

## Αναφορά – Β φάση

### Υλοποίηση

#### A) Αλλαγές σε σχέση με την Α φάση

I. Εγινε διάσπαση του documents.idx σε 2 αρχεία documents.idx, documents\_meta.idx. Τα περιεχόμενα τους έχουν ως:

- Documents\_meta.idx:

Doc Id

Weight

Max frequency

Document length (number of tokens)

Citations pagerank

Authors pagerank

Size of document (size of fields in documents.idx)

Offset to documents.idx

- Documents.idx:

Year

Title size

Author size

Author IDs size

Journal name size

Title

Author

Author IDs

Journal name

Το documents\_meta.idx φορτώνεται πάντα στη μνήμη όποτε κρίνεται απαραίτητο ενώ το documents.idx μένει στο δίσκο. Υποστηρίζεται επίσης η δυνατότητα να φορτωθεί και το documents.idx στη μνήμη αν και αυτό απαιτεί επιπλέον αλλαγές στον κώδικα.

Στο postings.idx αποθηκεύονται μόνο τα TF και offsets στο documents\_meta.idx

II. Εγινε αντικατάσταση της συνάρτησης βαθμολόγησης BM25 με την BM25+ η οποία:

- α. Λύνει το πρόβλημα της BM25 που δίνει αρνητικά σκορ για όρους που εμφανίζονται σε πάνω από τα μισά documents της συλλογής.
- β. Μειώνει το πέναλτι στο σκορ για documents τα οποία έχουν πάρα πολλούς όρους.

#### B) Εκτέλεση

Η κεντρική κλάση είναι η Themis από την οποία έχουμε τη δυνατότητα:

- Είτε να συμπληρώσουμε κώδικα και να καλέσουμε τις κλάσεις που εμείς θέλουμε.
- Είτε να περάσουμε ως πρώτη παράμετρο το string 'gui' ώστε να ενεργοποιήσουμε το GUI που παρέχει πρόσβαση στις περισσότερες λειτουργίες που χρειαζόμαστε.

Για όσες παραμέτρους δεν υπάρχει η δυνατότητα αλλαγής από το GUI, η αλλαγή γίνεται από το config file.

### C) Config

Το αρχείο διαβάζεται πάντα εκ νέου όταν:

- Φορτώνεται το index στη μνήμη.
- Δημιουργείται νέο index.

Επιπλέον, προστέθηκαν οι παρακάτω παράμετροι με τις αντίστοιχες default τιμές.

- PAGERANK\_THRESHOLD = 1E-8  
Μέγιστη τιμή της διαφοράς μεταξύ δύο σκορ για διαδοχικά iterations ώστε να θεωρήσουμε ότι ο Pagerank έχει συγκλίνει.
- PAGERANK\_DAMPING\_FACTOR = 0.85  
Πιθανότητα να μείνει ο random surfer στο ίδιο document.
- CITATIONS\_GRAPH\_PATH = D:/s2/citations\_graph/  
Αποθήκευση του γράφου των Citations.
- INDEX\_TMP\_PATH = /home/team2/index\_tmp/  
Φάκελος για την αποθήκευση οποιουδήποτε προσωρινού αρχείου παράγεται κατά την δημιουργία των μερικών Indexes. Σβήνεται μετά το πέρας της δημιουργίας του τελικού index.
- DOCUMENTS\_META\_FILENAME=documents\_meta.idx  
Το όνομα του αρχείου που περιγράφηκε στο **A)**
- QUERY\_EXPANSION\_ENABLED=false  
Ενεργοποίηση ή όχι κάποιου query expansion.
- QUERY\_EXPANSION\_MODEL=Glove  
Το προκαθορισμένο query expansion.
- EVALUATION\_FILENAME=evaluation.txt  
Όνομα του αρχείου για την αποθήκευση των αποτελεσμάτων της αξιολόγησης.

### D) Αναζήτηση (Κλάση Search)

a) Ο χρήστης μπορεί να διαλέξει να δει οποιαδήποτε από τα παρακάτω πεδία για ένα document.

