Επεκτάσεις σε ένα Ολοκληρωμένο Σύστημα

Εισαγωγή

- Για τον υπολογισμό των βαρών, κρατάμε το idf για κάθε όρο του λεξικού και το tf για κάθε κόμβο των postings
- Στόχος μας είναι να υπολογίσουμε το σκορ για κάθε όρο του ερωτήματος και για κάθε αντίστοιχο κόμβο στις postings
- Θέλουμε να περιορίσουμε το πλήθος των εγγράφων στους υπολογισμούς
- Το τελικό βήμα είναι να επιλέξουμε τα Κ έγγραφα με το μεγαλύτερο σκορ
- Μπορούμε να ταξινομήσουμε τη λίστα με τα σκορ
- Καλύτερη λύση είναι να υιοθετήσουμε ένα σωρό (heap)
- Μέσω του σωρού εξάγουμε τα top-K έγγραφα
- Αν έχουμε Λ<N έγγραφα με μη μηδενικό σκορ χρειαζόμαστε 2*Λ συγκρίσεις στο σωρό
- Η εξαγωγή των Κ εγγράφων θα απαιτήσει logΛ συγκρίσεις

Ανάκτηση των top-K Εγγράφων

- Η προηγούμενη τεχνική εστιάζει στο να εξάγει τα K έγγραφα μετά τους υπολογισμούς
- Όμως μπορούμε να παράξουμε τα K έγγραφα τα οποία είναι πιθανό να είναι στα έγγραφα με το υψηλότερο σκορ για ένα ερώτημα
- Στόχος είναι να μειώσουμε το κόστος παραγωγής των K εγγράφων
- Το κύριο κόστος εστιάζεται στον υπολογισμό της ομοιότητας μεταξύ του ερωτήματος και ενός μεγάλου αριθμού εγγράφων
- Η τεχνική ονομάζεται inexact top-K document retrieval

Ανάκτηση των top-K Εγγράφων

- Γενικά βασιζόμαστε στην πεποίθηση πως πιθανώς τα ‘κανονικά’ top-K έγγραφα να μην αποτελούν τις βέλτιστες λύσεις για το ερώτημα
- Μια ιδέα είναι:
  - Βρίσκουμε ένα σύνολο A με έγγραφα τα οποία είναι ‘ανταγωνιστές’ για το ερώτημα (K<A<<N) – το σύνολο A δεν περιέχει απαραίτητα τα top-K έγγραφα με το μεγαλύτερο σκορ αλλά είναι πιθανό να περιέχει πολλά έγγραφα με σκορ κοντά στο top-K
  - Επιστρέφουμε τα top-K έγγραφα του A μετά τον υπολογισμό του σκορ

Index Elimination

- Γενικά, για ένα ερώτημα με πολλαπλούς όρους, εξετάζουμε έγγραφα που περιλαμβάνουν τουλάχιστον ένα από τους όρους
- Εφαρμόζουμε τις ακόλουθες ιδέες:
  - Εξετάζουμε έγγραφα για τα οποία το idf είναι πάνω από ένα όριο – διασχίζουμε στις postings μόνο τα έγγραφα με υψηλό idf – γενικά οι postings με χαμηλό idf τείνουν να είναι μεγαλύτερες, αν δεν τις λάβουμε υπόψιν μας στους υπολογισμούς τότε γλιτώνουμε κόστος
  - Εξετάζουμε έγγραφα τα οποία περιέχουν πολλούς από τους όρους του ερωτήματος – υπολογίζουμε το σκορ για τα έγγραφα που περιέχουν όλους ή πολλούς από τους όρους των ερωτημάτων

Champions Lists

- Δοσμένου ενός ερωτήματος, κατασκευάζουμε ένα σύνολο Α εργαζόμαστε ως εξής:
  - Σε φάση προεπεξεργασίας, για κάθε όρο υπολογίζουμε το σύνολο r εγγράφων με το μεγαλύτερο βάρος για ένα όρο
  - Αυτά είναι τα έγγραφα με το μεγαλύτερο tf
  - Αυτά τα r έγγραφα απαρτίζουν την champions list για τον κάθε όρο
- Παίρνουμε την ένωση των champions lists για όλους τους όρους και υπολογίζουμε τα σκορ για την ένωση
- Η κρίσιμη παράμετρος είναι το r
- Πρέπει να είναι αρκετά μεγάλο σε σχέση με το Κ
- Μπορούμε να έχουμε διαφορετικές τιμές για διαφορετικούς όρους

Static Quality Scores

- Σε κάποιες μηχανές αναζήτησης υπολογίζεται μια τιμή ‘ποιότητας’ των εγγράφων η οποία είναι στατική
- Η τιμή αυτή συμβολίζεται με g(d)
- Η τιμή είναι ανεξάρτητη των ερωτημάτων
- Συνήθως η τιμή είναι στο διάστημα [0,1]
- Παράδειγμα:
  - Η τιμή μπορεί να απεικονίζει τις θετικές κριτικές σε ένα δικτυακό τόπο που παρουσιάζει νέα

Static Quality Scores

- Το τελικό σκορ για ένα έγγραφο είναι ένας συνδυασμός αυτής της στατικής τιμής και μιας τιμής που εξαρτάται από το ερώτημα
- Μια απλή προσέγγιση είναι η ακόλουθη:

$$\text{net-score}(q,d) = g(d) + \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}$$

- Σε αυτή τη μέθοδο η στατική τιμή συνεισφέρει ισόποσα για τον υπολογισμό της τελικής τιμής

Static Quality Scores

- Αρχικά, παράγουμε τα έγγραφα σε φθίνουσα σειρά της g(d) για κάθε όρο σε κάθε postings list
- Παίρνουμε την τομή των λιστών
- Στη συνέχεια μπορούμε να πάρουμε μια κατάλληλη τιμή του r και για κάθε όρο να διατηρούμε μια global champions list των r εγγράφων
- Η καθολική λίστα θα περιλαμβάνει τις υψηλότερες τιμές των g(d)+tf-idf$_{t,d}$
- Όταν τίθεται το ερώτημα, απλά υπολογίζουμε το net-score για τα έγγραφα που ανήκουν στην ένωση των global champions lists

Impact Ordering

- Μέχρι στιγμής έχουμε δει πως ταξινομούμε τις postings lists είτε ως προς το ID είτε ως προς το g(d)
- Μια άλλη ιδέα είναι να μην ταξινομήσουμε τις λίστες ως προς κάποια κοινή μετρική αλλά ως προς το tf
- Προφανώς, η παράλληλη συγχώνευση των λιστών δεν είναι εύκολη υπόθεση
- Ο λόγος είναι ότι η ταξινόμηση των λιστών θα μεταβάλλεται
- Για να μειώσουμε το πλήθος των εγγράφων κατά τον υπολογισμό των τιμών έχουν προταθεί:
  - Όταν διασχίζουμε μια λίστα για ένα όρο του ερωτήματος, σταματούμε όταν επεξεργαστούμε ένα prefix της λίστας – είτε μετά από ένα πλήθος εγγράφων r ή όταν το tf ξεπεράσει ένα όριο
  - Θεωρούμε τους όρους των ερωτημάτων σε φθίνουσα σειρά idf οπότε εστιάζουμε στους όρους που είναι πιθανό να συνεισφέρουν περισσότερο

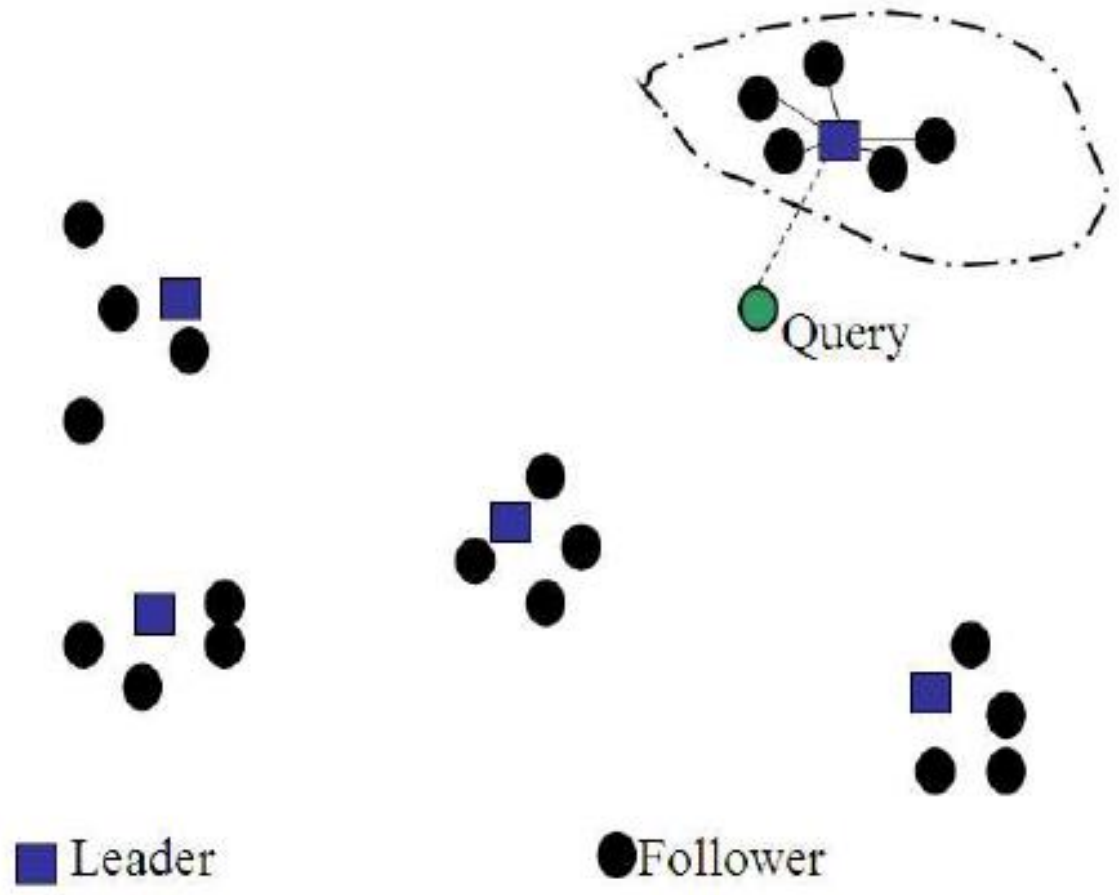
Cluster Pruning

- Υιοθετούμε ένα στάδιο προεπεξεργασίας
- Παράγουμε συστάδες (clustering) των διανυσμάτων των εγγράφων
- Όταν τίθεται το ερώτημα τότε εστιάζουμε σε ένα μικρό αριθμό συστάδων για να υπολογίσουμε τις τελικές τιμές

Cluster Pruning

- Το στάδιο της προεπεξεργασίας περιλαμβάνει:
  - Επιλέγουμε $\sqrt{N}$ τυχαία έγγραφα από τη συλλογή – καλούνται leaders
  - Για κάθε έγγραφο από τα υπόλοιπα, υπολογίζουμε τον κοντινότερο leader – τα υπόλοιπα έγγραφα ονομάζονται followers
- Για κάθε leader ο αναμενόμενος αριθμός των followers είναι $\frac{N}{\sqrt{N}} = \sqrt{N}$
- Η επεξεργασία των ερωτημάτων έχει ως εξής:
  - Δοσμένου ενός ερωτήματος q, βρίσκουμε τον leader που είναι ο κοντινότερος – υπολογίζουμε τα cosine similarities για καθένα από τους $\sqrt{N}$ leaders
  - Το σύνολο των υποψηφίων εγγράφων περιλαμβάνει τον πιο κοντινό leader καθώς και τους followers του

