποκαλούμε αυτό το σύνολο των r docs ως η champion list για τον *t*
  - (ή και fancy list ή top docs για τον *t*)
- Παρατηρήστε ότι το *r* πρέπει να επιλεγεί κατά την διάρκεια κατασκευής του inverted index
  - Συνεπώς, είναι πιθανόν να ισχύει ότι: $r < K$
- Όταν υποβληθεί ένα ερώτημα, υπολογίζουμε τα scores για τα docs στην champion list του query term
  - Επιλέγουμε τα *K* top-scoring docs μεταξύ αυτών

Score στατικής ποιότητας

- Ζητούμε τα top-ranking documents να είναι ταυτόχρονα *relevant* και *authoritative*
- Η *relevance* μοντελοποιείται με τα cosine scores
- Η *authority* είναι τυπικά μια ιδιότητα που δεν εξαρτάται από το ερώτημα, δηλ., query-independent
- Παραδείγματα ποσοτικοποίησης της authority
  - Wikipedia μεταξύ των websites
  - Άρθρα σε κάποιες εφημερίδες
  - Ένα άρθρο με πολλά citations
  - Πολλά diggs, Y!buzzes ή del.icio.us marks
  - (Pagerank)



Μοντελοποίηση της αυθεντίας (authority)

- Ανάθεση σε κάθε έγγραφο *d* ένα *query-independent* quality score με τιμές στο διάστημα [0,1]
  - Ας το συμβολίσουμε με *g(d)*
- Επομένως, μια ποσότητα, όπως για παράδειγμα ο αριθμός των incoming links, την κανονικοποιούμε στο διάστημα [0,1]
  - Άσκηση: Προτείνετε έναν μαθηματικό τύπο γι' αυτό

Η έννοια του net score

- Θεωρήστε ένα απλό συνολικό score το οποίο συνδυάζει την cosine relevance και την αυθεντία (authority), δηλαδή:
- **net-score(q,d) = $g(d)$ + cosine(q,d)**
  - Φυσικά, μπορούμε να χρησιμοποιήσουμε κάποιον άλλον γραμμικό συνδυασμό και όχι την ίση συνεισφορά
  - Στην πραγματικότητα, κάθε συνάρτηση των δυο "σημάτων" που ποσοτιμοποιούν την ικανοποίηση του χρήστη
- Συνεπώς, αναζητούμε τα top K docs με βάση το by net score

Top K με το net score – γρήγορες μέθοδοι

- Πρώτη ιδέα: Διάταξη όλων των postings με βάση το $g(d)$
- Το κλειδί: αυτό είναι κοινή διάταξη για όλα τα postings
- Συνεπώς, μπορούμε ταυτόχρονα να διασχίσουμε τις postings λίστες των query terms για
  - Intersection των postings
  - Υπολογισμό του cosine score

Γιατί διάταξη των postings ως προς *g(d)*;

- Με τη *g(d)*-ordering, τα top-scoring docs είναι πιθανό να εμφανίζονται νωρίς κατά την διάσχιση των postings λιστών
- Σε εφαρμογές με αυστηρές απαιτήσεις σε χρονική απόκριση (π.χ., απαιτείται να επιστρέψουμε οποιαδήποτε αποτελέσματα αναζήτηση βρήκαμε μέσα σε 50 ms), μάς επιτρέπει να σταματήσουμε την διάσχιση των postings σχετικά νωρίς

Champion lists στο *g(d)*-ordering

- Μπορούμε να συνδυάσουμε τις champion lists με το *g(d)*-ordering
- Διατηρούμε για κάθε όρο μια champion list των *r* docs με το υψηλότερο $\boldsymbol{g(d) + tf\text{-}idf_{td}}$
- Αναζητούμε τα top-*K* αποτελέσματα μόνο από αυτές τις champion lists

Η πλήρης εικόνα