TITLE, AUTHORS\_NAMES, AUTHORS\_IDS, YEAR, JOURNAL\_NAME,  
CITATIONS\_PAGERANK, VSM\_WEIGHT, TOKEN\_COUNT, MAX\_TF, DOCUMENT\_SIZE

- b) Πριν κάθε αναζήτηση δε γίνεται text preprocessing παρά μόνο απλή διάσπαση του query σε tokens και εφαρμογή stemming/storwords (μόνο αν αυτά έχουν γίνει και στο index).
- c) Για την ώρα, η αναζήτηση μέσω GUI εμφανίζει μόνο τα top-10 αποτελέσματα και δεν υπάρχει δυνατότητα αλλαγής αυτής της τιμής μέσω GUI. Οπότε, για την εμφάνιση περισσότερων αποτελεσμάτων μπορεί είτε να γίνει απευθείας αλλαγή στον κώδικα της κεντρικής κλάσης Themis, είτε να χρησιμοποιηθεί η μη GUI εκδοχή όπου μπορούμε να καθορίσουμε το πλήθος αποτελεσμάτων που θέλουμε.

- d) Αναζήτηση μπορεί να γίνει χρησιμοποιώντας παράλληλα και κάποιο query expansion dictionary. Για κάθε όρο του αρχικού query κρατάμε 1 επιπλέον νέο όρο με βάρος 0.5.
- e) Για τη δημιουργία του τελικού query, οι όροι που εμφανίζονται πολλαπλές φορές συνεννώνονται σε ένα όρο ο οποίος έχει τελικό βάρος το άθροισμα όλων των βαρών.

Για την ώρα η αναζήτηση είναι αρκετά απαιτητική λόγω του ότι όλα τα αποτελέσματα επιστρέφονται (είναι σύνηθες να υπάρχουν >1M αποτελέσματα).

## E) Αξιολόγηση

Κάθε νέα αξιολόγηση σώζεται στο INDEX\_PATH με το προκαθορισμένο όνομα που εμφανίζεται στο config file. Ο διαχωρισμός των διαφορετικών evaluation μεταξύ τους γίνεται απλά ως:

- Κάθε αρχείο έχει το timestamp της ημερομηνίας που ξεκίνησε η αξιολόγηση στο τέλος του ονόματος του.
- Κάθε αρχείο περιλαμβάνει στην αρχή του όλο το configuration για τη συγκεκριμένη αξιολόγηση, πχ stemming, query expansion model κλπ.

## F) Indexing

Κατά τη διαδικασία του indexing παράγονται διάφορα προσωρινά αρχεία τα οποία χρησιμοποιούνται σε μετέπειτα στάδια με κύριο στόχο τη μείωση του συνολικού χρόνου και της μνήμης που απαιτείται. Η σειρά των βημάτων είναι τέτοια ώστε να ελαχιστοποιείται το overhead σε χώρο στη μνήμη/δίσκο.

Αναλυτικά η διαδικασία έχει ως:

1. Ανάγνωση των αρχείων της συλλογής. Για κάθε document γίνονται οι κατάλληλες εγγραφές στα documents.idx, documents\_meta.idx ενώ κρατάμε στη μνήμη το partial vocabulary/posting. Παράλληλα για κάθε document γίνεται εγγραφή σε προσωρινό αρχείο όλων των <term, TF>. Το αρχείο αυτό χρησιμοποιείται κατά τον υπολογισμό των βαρών ώστε να μη χρειαστεί να ξαναδιατρέξουμε τη συλλογή.
2. Ταξινόμηση του partial vocabulary και dump των partial vocabulary/posting στο δίσκο.
3. Μόλις ολοκληρωθεί το dump όλων των partial vocabularies/postings γράφεται και όλο το configuration για το indexing στο meta.idx
4. Ακολουθεί το merge των των partial vocabularies το οποίο γίνεται σε ένα πέρασμα. Παράλληλα για κάθε όρο γίνεται εγγραφή σε προσωρινό αρχείο όλων των <partial index ID, DF> στα οποία εμφανίζεται. Το αρχείο αυτό χρησιμοποιείται κατά το merge των partial postings. Επιπλέον οι εγγραφές είναι ταξινομημένες ανά αύξουσα σειρά index ID ώστε όλα τα Postings για κάθε όρο να μπου και αυτά ταξινομημένα ανά αύξουσα τιμή offset στο postings.idx
5. Τα partial vocabularies σβήνονται όλα μετά το τέλος του merge τους.
6. Ακολουθεί ο υπολογισμός των βαρών. Εδώ χρησιμοποιείται το προσωρινό αρχείο που δημιουργήθηκε στο 1 το οποίο και σβήνεται μετά το πέρας της διαδικασίας. Για την επιτάχυνση της διαδικασίας το documents\_meta.idx γίνεται map στη μνήμη.
7. Ακολουθεί το merge των partial postings σε ένα πέρασμα. Εδώ χρησιμοποιείται το προσωρινό αρχείο που δημιουργήθηκε στο 4 το οποίο και σβήνεται μετά το πέρας της διαδικασίας.
8. Τα partial postings σβήνονται όλα μετά το τέλος του merge τους.
9. Υπολογισμός των Pagerank σκορ.