Cluster Pruning



Cluster Pruning

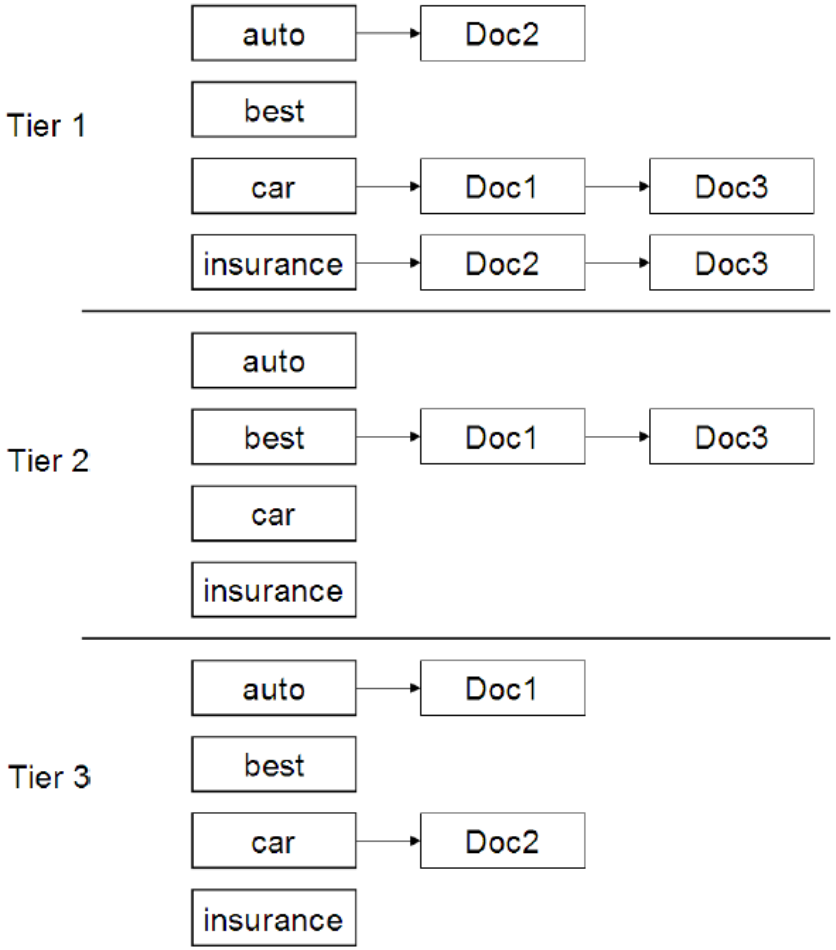
- Η τυχαία επιλογή των leaders είναι γρήγορη και αντανακλά την κατανομή των εγγράφων
- Μια παραλλαγή της μεθόδου βασίζεται σε δύο θετικές σταθερές b1, b2
- Κατά την προεπεξεργασία 'συνδέουμε' κάθε follower με τους b1 κοντινότερους leaders αντί για ένα
- Κατά την επεξεργασία του ερωτήματος, θεωρούμε τους b2 κοντινότερους leaders αντί για ένα
- Το βασικό σχήμα υποθέτει b1=b2=1
- Αν αυξήσουμε τα b1, b2, αυξάνουμε την πιθανότητα να βρούμε K έγγραφα που πιθανώς να είναι στα top-K έγγραφα
- Μεγάλες τιμές για τα b1, b2 αυξάνουν τον υπολογιστικό χρόνο

Τοποθετώντας τα όλα μαζί

Tiered Indexes

- Υιοθετώντας την τεχνική του index elimination υπάρχει η πιθανότητα να παράξουμε λιγότερα από K έγγραφα
- Μια λύση σε αυτό το πρόβλημα είναι να χρησιμοποιήσουμε tiered indexes
- Μπορούμε για το πρώτο index να υιοθετήσουμε ένα όριο tf 20 και πάνω, για το δεύτερο ένα όριο 10, κ.ο.κ.

Tiered Indexes



Terms Proximity

- Έστω ένα ερώτημα με τουλάχιστον δύο όρους
- Έστω ω είναι το μικρότερο παράθυρο (πλήθος λέξεων) σε ένα έγγραφο που περιλαμβάνει όλους τους όρους
- Παράδειγμα:
  - Έγγραφο: ο δήμος προχώρησε σε παραχώρηση αδειών
  - Ερώτημα: δήμος παραχώρηση
  - Μικρότερο παράθυρο: 4
- Όσο μικρότερο είναι το ω τόσο καλύτερα ταιριάζει το έγγραφο στο ερώτημα
- Αν το έγγραφο δεν περιλαμβάνει όλους τους όρους τότε θέτουμε στο ω ένα πολύ μεγάλο αριθμό
- Η τεχνική υιοθετείται για να στηρίξει τις proximity-weighted scoring functions που χρησιμοποιούν οι σύγχρονες μηχανές αναζήτησης

Designing Scoring Functions

- Στόχος των διεπαφών στις μηχανές αναζήτησης είναι να αποκρύψουν την πολυπλοκότητα των υπολογισμών από τους χρήστες
- Με αυτό τον τρόπο στηρίζουν την εισαγωγή ερωτημάτων σε ελεύθερο κείμενο
- Όμως, πως χειριζόμαστε τα ερωτήματα;
- Εξαρτάται από:
  - Το πλήθος των ερωτημάτων / χρηστών
  - Την κατανομή των ερωτημάτων
  - Τη συλλογή των εγγράφων

Designing Scoring Functions

- Ένας query parser αναλαμβάνει να πάρει το ερώτημα των χρηστών και να το μετατρέψει σε ερώτημα με τελεστές που θα εκτελεστεί στα ευρετήρια που διατηρούνται
- Πολλές φορές εκτελείται ένα πλήθος ερωτημάτων
  - Εκτελούμε το ερώτημα ως ένα phrase query – υπολογίζουμε και ταξινομούμε ως προς το vector space scoring value
  - Αν τα αποτελέσματα περιλαμβάνουν λιγότερα από 10 έγγραφα, τότε εκτελούμε ερωτήματα με λιγότερους όρους – υπολογίζουμε και ταξινομούμε
  - Αν και πάλι πάρουμε λίγα αποτελέσματα, τότε εκτελούμε ερωτήματα με ένα όρο

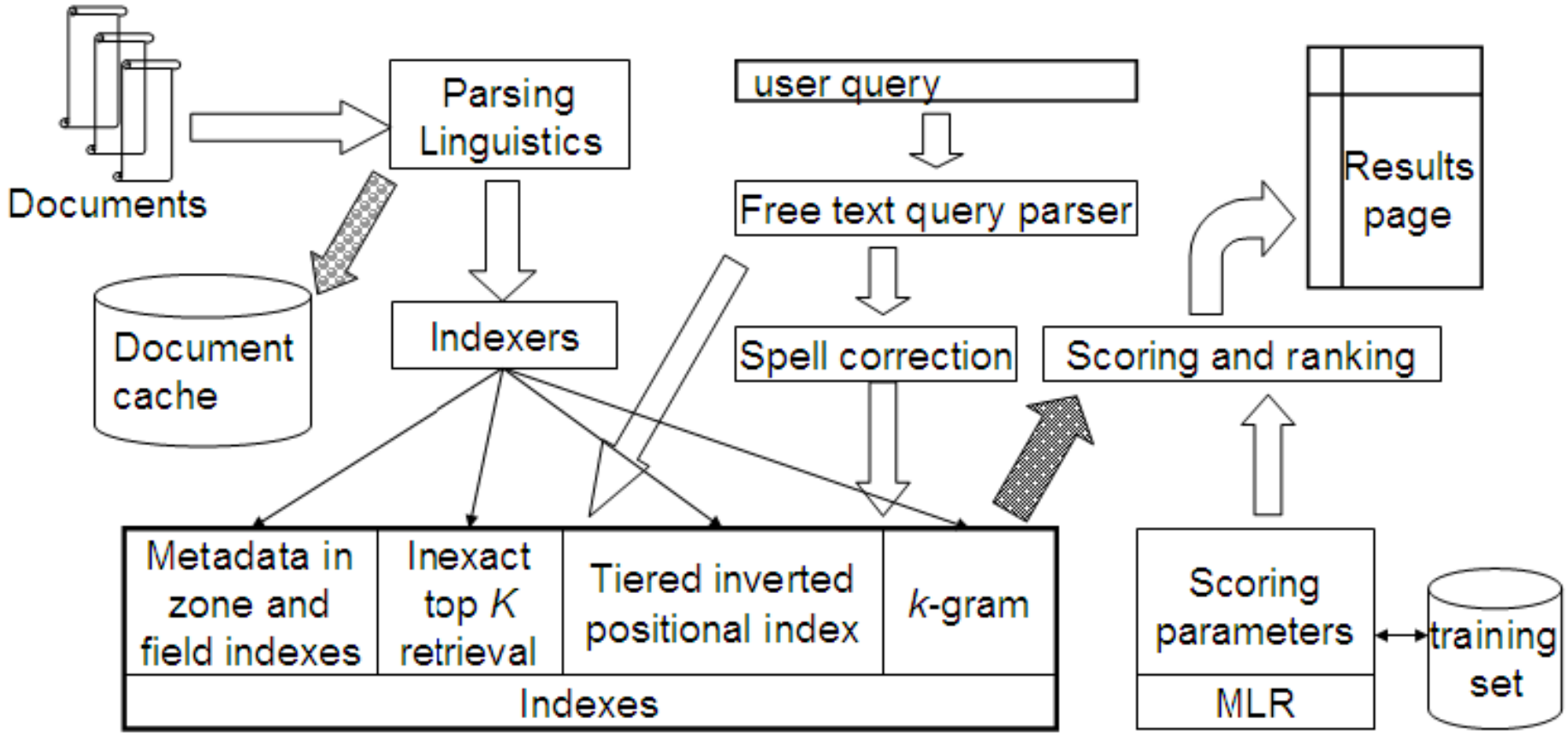
Designing Scoring Functions

- Γενικά, οι τελικές τιμές / σκορ υπολογίζονται σαν aggregated values
- Για παράδειγμα μπορούμε να υπολογίσουμε το cosine similarity, το static quality measure, το proximity weighting value, κ.λπ.
- Η ‘συγχωνευμένη’ τιμή ονομάζεται accumulated evidence για ένα έγγραφο

Designing Scoring Functions

- Η επιλογή των parsers, των μεθόδων για το aggregation, κ.λπ. εξαρτάται από τη μηχανή αναζήτησης
- Κάποιες προσφέρουν toolkits όπου μπορούμε να θέσουμε τιμές στις παραμέτρους και να ‘επηρεάσουμε’ τον υπολογισμό των τελικών τιμών

Ολοκληρωμένα Συστήματα



Ανάκτηση Πληροφορίας από XML έγγραφα

Εισαγωγή

- Ο κύριος αποθηκευτικός χώρος που υιοθετείται για IR είναι οι βάσεις δεδομένων
- Όμως υπάρχει η ανάγκη να ανακτήσουμε πληροφορίες από μη δομημένα έγγραφα (unstructured text)
- Οι βάσεις δεδομένων βασίζονται στο σχεσιακό μοντέλο
- Υπάρχουν επίσης πολλές άλλες πηγές δομημένων (structured) πληροφοριών που μπορούν να χρησιμοποιηθούν
- Τα ερωτήματα μπορεί και αυτά να είναι είτε δομημένα ή αδόμητα
- Παραδείγματα δομημένων πηγών:
  - ψηφιακές βιβλιοθήκες
  - Βάσεις δεδομένων
  - Blogs
  - Κείμενα με tags

Εισαγωγή

- Σε όλες αυτές τις εφαρμογές θέλουμε να ‘τρέξουμε’ ερωτήματα που να συνδυάζουν κείμενα με δομημένη πρόσβαση στην πληροφορία
- Παραδείγματα:
  - Give me a full-length article on fast fourier transforms – ψηφιακές βιβλιοθήκες
  - Δώσε μου άρθρα που σχετίζονται με τουριστικές πληροφορίες για την πόλη της Λαμίας – tagged κείμενα

