

HY463 Αναφορά Project – Β φάση

Υλοποίηση: Αναστάσιος Χατζηαλεξίου - CSD 3902

Υλοποίηση

A) Αλλαγές σε σχέση με την Α φάση

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
Offset to the 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.

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) Exceptions

Η διαχείριση των exceptions υπάρχει αλλά είναι τυπική. Αν ωστόσο δούμε κάποιο exception είναι προτιμώτερο να επανεκκινήσουμε το πρόγραμμα.

D) 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.

E) Αναζήτηση (Κλάση Search)

Για την ώρα η αναζήτηση μέσω GUI εμφανίζει μόνο τα top-10 αποτελέσματα. Γενικότερα η αναζήτηση υποστηρίζει την αναζήτηση/εμφάνιση αποτελεσμάτων σε οποιοδήποτε εύρος [A, B].

Αποτελέσματα όρων που εμφανίζονται σε μια αναζήτηση χρησιμοποιούνται και στην αμέσως επόμενη αναζήτηση αν εμφανίζονται ξανά οι ίδιοι όροι.

Αναζήτηση μπορεί να γίνει χρησιμοποιώντας παράλληλα και κάποιο query expansion model. Ολα επιστρέφουν 1 επιπλέον νέο όρο με βάρος 0.5 για κάθε όρο του αρχικού query.

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.

F) 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 όλες τις απαραίτητες επιλογές που διαφοροποιούν τις αξιολογήσεις, πχ stemming, query expansion model κλπ.

G) 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 ξαναπεράσουμε τη συλλογή.
2. Dump του partial vocabulary/posting στο δίσκο.
3. Μόλις ολοκληρωθεί το dump όλων των partial vocabularies/postings γράφονται και όλες οι πληροφορίες για το indexing στο meta.idx

4. Ακολουθεί το merge των των partial vocabularies το οποίο γίνεται σε ένα πέρασμα για εξοικονόμηση χρόνου. 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 μπουν και αυτά ταξινομημένα ανά αύξουσα τιμή offset στο documents_meta.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 φορτώνεται το documents_meta.idx 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.

Επιπλέον έχουν γίνει οι παρακάτω βελτιστοποιήσεις:

- Τα partial vocabulary κατά τη διάρκεια της δημιουργίας τους δεν υπάρχει λόγος να βρίσκονται στη μνήμη ταξινομημένα. Η ταξινόμηση τους γίνεται πριν την εγγραφή τους στο δίσκο.
- Τα partial vocabularies/postings βρίσκονται στη μνήμη σε μια ενιαία δομή.

H) Pagerank

Λόγω του ότι ο υπολογισμός των σκορ είναι αρκετά απαιτητική διαδικασία, για να μειωθούν οι απαιτήσεις χώρου/χρόνου ακολουθήκαν τα παρακάτω βήματα:

1. Αρχικά φορτώνεται το documents_meta.idx στη μνήμη και κάθε document ID αντιστοιχείται σε ένα ακέραιο ID.
2. Στη συνέχεια γίνεται ένα πέρασμα πάνω από τη συλλογή το οποίο έχει διπλό σκοπό. 1^{ον} να φιλτράρει τα document ID τα οποία δεν εμφανίζονται καθόλου στη συλλογή. 2^{ον} 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 πια, οπότε μπορούμε να αρχικοποιήσουμε το γράφο διαβάζοντας τα δεδομένα που γράψαμε στο 2. Ο γράφος αποτελείται από ένα απλό array από κόμβους, με το ID κόμβου να αντιστοιχεί στη θέση του κόμβου στο array. Αυτό σημαίνει ότι κάθε κόμβος δεν χρειάζεται να περιλαμβάνει κάποιο ID από τη στιγμή που διαβάζουμε τα documents πάντα με την ίδια σειρά.
4. Τέλος, τρέχουμε τον αλγόριθμο μέχρι να συγκλίνουν τα σκορ και επιστρέφουμε μόνο ένα τελικό array με τα σκορ.

5. Ο γράφος δε χρειάζεται πια, οπότε φορτώνουμε το `documents_meta.idx` στη μνήμη και γράφουμε τα τελικά σκορ.

Το σκορ για τους κόμβους με 0 Out edges μοιράζεται εξίσου σε όλους τους κόμβους σε κάθε iteration. Επιπλέον στον γράφο δε συμπεριλήφθηκαν ακμές που δείχνουν στον ίδιο κόμβο ενώ διπλές ακμές προς άλλους κόμβους θεωρήθηκαν ως μια ακμή.