## G) Pagerank

Η διαδικασία περιλαμβάνει μόνο τον υπολογισμό των σκορ με βάση το γράφο των citations.

Για την μείωση των απαιτήσεων σε μνήμη και χρόνο, ακολουθήκαν τα παρακάτω βήματα:

1. Αρχικά φορτώνεται το `documents_meta.idx` στη μνήμη και κάθε document ID αντιστοιχείται σε ένα ακέραιο ID ο οποίος είναι η αύξουσα θέση του document ID στο `documents_meta.idx`.
2. Στη συνέχεια γίνεται ένα πέρασμα πάνω από τη συλλογή το οποίο έχει διπλό σκοπό. 1<sup>ον</sup> να φιλτράρει τα document ID τα οποία δεν εμφανίζονται καθόλου στη συλλογή. 2<sup>ον</sup> να γράψει όλα τα απαραίτητα δεδομένα του γράφου σε προσωρινό αρχείο στο δίσκο, δηλαδή, για κάθε κόμβο τον αριθμό των Out edges και μια λίστα με τα In edges (ακέραιοι ID).
3. Η δομή που φτιάξαμε στο 1 δε χρειάζεται περαιτέρω, οπότε μπορούμε να αρχικοποιήσουμε το γράφο διαβάζοντας τα δεδομένα που γράψαμε στο 2. Ο γράφος αποτελείται από ένα απλό array από κόμβους οι οποίοι εμφανίζονται στο array με την ίδια σειρά που εμφανίζονται στο `documents_meta.idx` αφού με την ίδια σειρά εμφανίζονται και τα δεδομένα τους στο αρχείο που γράψαμε στο 2.
4. Τέλος, τρέχουμε τον αλγόριθμο μέχρι να συγκλίνουν τα σκορ και επιστρέφουμε μόνο ένα τελικό array με τα σκορ. Πάλι η σειρά εμφάνισης των σκορ αντιστοιχεί στη σειρά εμφάνισης των κόμβων που αντιστοιχεί στη σειρά εμφάνισης των documents στο `documents_meta.idx`.
5. Ο γράφος δε χρειάζεται πια, οπότε φορτώνουμε το `documents_meta.idx` στη μνήμη και γράφουμε τα τελικά σκορ.

Επιπλέον στον γράφο δε συμπεριλαμβάνονται ακμές που δείχνουν στον ίδιο κόμβο ενώ διπλές ακμές προς άλλους κόμβους θεωρούνται ως μια ακμή.

## H) Επιπλέον λειτουργικότητα

- Κλάση `S2TextualEntryCharHistogram`: Διαβάζει μια συλλογή και δημιουργεί ένα ιστόγραμμα των χαρακτήρων που εμφανίζονται σε κάθε document και της συχνότητας εμφάνισης τους. Είναι χρήσιμη αν θέλουμε να βρούμε delimiters για tokenization των documents.
- Κλάση `CreateCitationsGraph`: Διαβάζει μια συλλογή, δημιουργεί τον πλήρες γράφο των citations και ταυτόχρονα υπολογίζει στατιστικά στοιχεία για αυτόν.

## Indexing the whole collection (108GB, 47M documents)

Κάθε partial index περιλάμβανε 500K documents και η ευρετηρίαση έγινε εξ ολοκλήρου στον SSD. Τα stowords, stemming είναι ενεργοποιημένα και στην ευρετηρίαση αλλά και στην αξιολόγηση.

### Μεγέθη τελικών αρχείων:

- Documents\_meta.idx: 3760MB
- Documents.idx: 9700MB
- Vocabulary.idx: 358MB
- Postings.idx: 40345MB

Σύνολο: 54163MB

Για την ευρετηρίαση χρειάστηκαν επιπλέον ~42237MB στο δίσκο (πέραν του χώρου που καταλαμβάνει το τελικό Index).