Extensive Markup Language - XML

- Πρόκειται για ένα ‘πρότυπο’ για την ‘κωδικοποίηση’ δομημένων εγγράφων
- Μας ενδιαφέρει η περίπτωση όπου η XML κωδικοποιεί δεδομένα εκτός από κείμενα
- Παράδειγμα:
  - Μια επιχείρηση μπορεί να θέλει να καταγράψει τις πληροφορίες για το σύστημα διαχείρισης των πόρων της και να εξάγει στη συνέχεια γραφήματα
- Τέτοιου είδους εφαρμογές ονομάζονται προσανατολισμένες στα δεδομένα (data centric)
- Ο λόγος είναι ότι αριθμητικά και δεδομένα ιδιότητα-τιμή κυριαρχούν
- Σε αυτού του είδους τις εφαρμογές τα κείμενα αποτελούν ένα μικρό ποσοστό
- Ο κυρίαρχος όρος είναι δομημένη ανάκτηση ενώ μπορούμε συναντήσουμε και τον όρο ημι-δομημένη ανάκτηση

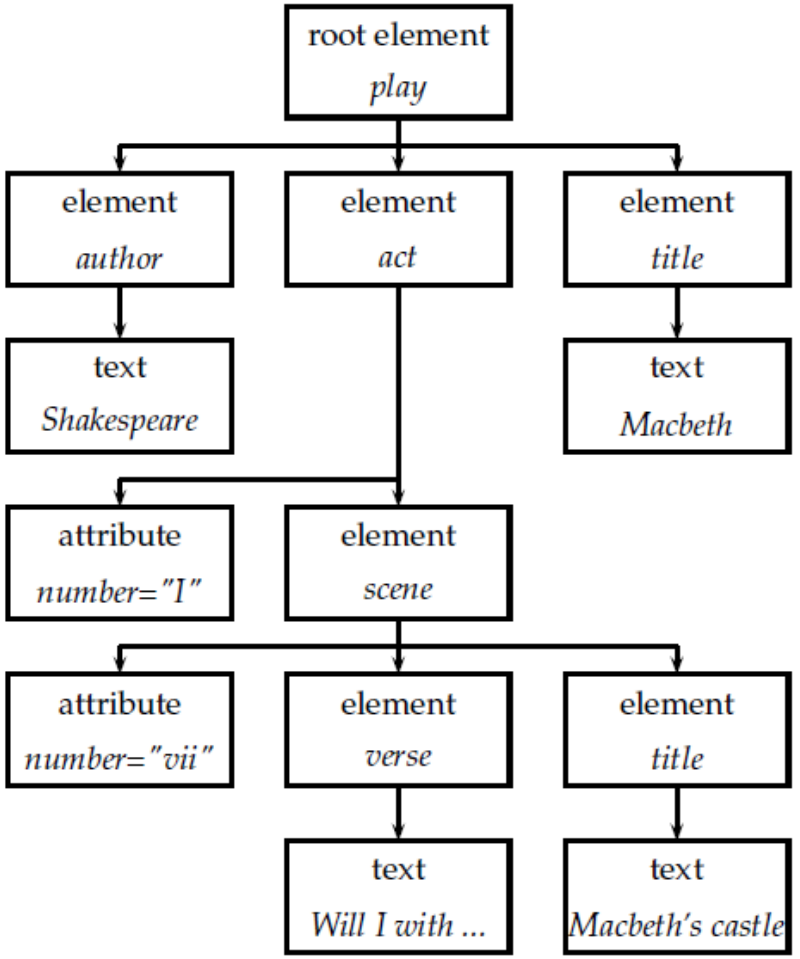
Extensive Markup Language - XML

- Ένα XML έγγραφο είναι ένα ‘ταξινομημένο’, ιεραρχικό δένδρο
- Κάθε κόμβος του δένδρου είναι ένα XML στοιχείο (XML element)
- Κάθε στοιχείο γράφεται ανοίγοντας και κλείνοντας ένα συγκεκριμένο tag
- Κάθε κόμβος / στοιχείο έχει μια τιμή και μπορεί να έχει παιδιά
- Οι εσωτερικοί κόμβοι κωδικοποιούν είτε τη δομή του εγγράφου ή τα μεταδεδομένα

Extensive Markup Language - XML

- Παράδειγμα:

```
<play>
<author>Shakespeare</author>
<title>Macbeth</title>
<act number="I">
<scene number="vii">
<title>Macbeth's castle</title>
<verse>Will I with wine and wassail ...</verse>
</scene>
</act>
</play>
```



Extensive Markup Language - XML

- Η προσπέλαση των XML εγγράφων γίνεται μέσω του XML Document Object Model (DOM)
- Αναπαριστά τα στοιχεία,τις ιδιότητες και το κείμενο στους κόμβους ενός δένδρου
- Με ένα API μπορούμε να προσπελάσουμε ένα έγγραφο ξεκινώντας από τη ρίζα και προχωρώντας προς τα φύλλα

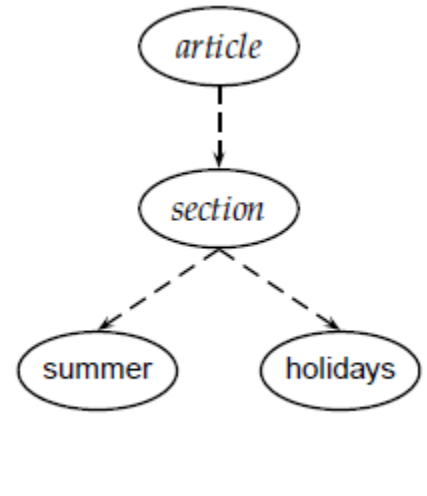
Extensive Markup Language - XML

- Το Xpath είναι το μέσο για να απαριθμήσουμε μονοπάτια σε μια συλλογή XML εγγράφων
- Το schema θέτει περιορισμούς στη δομή ενός XML εγγράφου
- Για το σχήμα των εγγράφων υπάρχουν δύο πρότυπα: το XML DTD (document type definition) και το XML Schema

Extensive Markup Language - XML

- Μια κοινή μορφοποίηση για τα XML ερωτήματα ειναι το NEXI (Narrowed Extended Xpath I)
- Παράδειγμα:
  - Αναπαριστούμε το ερώτημα με 4 γραμμές αλλά το διαβάζουμε σαν μια ενότητα
  - Το //section είναι ενσωματωμένο στο //article
  - Το συγκεκριμένο ερώτημα αναζητά τμήματα που σχετίζονται με summer holidays που είναι τμήματα άρθρων από το 2001 μέχρι το 2002

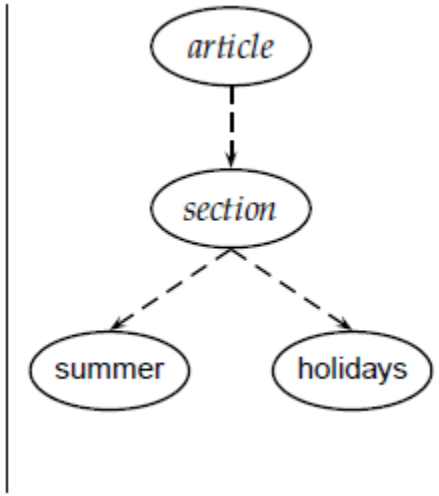
```
//article
[.//yr = 2001 or .//yr = 2002]
//section
[about(.,summer holidays)]
```



Extensive Markup Language - XML

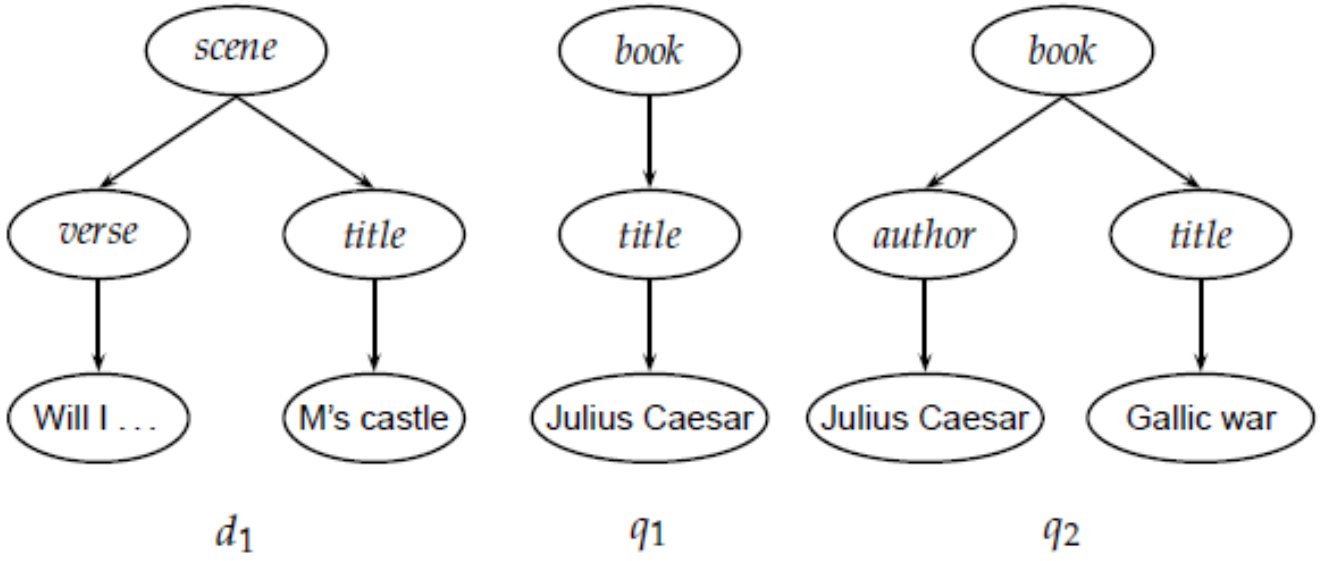
- Παράδειγμα (συνέχεια):
  - Η τελεία αναφέρεται στο στοιχείο που η κάθε αναφορά τροποποιεί
  - Η αναφορά [.//yr=2001 or .//yr=2002] τροποποιεί το //article
  - Το about είναι ένας περιορισμός που χρησιμοποιείται για το ranking

```
//article
[.//yr = 2001 or .//yr = 2002]
//section
[about(.,summer holidays)]
```



Extensive Markup Language - XML

- Αν απορρίψουμε τις ιδιότητες συσχέτισης των στοιχείων τότε μπορούμε να αναπαραστήσουμε τα έγγραφα ως δένδρα που περιέχουν μόνο κόμβος που απεικονίζουν τα στοιχεία
- Παράδειγμα:

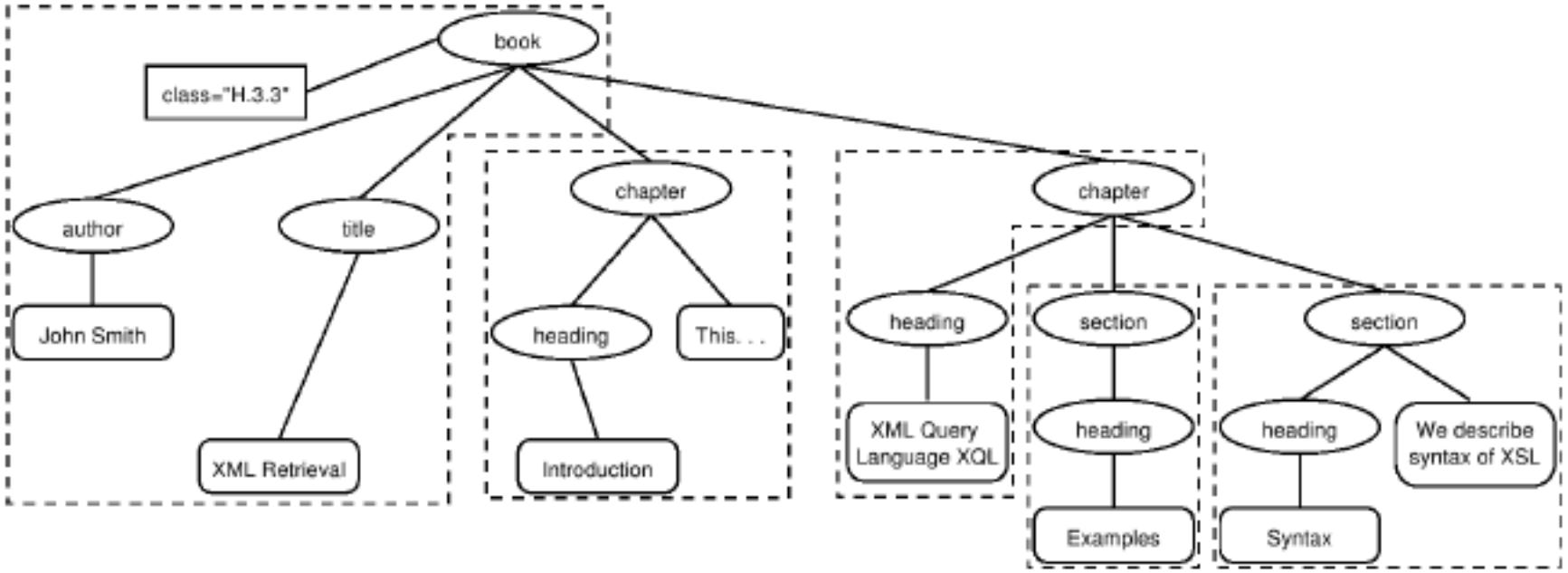


Προκλήσεις

- Προκλήσεις:
  - Οι χρήστες θέλουν να πάρουν τμήματα των εγγράφων και όχι ολόκληρα τα έγγραφα
    - Το κριτήριο για την επιλογή του πιο κατάλληλου τμήματος είναι:
      - Ένα σύστημα πρέπει πάντα να ανακτά το πιο κατάλληλο και συγκεκριμένο τμήμα ενός εγγράφου ως απάντηση σε ένα ερώτημα (αρχή της ανάκτησης σε δομημένα έγγραφα – structured document retrieval principle)
    - Η συγκεκριμένη αρχή στοχεύει στην ανάκτηση του μικρότερου δυνατού τμήματος των εγγράφων
  - Άλλη πρόκληση είναι το ποια τμήματα θα υιοθετηθούν στο ευρετήριο
    - Για μη δομημένα έγγραφα κατάλληλα τμήματα είναι: αρχεία, μηνύματα ηλεκτρονικού ταχυδρομείου, ιστοσελίδες, κ.λπ.
    - Στα δομημένα έγγραφα έχουν προταθεί διάφορες λύσεις

Προκλήσεις

- Μια λύση είναι να ομαδοποιήσουμε τους κόμβους ενός XML εγγράφου σε μη επικαλυπτόμενα ψευδο-έγγραφα
- Παράδειγμα: βιβλία, κεφάλαια και τμήματα θεωρούνται σαν μονάδες του ευρετηρίου αλλά είναι μη επικαλυπτόμενα



Προκλήσεις

- Άλλη λύση είναι να υιοθετήσουμε τα μεγαλύτερα στοιχεία, π.χ. Το βιβλίο
- Στη συνέχεια προχωρούμε σε επεξεργασία για να βρούμε ποιο είναι το στοιχείο που αποτελεί την καλύτερη ενότητα

Προκλήσεις

- Άλλη λύση είναι να ψάξουμε όλα τα φύλλα του δένδρου και να βρούμε το πιο σχετικό
- Στη συνέχεια ‘ανεβαίνουμε’ προς τα πάνω και επεκτείνουμε τα τμήματα που απεικονίζονται από τα φύλλα προς πιο μεγάλες ενότητες

Προκλήσεις

- Τελευταία λύση είναι να βάλουμε στο ευρετήριο όλα τα στοιχεία
- Προφανώς υπάρχει πρόβλημα με τον όγκο των στοιχείων
- Επίσης, στα XML έγγραφα, αρκετά στοιχεία δεν έχουν κάποιο ιδιαίτερο νόημα όσον αφορά σε απαντήσεις ερωτημάτων
- Αν βάλουμε όλα τα στοιχεία στο ευρετήριο αυτό σημαίνει ότι τα αποτελέσματα των αναζητήσεων θα έχουν πλεονάζουσα πληροφορία
- Παράδειγμα:
  - Τα φύλλα θα εμφανίζονται αρκετές φορές στα αποτελέσματα
- Η επιστροφή πλεονάζουσας πληροφορίας δεν είναι και τόσο αποδοτική για τους χρήστες

Προκλήσεις

- Η πλεονάζουσα πληροφορία προκαλείται από τα στοιχεία που είναι nested στη δομή του εγγράφου
- Συνεπώς, μπορούμε να εφαρμόσουμε κάποιους περιορισμούς:
  - Αποκλείουμε όλα τα ‘μικρά’ στοιχεία
  - Αποκλείουμε όλα τα στοιχεία που δεν ενδιαφέρουν τους χρήστες (πρέπει να βασιστούμε σε log files)
  - Αποκλείουμε όλα τα στοιχεία των οποίων οι απόγονοι κρίνονται άσχετοι (εφόσον έχουμε δεδομένα συσχέτισης)
  - Κρατάμε μόνο τα στοιχεία που κρίνουν οι experts (π.χ. Σχεδιαστές ή βιβλιοθηκονόμοι)

Προκλήσεις

- Αν οι λίστες των αποτελεσμάτων περιέχουν ακόμα nested στοιχεία, τότε μπορούμε σε επόμενη φάση επεξεργασίας να απορρίψουμε κάποια από αυτά
- Εναλλακτικά, μπορούμε να ‘σημαδέψουμε’ τους όρους των ερωτημάτων ώστε να αποκτήσουμε την προσοχή των χρηστών
- Όταν οι χρήστες είναι γνώστες των σχημάτων των εγγράφων μπορούν να καθορίσουν τον τύπο των στοιχείων που επιθυμούν

Προκλήσεις

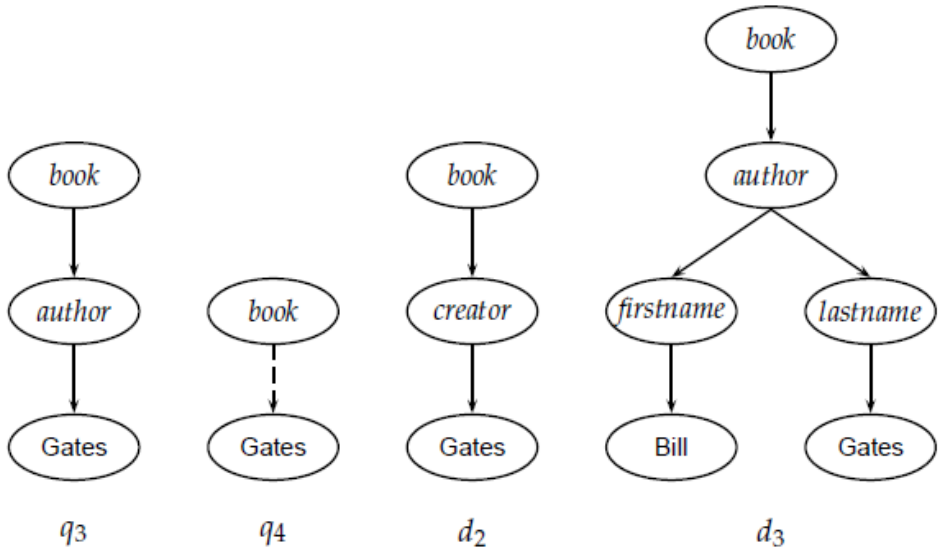
- Μια άλλη πρόκληση είναι να αναγνωρίσουμε το διαφορετικό πλαίσιο (context) όπου παρουσιάζεται ένας όρος όταν υπολογίζουμε διάφορες στατιστικές για να βγάλουμε ένα τελικό ranking
- Παράδειγμα:
  - Όρος: Gates
  - Κόμβος: author
  - Η αναφορά είναι άσχετη με ένα κόμβο section όταν χρησιμοποιείται για να αναφερθούμε στον πληθυντικό της λέξης gate

Προκλήσεις

- Μια λύση είναι να υπολογίσουμε το idf για τα ζεύγη XML-context/term
- Παράδειγμα:
  - Υπολογίζουμε διαφορετικό αποτέλεσμα για τα author#"Gates" & section#"Gates"
- Υπάρχουν και επιπρόσθετα προβλήματα που έχουν να κάνουν με τη διαφοροποίηση του context
- Παράδειγμα:
  - Δεν διαχωρίζουμε τα ονόματα των authors από τα ονόματα των επιχειρήσεων όταν και τα δύο έχουν κοινό πατέρα το name

Προκλήσεις

- Άλλο πρόβλημα σχετίζεται με την ετερογένεια των σχημάτων
- Διαφορετικά σχήματα συναντώνται σε συλλογές XML εγγράφων
- Κάθε έγγραφο μπορεί να προέρχεται από διαφορετικές πηγές
- Το φαινόμενο αυτό ονομάζεται schema heterogeneity ή schema diversity
- Στο παράδειγμα το q_3 δεν θα συνδεθεί με τα έγγραφα παρά το ότι είναι σχετικά
- Κάποιες λύσεις είναι να έχουμε ημι-αυτόματο matching ή approximate matching

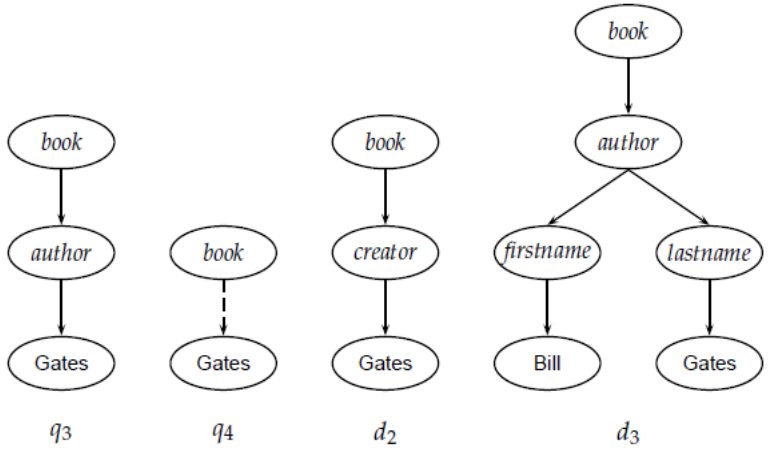


Προκλήσεις

- Οι χρήστες δεν είναι γνώστες των ονομάτων και της δομής των εγγράφων
- Ιδεατά, στη διεπαφή μπορούμε να παρουσιάσουμε τη δομή του δένδρου και να τους επιτρέψουμε να καθορίσουν τα στοιχεία για τα οποία θέτουν το ερώτημα
- Η προσέγγιση επιβάλλει περίπλοκο σχεδιασμό των διεπαφών όπου τίθενται τα ερωτήματα

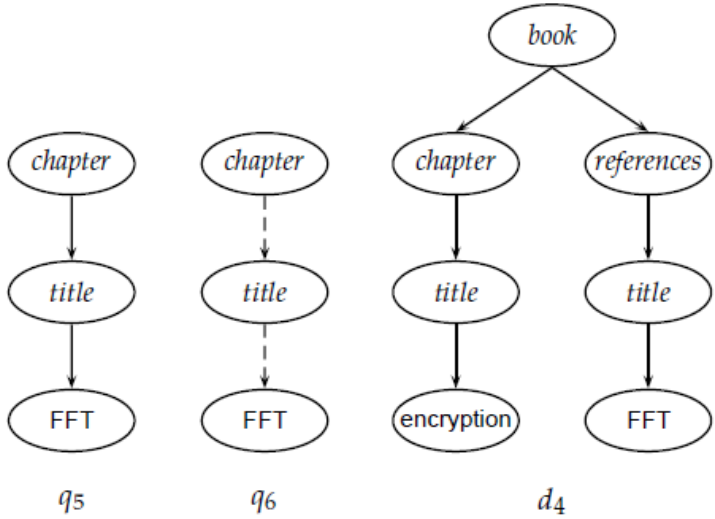
Προκλήσεις

- Μπορούμε επίσης να μεταφράσουμε όλες τις σχέσεις πατέρα-παιδιού σαν συσχετίσεις όπου επιτρέπουμε οποιοδήποτε αριθμό κόμβων
- Αυτά τα ερωτήματα τα καλούμε εκτεταμένα ερωτήματα (extended queries)
- Στο q_4 βλέπουμε τις καθοδικές συσχετίσεις με διακεκομμένες γραμμές
- Η συσχέτιση book//#"Gates" μας δείχνει ένα βιβλίο το οποίο αναφέρει κάπου το Gates ενώ το μονοπάτι μπορεί να είναι αρκετά μεγάλο