### Χρόνοι:

- Partial indexes + documents.idx + documents\_meta.idx: 7578s
- Merge partial vocabularies: 67s
- Merge partial postings: 211s
- Weights: 1053s
- Create pagerank graph: 2580s
- Pagerank iterations: 45 iterations, total time 436s

Σύνολο: 11925s = 3h 19m

### Pagerank:

Χρησιμοποιήθηκε damping factor = 0.85 και threshold =  $1e-8 < 1/\text{\#documents} = 2e-8$

Επιπλέον, ο γράφος των Publications φαίνεται να είναι αραιός: 30M sinks και 348M Out ακμές.

**Όλα τα αποτελέσματα βρίσκονται στο φάκελο results.**

## Evaluation

Ο χρόνος αναζήτησης ανά 1M αποτελέσματα χωρίς τη χρήση query expansion κυμαίνεται μεταξύ 2.2 και 2.6s

**Query expansion: No, Pagerank: No, Stemming: Yes, Stopwords: Yes**

	VSM	Okapi
Average precision	Average: 0.69896 Min: 0 Max: 1	Average: 0.70528 Min: 0 Max: 1
nDCG	Average: 0.80272 Min: 0 Max: 1	Average: 0.80716 Min: 0 Max: 1
Time	Total: 1.0h 17.0m	Total: 57.0m 20.0s

**Query expansion: extJWNL, Pagerank: No, Stemming: Yes, Stopwords: Yes**

	VSM	Okapi
Average precision	Average: 0.69933 Min: 0 Max: 1	Average: 0.70298 Min: 0 Max: 1
nDCG	Average: 0.8032 Min: 0 Max: 1	Average: 0.80549 Min: 0 Max: 1
Time	Total: 1.0h 50.0m	Total: 1.0h 26.0m

**Query expansion: Glove, Pagerank: No, Stemming: Yes, Stopwords: Yes**

	VSM	Okapi
Average precision	Average: 0.70014 Min: 0 Max: 1	Average: 0.69573 Min: 0 Max: 1
nDCG	Average: 0.80393 Min: 0 Max: 1	Average: 0.79997 Min: 0 Max: 1
Time	Total: 1.0h 46.0m	Total: 1.0h 24.0m

**Query expansion: No, Pagerank weight: 0.25, Model weight: 0.75, Stemming: Yes, Stopwords: Yes**

	VSM	Okapi
Average precision	Average: 0.70206 Min: 0 Max: 1	Average: 0.71684 Min: 0 Max: 1
nDCG	Average: 0.80574 Min: 0 Max: 1	Average: 0.81673 Min: 0 Max: 1
Time	Total: 1.0h 9.0m	Total: 1.0h 7.0m

## Συμπεράσματα:

- A. Τα query expansion models όπως και η χρήση Pagerank δε φαίνεται να έχουν σημαντική επίπτωση στα τελικά σκορ. Παρακάτω φαίνονται οι διαφορές σε σχέση με τα αποτελέσματα της αρχικής αξιολόγησης (με bold).

	Average of Average precision	Average of nDCG
<b>VSM</b>	<b>0.69896</b>	<b>0.80272</b>
VSM/Glove	0.70014 (+0.118%)	0.80393 (+0.121%)
VSM/extJWNL	0.69933 (+0.037%)	0.8032 (+0.048%)
VSM/Pagerank	0.70206 (+0.31%)	0.80574 (+0.302%)
<b>Okapi</b>	<b>0.70528</b>	<b>0.80716</b>
Okapi/Glove	0.69573 (-0.955%)	0.79997 (-0.719%)
Okapi/extJWNL	0.70298 (-0.23%)	0.80549 (-0.167%)
Okapi/Pagerank	0.71684 (+1.156%)	0.81673 (+0.957%)