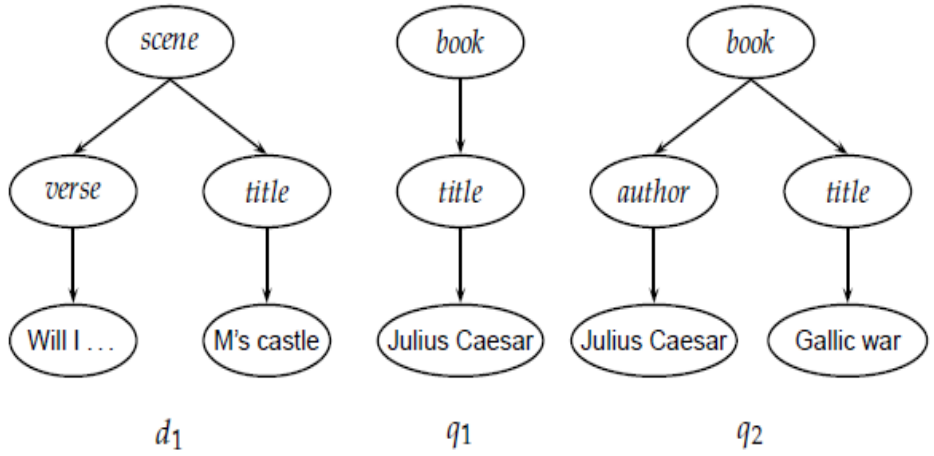
Προκλήσεις

- Στο ερώτημα q_5 αναζητούμε το FFT
- Ας υποθέσουμε πως δεν υπάρχει τέτοιο κεφάλαιο αλλά υπάρχει μια αναφορά σε βιβλία στο d_4
- Η αναφορά σε βιβλίο δεν είναι ακριβώς αυτό που ψάχνει ο χρήστης αλλά είναι καλύτερο από το να μην επιστρέψουμε κάτι
- Σε αυτό το παράδειγμα τα εκτεταμένα ερωτήματα δεν προσφέρουν κάτι



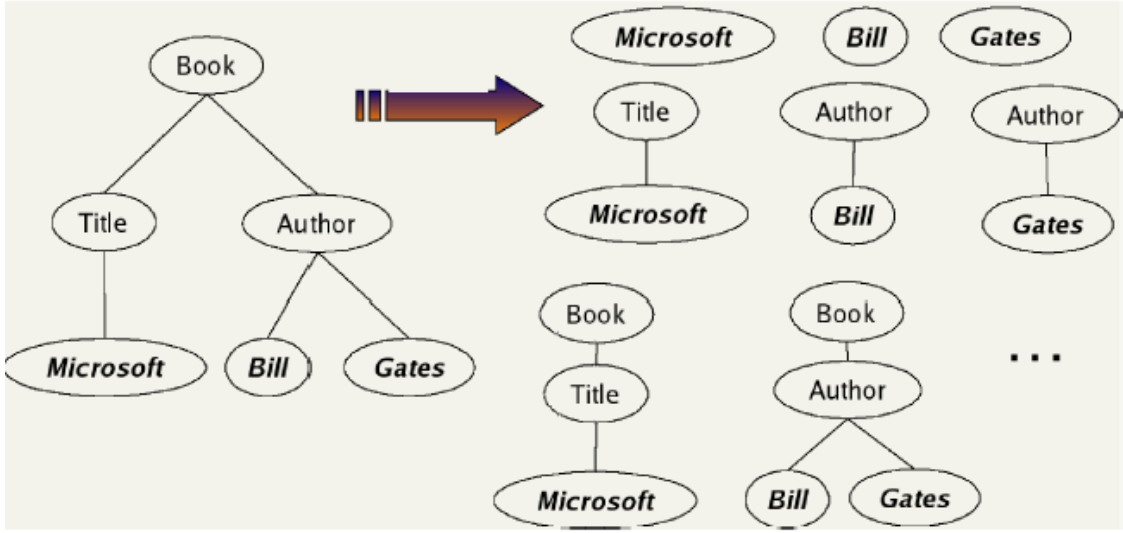
Vector Space Model for XML Documents

- Μπορούμε να εφαρμόσουμε το vector space μοντέλο και στα XML έγγραφα
- Έστω ότι με βάση το διπλανό σχήμα ψάχνουμε ένα βιβλίο με τίτλο Julius Caesar
- Θέλουμε να έχουμε ένα ταίριασμα με το q_1 αλλά όχι με το q_2
- Για την XML ανάκτηση πρέπει να διαχωρίσουμε τη λέξη Caesar στον τίτλο από τη λέξη Caesar στο συγγραφέα
- Ένας τρόπος για να το κάνουμε αυτό είναι να έχουμε κάθε διάσταση στο vector space να κωδικοποιεί κάθε λέξη μαζί με τη θέση της μέσα στο δένδρο



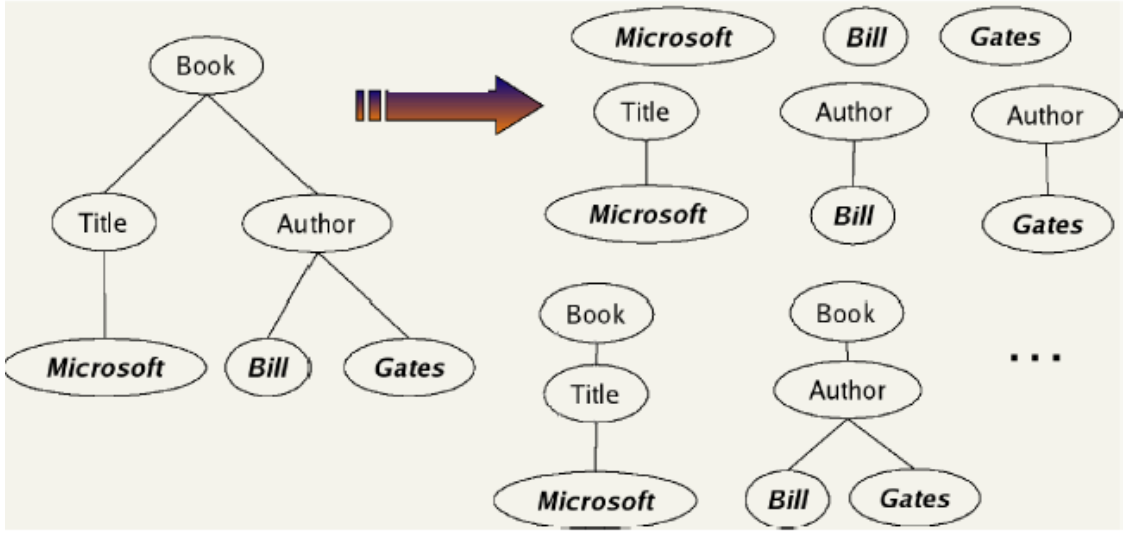
Vector Space Model for XML Documents

- Αρχικά σπάμε τον κάθε κόμβο σε πολλαπλούς, ένα για κάθε λέξη
- Το φύλλο Bill Gates θα σπάσει σε δύο φύλλα Bill & Gates
- Ορίζουμε τις διαστάσεις του vector space να αντιστοιχούν σε υπο-δένδρα (lexicalized subtrees) των εγγράφων
- Κάθε υπο-δένδρο περιλαμβάνει τουλάχιστον ένα όρο του λεξικού



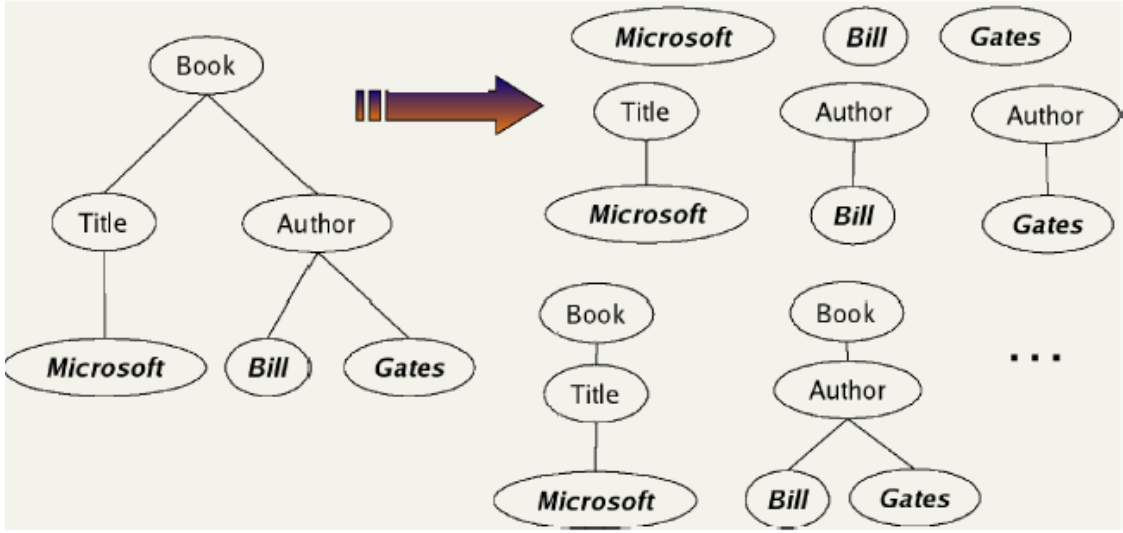
Vector Space Model for XML Documents

- Μπορούμε τώρα να αναπαραστήσουμε τα έγγραφα και τα ερωτήματα σαν διανύσματα και να υπολογίσουμε τα ταιριάσματα
- Υπάρχει ένα trade off ανάμεσα στις διαστάσεις των διανυσμάτων και την ακρίβεια των αποτελεσμάτων
- Αν περιορίσουμε τις διαστάσεις στους όρους του λεξικού, το μοντέλο θα ανακτήσει πολλά έγγραφα που δεν ταιριάζουν με τη δομή του ερωτήματος



Vector Space Model for XML Documents

- Αν δημιουργήσουμε μια ξεχωριστή διάσταση για κάθε υπο-δένδρο, τότε οι διαστάσεις που θα προκύψουν θα είναι πολλές
- Ένας συμβιβασμός είναι να υιοθετήσουμε όλα τα μονοπάτια που τερματίζουν με ένα συγκεκριμένο όρο του λεξικού
- Καλούμε κάθε ζεύγος XML-context/term ως structural term και το συμβολίζουμε με <c,t>



Vector Space Model for XML Documents

- Μια απλή μετρική για την ομοιότητα ενός μονοπατιού σε ένα ερώτημα c_q με ένα μονοπάτι σε ένα έγγραφο c_d είναι η ακόλουθη ομοιότητα πλαισίου (context resemblance) C_R:

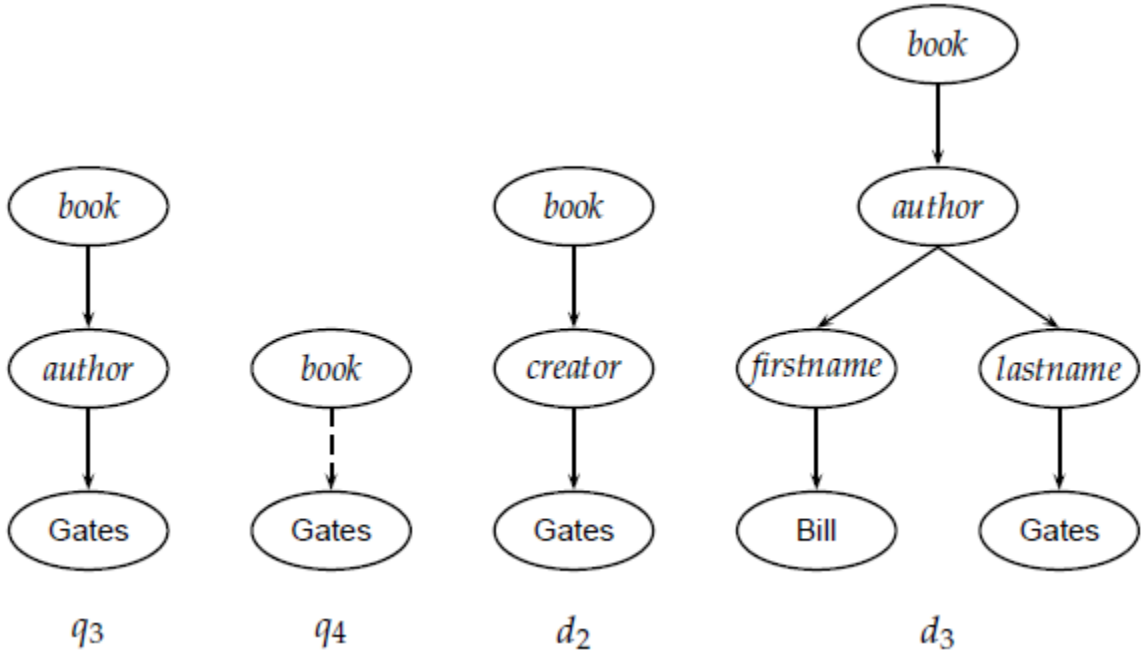
$$C_R(c_q, c_d) = \begin{cases} \frac{1+|c_q|}{1+|c_d|} & \text{if } c_q \text{ matches } c_d \\ 0 & \text{if } c_q \text{ does not match } c_d \end{cases}$$

- $|c_q|$ και $|c_d|$ είναι το πλήθος των κόμβων στα μονοπάτια του ερωτήματος και του εγγράφου
- Έχουμε ταίριασμα αν μπορέσουμε να μετατρέψουμε το c_q στο c_d εισάγοντας επιπλέον κόμβους

Vector Space Model for XML Documents

- Παράδειγμα:
  - $C_R(c_{q4}, c_{d2})=3/4=0.75$
  - $C_R(c_{q4}, c_{d3})=3/5=0.6$
- Βασιζόμαστε στα σχετικά μονοπάτια από την κορυφή μέχρι τα φύλλα

$$C_R(c_q, c_d) = \begin{cases} \frac{1+|c_q|}{1+|c_d|} & \text{if } c_q \text{ matches } c_d \\ 0 & \text{if } c_q \text{ does not match } c_d \end{cases}$$



Vector Space Model for XML Documents

- Το τελικό αποτέλεσμα θα υπολογιστεί σαν μια παραλλαγή του αρχικού vector space μοντέλου
- Ισχύει:

$$\text{SIMNOMERGE}(q,d) = \sum_{c_k \in B} \sum_{c_l \in B} \text{CR}(c_k, c_l) \sum_{t \in V} \text{weight}(q,t,c_k) \frac{\text{weight}(d,t,c_l)}{\sqrt{\sum_{c \in B, t \in V} \text{weight}^2(d,t,c)}}$$

- V είναι το λεξικό των μη δομημένων όρων
- B είναι το σύνολο όλων των XML εγγράφων
- weight(q,t,c), weight(d,t,c) είναι τα βάρη για τον όρο t στο XML context c στο ερώτημα q και στο έγγραφο d
- Τα βάρη μπορεί να υπολογιστούν με την τεχνική idf-wf

Vector Space Model for XML Documents

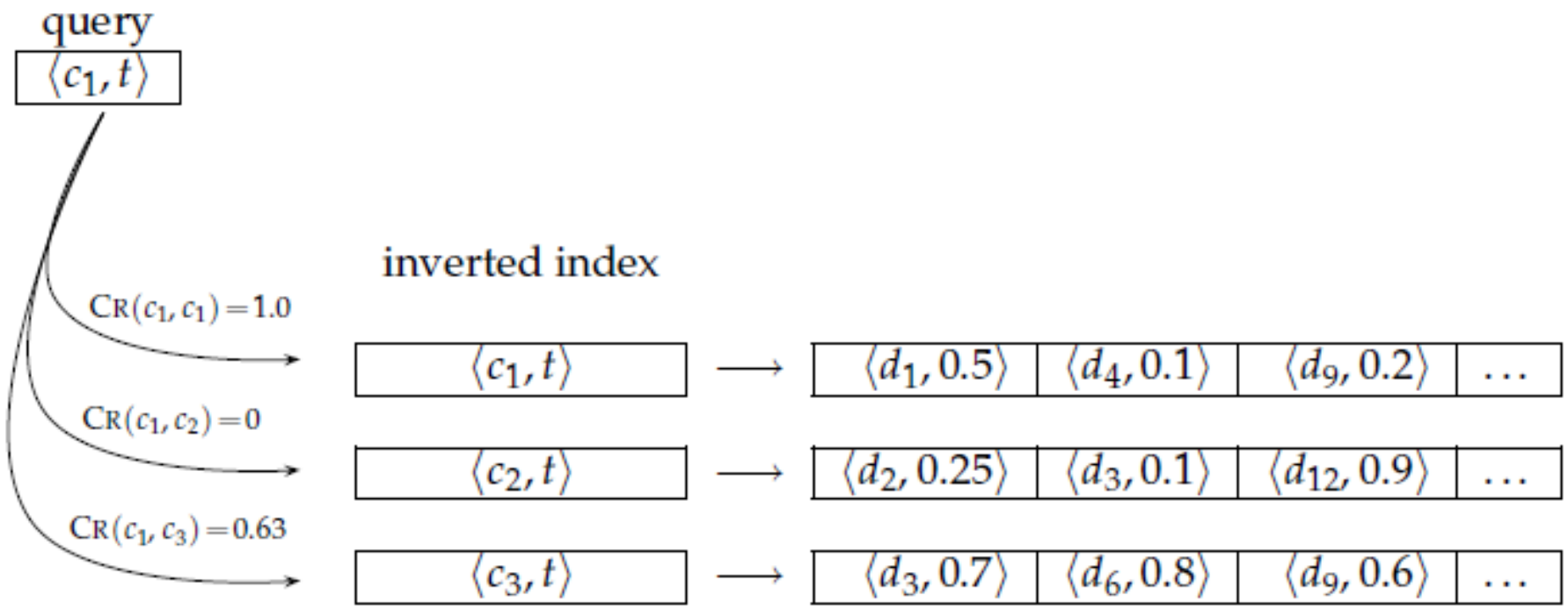
$$\text{SIMNOMERGE}(q,d) = \sum_{c_k \in B} \sum_{c_l \in B} \text{CR}(c_k, c_l) \sum_{t \in V} \text{weight}(q,t,c_k) \frac{\text{weight}(d,t,c_l)}{\sqrt{\sum_{c \in B, t \in V} \text{weight}^2(d,t,c)}}$$

- Η τιμή της μετρικής μπορεί να είναι πάνω από 1.0
- Η διαίρεση με τη ρίζα του αθροίσματος των τετραγώνων υιοθετείται για να κανονικοποιήσουμε ως προς το μήκος του εγγράφου
- Δεν κανονικοποιούμε ως προς το μήκος του ερωτήματος αφού η ποσότητα (ρίζα του αθροίσματος των τετραγώνων) είναι ίδια για όλα τα έγγραφα

Vector Space Model for XML Documents

```
SCOREDOCUMENTSWITHSIMNOMERGE(q, B, V, N, normalizer)
 1  for n ← 1 to N
 2  do score[n] ← 0
 3  for each ⟨c_q, t⟩ ∈ q
 4  do w_q ← WEIGHT(q, t, c_q)
 5      for each c ∈ B
 6      do if CR(c_q, c) > 0
 7            then postings ← GETPOSTINGS(⟨c, t⟩)
 8                 for each posting ∈ postings
 9                 do x ← CR(c_q, c) * w_q * weight(posting)
10                    score[docID(posting)] += x
11  for n ← 1 to N
12  do score[n] ← score[n]/normalizer[n]
13  return score
```

Vector Space Model for XML Documents



Text Centric and Data Centric Retrieval

- Στα text-centric XML περιγραφές δίνουμε μεγαλύτερη βαρύτητα στο κείμενο
- Αυτό επιτυγχάνεται αν συμπεριλάβουμε κάποιους δομικούς περιορισμούς στην επεξεργασία μη δομημένης πληροφορίας
- Η ανάκτηση XML εγγράφων σε αυτές τις περιπτώσεις χαρακτηρίζεται από:
  - Μεγάλα πεδία κειμένου (τμήματα εγγράφων)
  - Μη ακριβές ταίριασμα
  - Ταξινόμηση των εγγράφων ως προς τη συσχέτισή τους

Text Centric and Data Centric Retrieval

- Στα data-centric XML έγγραφα κωδικοποιούμε κυρίως αριθμητικά δεδομένα
- Όταν εφαρμόζουμε ερωτήματα σε data-centric έγγραφα στοχεύουμε στον ακριβή προσδιορισμό των συνθηκών και την πλήρη ικανοποίησή τους
- Αυτό από μόνο του αποτυπώνει έμφαση στη δομή των XML εγγράφων

Πιθανοτικοθεωρητική Ανάκτηση Πληροφοριών

Εισαγωγή

- Έχουμε ήδη δει πως μπορούμε να κατατάξουμε τα έγγραφα σε σχετικά και μη σχετικά
- Με βάση αυτή την κατηγοριοποίηση, μπορούμε να υπολογίσουμε την πιθανότητα ένας όρος να συναντηθεί σε ένα σχετικό έγγραφο
- Η πιθανοτικοθεωρητική προσέγγιση προσφέρει μια διαφορετική βάση σε σχέση με τα προηγούμενα μοντέλα

Θεωρία Πιθανοτήτων

- Μια μεταβλητή Α αναπαριστά ένα συμβάν (event) που είναι ένα υπο-σύνολο του χώρου των πιθανών τιμών
- Μπορούμε να αναπαραστήσουμε το υποσύνολο με χρήση μια τυχαίας μεταβλητής και χρήση μια συνάρτησης που απεικονίζει τα αποτελέσματα σε πραγματικούς αριθμούς
- Το υποσύνολο είναι το domain πάνω στο οποίο η μεταβλητή Α παίρνει την τιμή της
- Συχνά, δεν μπορούμε να γνωρίζουμε με βεβαιότητα αν το συμβάν αληθεύει στον πραγματικό κόσμο
- Μπορούμε επίσης να ζητήσουμε την πιθανότητα του συμβάντος $0<=P(A)<=1$

Θεωρία Πιθανοτήτων

- Για δύο συμβάντα το κοινό συμβάν και των δύο περιγράφεται από τη συνδυασμένη πιθανότητα (joint probability) P(A,B)
- Η υπο συνθήκη πιθανότητα (conditional probability) P(A|B) εκφράζει την πιθανότητα ένα συμβάν A να απαντηθεί δεδομένου ότι έχει συμβεί το B
- Η σχέση ανάμεσα στις συνδυασμένες και τις υπο συνθήκη πιθανότητες δίνεται από τον αλυσιδωτό κανόνα (chain rule):

$$P(A, B) = P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

- Η συνδυαστική πιθανότητα δύο συμβάντων ισούται με το γινόμενο της πιθανότητας του ενός επί την πιθανότητα του άλλου δεδομένου ότι έχει συμβεί το πρώτο