- B. Απ' τη στιγμή που οι διαφορές μεταξύ των σκορ ανά μέθοδο αναζήτησης είναι πολύ μικρές, η παρακάτω ανάλυση αφορά όλες τις μεθόδους συγκεντρωτικά και είναι μόνο για τα queries με μηδενικό σκορ:
- Ψάχνουμε όρους που είναι stopwords. Για παράδειγμα, υπάρχει query 'computer' το οποίο τυχαίνει να είναι stopword. Χρειάζεται καλύτερη επιλογή των όρων που επιλέγονται ως stopwords.
  - Οι όροι του query δεν εμφανίζονται στα documents που έχουν κριθεί ως συναφή, πχ. υπάρχει query 'genomestrip' αλλά μόνο ο όρος 'genome' εμφανίζεται σε αυτά. Εδώ ενδεχομένως να βοηθούσε κάποιο pattern matching.
  - Τα documents που έχουν κριθεί ως συναφή δεν σχετικά μεταξύ τους και οι όροι του query δεν εμφανίζονται σε αυτά. Εδώ δεν υπάρχει κάποια προφανής λύση.
  - Οι όροι του query εμφανίζονται στα documents διατυπωμένοι διαφορετικά, πχ ψάχνουμε για 'emule' ενώ στα documents εμφανίζεται ο όρος 'eDonkey'. Εδώ ενδεχομένως να βοηθούσε ένα πιο σωστό query expansion.
  - Δεν υπάρχει επαρκής πληροφορία στα documents που έχουν κριθεί ως συναφή που να επιτρέπει να κάνουμε σωστή αναζήτηση, πχ. η μόνη πληροφορία που εμφανίζεται είναι abstract: 'This collaboratively edited knowledgebase provides a common source of data for Wikipedia, and everyone else' και title: 'Wikidata: a free collaborative knowledgebase' και ο όρος αναζήτησης είναι 'blazegraph' ο οποίος αφορά μια συγκεκριμένη graph database. Εδώ ενδεχομένως να βοηθούσε ένα πιο σωστό query expansion αν και στη συγκεκριμένη περίπτωση μάλλον δε θα ήταν χρήσιμο.
  - Οι όροι του query δεν εμφανίζονται στα documents που έχουν κριθεί ως συναφή και μόνο από το περιεχόμενο τους μπορούμε να συμπεράνουμε ότι είναι συναφή. Πχ ψάχνουμε για 'spirotube' το οποίο είναι το όνομα ενός device για χρήση σε ασθενείς με άσθμα ενώ τα συναφή documents αναφέρουν μόνο τον όρο 'asthma'. Εδώ ενδεχομένως να βοηθούσε ένα πιο σωστό query expansion.
  - Οι όροι του query έχουν ορθογραφικά λάθη, πχ. 'similarity' αντί για 'similarity'. Εδώ θα βοηθούσε κάποιο pattern matching.
  - Οι όροι του query εμφανίζονται στα documents που έχουν κριθεί ως συναφή ανακατεμένοι με άλλα σύμβολα πχ ψάχνουμε για 'Thiahelicene' ενώ στα documents εμφανίζεται ο 'tetrathia[9]helicene' τον οποίο εμείς σπάσαμε κατά τη διάρκεια του Indexing σε 'tetrathia', '9', 'helicene'. Εδώ ενδεχομένως να βοηθούσε κάποιο pattern matching ή proximity search.

C. Άλλα πιθανά σενάρια που δικαιολογούν χαμηλά σκορ:

- Αγνοούμε κάποια split delimiters. Εδώ θα βοηθούσε κάποιο pattern matching ή η αύξηση των split delimiters.
- Χρησιμοποιούμε πάρα πολλά split delimiters. Αυτό είναι πιθανό, πχ σπάσαμε hyphenated λέξεις στα δύο ενώ δεν έπρεπε. Εδώ ενδεχομένως να βοηθούσε κάποιο proximity search ή η αποφυγή του διαχωρισμού hyphenated λέξεων.
- Η χρήση hashmap για το vocabulary μας περιορίζει σε ένα απλό hit or miss των όρων αναζήτησης. Αρα ενδεχομένως να επιστρέφουμε στις αρχικές θέσεις documents τα οποία περιέχουν τους όρους της αναζήτησης ενώ τα αυτά τα documents έχουν κριθεί ως μη συναφή. Άλλες δομές αναζήτησης είναι προτιμότερες, πχ. suffix arrays, ώστε να υποστηρίξουμε και άλλες λειτουργίες πάνω σε strings.
- Ο γράφος στο Pagerank είναι αραιός και δε βοηθάει όσο θα περίμενε κανείς.
- Δε γίνεται σωστό query expansion. Αυτό είναι πιθανόν μιας και προσθέτουμε στη νέα λίστα των όρων αναζήτησης μόνο τον πρώτο νέο όρο του query expansion, για κάθε όρο του αρχικού query, χωρίς παραιτέρω ανάλυση.
- Δε γίνεται σωστό stemming, πχ. όροι που δε σχετίζονται μεταξύ τους καταλήγουν να έχουν την ίδια ρίζα.

Τέλος, θα μπορούσαμε να βελτιώσουμε τα αποτελέσματα με την εξαγωγή επιπλέον tokens κατά τη διάρκεια του indexing από το κύριο σώμα των documents (χρήση των json πεδίων s2PdfUrl/s2Url).