Θεωρία Πιθανοτήτων

- Όταν γράφουμε $P(\overline{A})$ συμβολίζουμε την πιθανότητα του συμπληρώματος ενός συμβάντος
- Οπότε έχουμε:

$$P(\overline{A}, B) = P(B|\overline{A})P(\overline{A})$$

- Η θεωρία πιθανοτήτων συμπεριλαμβάνει και ένα κανόνα τμηματοποίησης (partition rule)
- Ο κανόνας μας λέει ότι αν ένα συμβάν B μπορεί να χωριστεί σε ένα εξαντλητικό σύνολο από μη σχετιζόμενες περιπτώσεις, τότε η πιθανότητα του B είναι το άθροισμα των πιθανοτήτων των περιπτώσεων αυτών
- Μια ειδική περίπτωση του κανόνα αυτού μας δίνει ότι:

$$P(B) = P(A, B) + P(\overline{A}, B)$$

Θεωρία Πιθανοτήτων

- Από όλα τα προηγούμενα μπορούμε να εξάγουμε τον κανόνα του Bayes ο οποίος αντιστρέφει τις υπο συνθήκη πιθανότητες

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \left[\frac{P(B|A)}{\sum_{X\in\{A,\overline{A}\}} P(B|X)P(X)}\right] P(A)$$

- Ξεκινούμε από την εκτίμηση του πόσο πιθανό είναι ένα συμβάν A – αυτή ονομάζεται prior probability P(A)
- Ο κανόνας του Bayes μας βοηθά να εξάγουμε την posterior probability P(A|B)
- Προφανώς έχουμε δει το B βασιζόμενοι στην πιθανότητα να συμβαίνει το B δεδομένου ότι συμβαίνει ή δεν συμβαίνει το A

Probability Ranking Principle

- Έστω ένα ερώτημα q και ένα έγγραφο d
- Έστω επίσης πως η τυχαία μεταβλητή $R_{q,d}$ αποτελεί μια ένδειξη ας λέει αν το d είναι σχετικό με το q
- Παίρνει την τιμή 1 όταν είναι σχετικό και την τιμή 0 σε διαφορετική περίπτωση
- Στη συνέχεια θα χρησιμοποιούμε μόνο το R για να αποτυπώσουμε τη μεταβλητή

Probability Ranking Principle

- Χρησιμοποιώντας το πιθανοτικοθεωρητικό μοντέλο, τα αποτελέσματα θα παορυσιαστούν στους χρήστες σε φθίνουσα σειρά πιθανότητας που απεικονίζει τη συσχέτιση ανάμεσα στο q και στο d
- Η πιθανότητα που προσπαθούμε να υπολογίσουμε είναι η P(R=1|d,q)
- Αυτή είναι η βάση για την τεχνική που ονομάζεται Probability Ranking Principle (PRP)

Probability Ranking Principle

- Στην πιο απλή μορφή της αρχής, δεν υπάρχει κόστος το οποίο αν θα έπρεπε να το λάβουμε υπόψιν μας για να διαφοροποιήσουμε τα βάρη
- Χάνουμε 1 ‘πόντο’ αν επιστρέψουμε ένα άσχετο έγγραφο ή αποτυγχάνοντας να επιστρέψουμε ένα σχετικό έγγραφο
- Πρόκειται για μια δυική κατάσταση
- Ονομάζεται 1/0 loss
- Ο στόχος είναι να επιστρέψουμε το καλύτερο δυνατό αποτέλεσμα σαν τα top-k έγγραφα για οποιαδήποτε τιμή του k
- Το μοντέλο PRP απλά ταξινομεί όλα τα έγγραφα σε φθίνουσα σειρά της πιθανότητας P(R=1|d,q)

Probability Ranking Principle

- Επίσης, έχουμε τη δυνατότητα αντί για μια ταξινομημένη λίστα να υπολογίσουμε το ελάχιστο κόστος και με τον κανόνα του Bayes (Bayes Optimal Decision Rule) να επιστρέψουμε τα έγγραφα τα οποία είναι πιο πιθανά να είναι σχετικά από το να είναι άσχετα

$$d \text{ is relevant iff } P(R = 1|d, q) > P(R = 0|d, q)$$

Probability Ranking Principle

- Ας υποθέσουμε τώρα πως έχουμε κάποιο κόστος για την ανάκτηση των εγγράφων
- Έστω C_1 είναι το κόστος της μη ανάκτησης ενός σχετικού εγγράφου και C_0 το κόστος ανάκτησης ενός άσχετου εγγράφου
- τότε η PRP μας λέει ότι αν για ένα έγγραφο d και ένα σύνολο d' εγγράφων που δεν έχουν ανακτηθεί

$$C_0 \cdot P(R = 0|d) - C_1 \cdot P(R = 1|d) \leq C_0 \cdot P(R = 0|d') - C_1 \cdot P(R = 1|d')$$

- το d είναι το επόμενο που πρέπει να ανακτηθεί

The Binary Independence Model

- Το μοντέλο της δυικής ανεξαρτησίας (Binary Independence Model - BIM) υιοθετείται σε σύνδυασμό με την PRP
- Εισάγει κάποιες υποθέσεις που βοηθούν στην εκτίμηση της πιθανότητας P(R|d,q)
- Τα ερωτήματα και τα έγγραφα αναπαριστώνται ως δυαδικά διανύσματα (binary term incidence vectors)
- Ένα έγγραφο αναπαριστάται από το διάνυσμα $\vec{x} = (x_1, x_2, \ldots, x_M)$ όπου x_t=1 σημαίνει πως ο όρος t υπάρχει στο έγγραφο d και x_t=0 διαφορετικά
- Προφανώς, αρκετά έγγραφα θα έχουν την ίδια αναπαράσταση
- Με τον ίδιο τρόπο αναπαριστούμε και τα ερωτήματα με τα διανύσματα $\vec{q}$

The Binary Independence Model

- Ο όρος ‘ανεξαρτησία’ στο μοντέλο σημαίνει πως οι όροι μοντελοποιούνται ανεξάρτητα
- Το μοντέλο δεν αναγνωρίζει συσχέτιση μεταξύ των όρων
- Εδώ υπάρχει ένα μειονέκτημα

The Binary Independence Model

- Ο στόχος μας είναι να εκτιμήσουμε πως η συχνότητα των όρων, η συχνότητα των εγγράφων, το μήκος των εγγράφων και άλλες στατιστικέςηρεάζουν την κρίση για το αν ένα έγγραφο είναι σχετικό
- Επίσης, θέλουμε να συνδυάσουμε αποδοτικά τις στατιστικές για να υπολογίσουμε την προαναφερόμενη πιθανότητα
- Στη συνέχεια μπορούμε να κατατάξουμε σε φθίνουσα σειρά τα έγγραφα

The Binary Independence Model

- Στο BIM μοντελοποιούμε την πιθανότητα P(R|d,q) σε όρους των δυαδικών διανυσμάτων $\mathrm{P}(\mathrm{R}|\vec{x},\vec{q})$
- Στη συνέχεια μπορούμε να χρησιμοποιήσουμε τον κανόνα του Bayes

$$P(R=1|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})}$$

$$P(R=0|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})}$$

- Στον κανόνα, οι $\mathrm{P}(\vec{x}|R=1,\vec{q})$ και $\mathrm{P}(\vec{x}|R=0,\vec{q})$ αναπαριστούν τις πιθανότητες του αν ένα σχετικό ή μη σχετικό (αντίστοιχα) έγγραφο έχει ανακτηθεί τότε η δυαδική αναπράστασή του είναι ίση με $\vec{x}$

The Binary Independence Model

- Υιοθετούμε κάποιες στατιστικές για τη συλλογή των εγγράφων ώστε να εκτιμήσουμε τις πιθανότητες
- Οι πιθανότητες $\mathrm{P}(R = 1|\vec{q})$ και $\mathrm{P}(\mathrm{R} = 0|\vec{q})$ απεικονίζουν τις prior πιθανότητες της ανάκτησης ενός σχετικού ή μη σχετικού εγγράφου δεδομένου ενός ερωτήματος $\vec{q}$
- Οι πιθανότητες αυτές μπορούν να υπολογιστούν αν γνωρίζουμε το ποσοστό των σχετικών εγγράφων στη συλλογή μας
- Επίσης, πρέπει να έχουμε:

$$P(R = 1|\vec{x}, \vec{q}) + P(R = 0|\vec{x}, \vec{q}) = 1$$

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

- Μπορούμε να υπολογίσουμε απ' ευθείας την πιθανότητα $\mathrm{P}(R = 1|\vec{x}, \vec{q})$
- Ο υπολογισμός γίνεται ως εξής:

$$O(R|\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(R=1|\vec{q})P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q})P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \cdot \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})}$$

- Το αριστερό κλάσμα στον τελευταίο όρο της συνάρτησης είναι σταθερός για ένα ερώτημα
- Αφού προσπαθούμε να κατατάξουμε έγγραφα μπορούμε να το παραλείψουμε

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

$$O(R|\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(R=1|\vec{q})P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q})P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \cdot \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})}$$

- Ο δεξιός όρος απαιτεί εκτίμηση
- Η εκτίμηση γίνεται ως εξής:

$$\frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})} = \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}$$

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}$$

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

$$O(R|\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(R=1|\vec{q})P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q})P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \cdot \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})}$$

$$\frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})} = \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}$$

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}$$

- Αφού $x_t=\{0,1\}$ μπορούμε να χωρίσουμε τους όρους για να πάρουμε:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=1} \frac{P(x_t=1|R=1,\vec{q})}{P(x_t=1|R=0,\vec{q})} \cdot \prod_{t:x_t=0} \frac{P(x_t=0|R=1,\vec{q})}{P(x_t=0|R=0,\vec{q})}$$

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

- Ορίζουμε τις ακόλουθες μεταβλητές:

	document	relevant ($R = 1$)	nonrelevant ($R = 0$)
Term present	$x_t = 1$	p_t	u_t
Term absent	$x_t = 0$	$1 - p_t$	$1 - u_t$

- Κάνουμε μια επιπρόσθετη υπόθεση:
  - Οι όροι που δεν συμβαίνουν σε ένα ερώτημα μπορεί ισοπίθανα να τους συναντήσουμε σε σχετικά ή άσχετα έγγραφα
- Αν $q_t=0$, $p_t=u_t$
- Συνεπώς, στους υπολογισμούς χρειάζεται να υπολογίσουμε τους όρους που συναντώνται σε ένα ερώτημα:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}$$

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}$$

- Το αριστερό γινόμενο εφαρμόζεται στους όρους του ερωτήματος που βρέθηκαν στο έγγραφο
- Το δεξιό γινόμενο εφαρμόζεται σε όρους του ερωτήματος που δεν βρέθηκαν μέσα στο έγγραφο

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

- Μπορούμε να κάνουμε την επόμενη αλλαγή έτσι ώστε το δεξιό γινόμενο να αφορά σε όλα τους όρους του ερωτήματος:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} \cdot \prod_{t:q_t=1} \frac{1-p_t}{1-u_t}$$

- Εφόσον το δεξιό γινόμενο αφορά σε όλους τους όρους, αυτό σημαίνει πως είναι σταθερό για όλους τους όρους
- Η ποσότητα $O(R|\vec{q})$ είναι επίσης σταθερή
- Άρα, η μόνη ποσότητα που χρήζει εκτίμησης είναι το αριστερό γινόμενο

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

- Για να αποτυπώσουμε μια κοινή προσέγγιση για όλα τα ερωτήματα μπορούμε να πάρουμε το λογάριθμο του γινομένου
- Συνεπώς παίρνουμε την ακόλουθη εξίσωση:

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

- Η ποσότητα αυτή καλείται Retrieval Status Value (RSV)
- Παίρνουμε:

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{(1-p_t)} + \log \frac{1-u_t}{u_t}$$

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{(1-p_t)} + \log \frac{1-u_t}{u_t}$$

- Το c_t απεικονίζει το λόγο των πιθανοτήτων όταν το έγγραφο είναι σχετικό ή όταν είναι άσχετο
- Το $\frac{p_t}{1-p_t}$ απεικονίζει τις πιθανότητες ο όρος να συναντάται σε ένα σχετικό έγγραφο
- Το $\frac{u_t}{1-u_t}$ απεικονίζει τις πιθανότητες ο όρος να συναντάται σε ένα άσχετο έγγραφο
- Το c_t καλείται odds ratio

Εξαγωγή Συνάρτησης για την Κατάταξη των Όρων

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{(1-p_t)} + \log \frac{1-u_t}{u_t}$$

- Επειδή χρησιμοποιούμε το λογάριθμο της ποσότητας, αυτός θα είναι 0 αν ο όρος έχει τις ίδιες πιθανότητες να εμφανιστεί σε σχετικά και άσχετα έγγραφα
- Η ποσότητα θα είναι θετική όταν ο όρος έχει περισσότερες πιθανότητες να εμφανιστεί σε σχετικά έγγραφα
- Το τελικό σκορ ενός εγγράφου θα έχει ως εξής:

  $RSV_d = \sum_{x_t=q_t=1} c_t$
- Μετά τη μαθηματική ανάλυση ο στόχος μας είναι να εκτιμήσουμε το c_t για μια συλλογή εγγράφων και ένα ερώτημα

Θεωρητική Εκτίμηση των Πιθανοτήτων

- Για κάθε όρο t προσπαθούμε να εκτιμήσουμε το c_t
- Βασιζόμαστε στον ακόλουθο πίνακα:

	documents	relevant	nonrelevant	Total
Term present	$x_t = 1$	s	$\mathrm{df}_t - s$	df_t
Term absent	$x_t = 0$	$S - s$	$(N - \mathrm{df}_t) - (S - s)$	$N - \mathrm{df}_t$
	Total	S	$N - S$	N

- Επίσης, παίρνουμε:

$$p_t = s/S$$
$$u_t = (\mathrm{df}_t - s)/(N - S)$$

- Οπότε:

$$c_t = K(N, \mathrm{df}_t, S, s) = \log \frac{s/(S-s)}{(\mathrm{df}_t - s)/((N - \mathrm{df}_t) - (S - s))}$$

Θεωρητική Εκτίμηση των Πιθανοτήτων

$$c_t = K(N, \mathrm{df}_t, S, s) = \log \frac{s/(S-s)}{(\mathrm{df}_t - s)/((N - \mathrm{df}_t) - (S-s))}$$

- Κατά τον παραπάνω υπολογισμό μπορεί να προκύψει πιθανότητα ίση με το 0
- Οπότε εφαρμόζουμε την ακόλουθη τεχνική (το κάτω δεξιά κελί στον προηγούμενο πίνακα αθροίζει σε N+2):

$$\hat{c}_t = K(N, \mathrm{df}_t, S, s) = \log \frac{(s+\frac{1}{2})/(S-s+\frac{1}{2})}{(\mathrm{df}_t - s + \frac{1}{2})/(N - \mathrm{df}_t - S + s + \frac{1}{2})}$$

Πρακτική Εκτίμηση των Πιθανοτήτων

- Πρακτικά υιοθετούμε στατιστικά στοιχεία για την εκτίμηση των προηγούμενων ποσοτήτων
- Έχουμε:
  - $u_t=df_t/N$ (πιθανότητα ύπαρξης του όρου σε άσχετο έγγραφο)

$$\log[(1-u_t)/u_t] = \log[(N-\mathrm{df}_t)/\mathrm{df}_t] \approx \log N/\mathrm{df}_t$$

Πρακτική Εκτίμηση των Πιθανοτήτων

- Το p_t μπορεί να εκτιμηθεί με διάφορους τρόπους:
  - Μπορούμε να υιοθετήσουμε τη συχνότητα των όρων σε γνωστά σχετικά έγγραφα
  - Μπορούμε να υποθέσουμε ότι είναι σταθερά για όλους τους όρους (π.χ. 0.5 – ισοπίθανη παρουσία σε σχετικά ή άσχετα έγγραφα) – μπορεί να αποδειχθεί καλή προσέγγιση σε περιπτώσεις μικρών εγγράφων
  - Μπορούμε να υιοθετήσουμε ένα απλό μαθηματικό τύπο για τον υπολογισμό (π.χ. $\frac{1}{3} + \frac{2}{3}\frac{df_t}{N}$

Πρακτική Εκτίμηση των Πιθανοτήτων

- Για την εμπλοκή του p_t στην ανατροφοδότηση συσχέτισης ακολουθούνται τα ακόλουθα βήματα:
  1. Αρχικά, μαντεύουμε ή υιοθετούμε μια αρχική σταθερά για τα u_t, p_t
  2. Βασιζόμαστε στην αρχική εκτίμηση για να παράξουμε μια καλύτερη εκτίμηση για τη συλλογή εγγράφων – κάποια έγγραφα τα παρουσιάζουμε στο χρήστη
  3. Αλληλεπιδρούμε με το χρήστη – μαθαίνουμε από το input του χρήστη και χωρίζουμε τη συλλογή V (έγγραφα που έχει κρίνει ο χρήστης) σε δύο αμοιβαία αποκλειόμενα σύνολα, $VR=\{d \in V, R_{d,q}=1\} \subset R, VNR=\{d \in V, R_{d,q}=0\}$
  4. Ξαναεκτιμούμε τα u_t, p_t βασιζόμενοι στα δύο σύνολα ως ακολούθως:

$$p_t = |VR_t|/|VR|$$

$$p_t = \frac{|VR_t| + \frac{1}{2}}{|VR| + 1}$$

Πρακτική Εκτίμηση των Πιθανοτήτων

- Διαδικασία εκτίμησης (συνέχεια)
  4. Εκτελούμε επαναληπτικά διάφορες ενημερώσεις των τιμών

$$p_t^{(k+1)} = \frac{|VR_t| + \kappa p_t^{(k)}}{|VR| + \kappa}$$

  5. Επαναλαμβάνουμε την προηγούμενη διαδικασία από το 2ο βήμα μέχρι ο χρήστης να μείνει ικανοποιημένος

Πρακτική Εκτίμηση των Πιθανοτήτων

- Στην ψευδο-ανατροφοδότηση συσχέτισης η διαδικασία έχει ως εξής:
  1. Ξεκινούμε με μια αρχική εκτίμηση όπως και πριν
  2. Καθορίζουμε μια εκτίμηση του πλήθους των σχετικών εγγράφων – αν δεν μπορούμε τότε παίρνουμε μια μικρή τιμή – παίρνουμε ως το V το σύνολο των εγγράφων με τα καλύτερα έγγραφα (υψηλότερο ranking)
  3. Συνεχίζουμε και βελτιώνουμε την εκτίμηση των p_t, u_t ως ακολούθως:

$$p_t = \frac{|V_t| + \frac{1}{2}}{|V| + 1}$$

$$u_t = \frac{\mathrm{df}_t - |V_t| + \frac{1}{2}}{N - |V| + 1}$$

  4. Επαναλαμβάνουμε τη διαδικασία από το 2ο βήμα μέχρι να συγκλίνουμε στο τελικό αποτέλεσμα

Πρακτική Εκτίμηση των Πιθανοτήτων

- Βασιζόμενοι σε όλα τα προηγούμενα μπορούμε να εκτιμήσουμε το c_t ως εξής:

$$c_t = \log\left[\frac{p_t}{1-p_t}\cdot\frac{1-u_t}{u_t}\right] \approx \log\left[\frac{|V_t|+\frac{1}{2}}{|V|-|V_t|+1}\cdot\frac{N}{\mathrm{df}_t}\right]$$

$$c_t = \log\frac{|V_t|+\frac{1}{2}}{|V|-|V_t|+1} + \log\frac{N}{\mathrm{df}_t}$$

Μοντέλο Okapi BM25

- Το BIM μοντέλο σχεδιάστηκε για μικρά τμήματα εγγράφων
- Στα σύγχρονα συστήματα είναι ξεκάθαρο πως οποιοδήποτε μοντέλο πρέπει να δίνει σημασία στη συχνότητα των όρων καθώς και στο μήκος των εγγράφων όπως έχουμε ήδη δει
- Το μοντέλο BM25 ή Okapi weighting σχεδιάστηκε ώστε στο πιθανοτικοθεωρητικό μοντέλο να λάβουμε υπόψιν μας τις προαναφερόμενες παραμέτρους

Μοντέλο Okapi BM25

- Η πιο απλή συνάρτηση για τον υπολογισμό του σκορ ενός εγγράφου είναι να λάβουμε υπόψιν μας το idf

$$RSV_d = \sum_{t \in q} \log \frac{N}{\mathrm{df}_t}$$

- Επίσης, μπορεί να υιοθετηθεί μια εναλλακτική τεχνική:

$$RSV_d = \sum_{t \in q} \log \frac{N - \mathrm{df}_t + \frac{1}{2}}{\mathrm{df}_t + \frac{1}{2}}$$

- Αν ένας όρος απαντάται σε πάνω από τα μισά έγγραφα τότε θα πάρουμε αρνητική τιμή – δεν επιθυμούμε μια τέτοια προσέγγιση
- Αν υιοθετήσουμε μια λίστα από stop words αυτό γενικά δεν θα συμβεί

Μοντέλο Okapi BM25

- Παράγουμε μια διαφορετική μορφή της συνάρτησης και εμπλέκουμε τη συχνότητα κάθε όρου και το μήκος των εγγράφων:

$$RSV_d = \sum_{t \in q} \log \left[\frac{N}{\mathrm{df}_t} \right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1 - b) + b \times (L_d / L_{\mathrm{ave}})) + \mathrm{tf}_{td}}$$

- L_d είναι το μήκος του εγγράφου, L_{ave} είναι το μέσο μήκος των εγγράφων
- Η μεταβλητή k1 είναι μια θετική σταθερά που υιοθετείται για σκοπούς calibration
- Το 0<=b<1 είναι επίσης μια tuning παράμετρος
- Αν b=1 παίρνουμε πλήρως υπόψιν μας το μέγεθος των εγγράφων

Μοντέλο Okapi BM25

- Αν το ερώτημα είναι αρκετά μεγάλο μπορούμε να υιοθετήσουμε μια παραπλήσια προσέγγιση και για την παρουσία των όρων μέσα στα ερωτήματα
- Οπότε καταλήγουμε στην ακόλουθη έξίσωση:

$$RSV_d = \sum_{t \in q} \left[\log \frac{N}{\mathrm{df}_t} \right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1 - b) + b \times (L_d / L_{\mathrm{ave}})) + \mathrm{tf}_{td}} \cdot \frac{(k_3 + 1)\mathrm{tf}_{tq}}{k_3 + \mathrm{tf}_{tq}}$$

- Το k3 είναι μια επιπρόσθετη παράμετρος
- Για τα ερωτήματα δεν έχουμε κανονικοποίηση ως προς το μήκος
- Συνήθως παίρνουμε: k1 & k3 στο [1.2,2] και b=0.75

Μοντέλο Okapi BM25

- Αν έχουμε και κρίσεις των χρηστών στη διάθεσή μας τότε έχουμε:

$$RSV_d = \sum_{t \in q} \log \left[\left[\frac{(|VR_t| + \frac{1}{2}) / (|VNR_t| + \frac{1}{2})}{(\mathrm{df}_t - |VR_t| + \frac{1}{2}) / (N - \mathrm{df}_t - |VR| + |VR_t| + \frac{1}{2})} \times \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1-b) + b(L_d / L_{\mathrm{ave}})) + \mathrm{tf}_{td}} \times \frac{(k_3 + 1)\mathrm{tf}_{tq}}{k_3 + \mathrm{tf}_{tq}} \right] \right]$$