I) 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. Στον γράφο δεν συμπεριλαμβάνονται ακμές που δείχνουν στον ίδιο κόμβο ενώ διπλές ακμές προς άλλους κόμβους θεωρούνται ως μια ακμή.

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

Κάθε partial index περιλάμβανε 500K documents και η ευρετηρίαση έγινε εξ ολοκλήρου στον SSD. Τα stopwords, stemming είναι πάντα ενεργοποιημένα και στην ευρετηρίαση αλλά και στην αξιολόγηση. Τέλος για τον Pagerank χρησιμοποιήθηκε damping factor = 0.85 και threshold = $1e-8 < 1/\text{\#documents} = 2e-8$.

Μεγέθη αρχείων:

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

Σύνολο: 54163MB

Μέγιστο μέγεθος αρχείων στο δίσκο: $\sim 96400\text{MB} = 1.78 * (\text{index size})$

Χρόνοι:

- Partial indexes + documents.idx + documents_meta.idx: 7578s
- Merge partial vocabularies: 67s
- Merge partial postings: 211s
- Weights: 1053s
- Pagerank graph: 2580s
- Pagerank iterations: 45 iterations, 436s

Σύνολο: 11925s = 3h 19m

Pagerank:

Ο γράφος των Publications φαίνεται να είναι αραιός με 30M sinks και 348M Out ακμές.

Όλα τα αποτελέσματα βρίσκονται στο φάκελο results.

Evaluation

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

	VSM	Okapi
Average precision	Average: 0.69896 Min: 0 Max: 1	Average: 0.70599 Min: 0 Max: 1
nDCG	Average: 0.80272 Min: 0 Max: 1	Average: 0.80724 Min: 0 Max: 1
Time	Total: 1.0h 17.0m Average per query: 7.29s	Total: 59.0m 32.0s Average per query: 5.63s

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

	VSM	Okapi
Average precision	Average: 0.69933 Min: 0 Max: 1	Average: 0.70377 Min: 0 Max: 1
nDCG	Average: 0.80319 Min: 0 Max: 1	Average: 0.80588 Min: 0 Max: 1
Time	Total: 1.0h 50.0m Average per query: 10.4s	Total: 1.0h 21.0m Average per query: 7.65s

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

	VSM	Okapi
Average precision	Average: 0.70014 Min: 0 Max: 1	Average: 0.69738 Min: 0 Max: 1
nDCG	Average: 0.80392 Min: 0 Max: 1	Average: 0.801 Min: 0 Max: 1
Time	Total: 1.0h 46.0m Average per query: 10.0s	Total: 1.0h 30.0m Average per query: 8.47s

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: 0.71631 Min: 0 Max: 1
nDCG	Average: 0.80574 Min: 0 Max: 1	Average: 0.81606 Min: 0 Max: 1
Time	Total: 1.0h 9.0m Average per query: 6.56s	Total: 1.0h 6.0m Average per query: 6.26s

Συμπεράσματα:

- A. Ο μέσος χρόνος αναζήτησης ανά 1M αποτελέσματα χωρίς τη χρήση query expansion είναι 2.6s
- B. 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.

	Average of Average precision	Average of nDCG
VSM	0.69896	0.80272
Okapi	0.70599	0.80724
VSM/Glove	0.70014 (+0.118%)	0.80392 (+0.12%)
Okapi/Glove	0.69738 (-0.861%)	0.801 (-0.624%)
VSM/extJWNL	0.69933 (+0.037%)	0.80319 (+0.047%)
Okapi/extJWNL	0.70377 (-0.222%)	0.80588 (-0.136%)
VSM/Pagerank	0.70206 (+0.31%)	0.80574 (+0.302%)
Okapi/Pagerank	0.71631 (+1.032%)	0.81606 (+0.882%)

C. Μερικοί λόγοι που επηρεάζουν τα σκορ είναι:

- Ψάχνουμε όρους που είναι stopwords. Για παράδειγμα, υπάρχει query 'computer' το οποίο τυχαίνει να είναι stopwords.
- Οι όροι του query δεν εμφανίζονται στα σχετικά documents πχ. υπάρχει query 'genomestrip' αλλά μόνο ο όρος 'genome' εμφανίζεται στα σχετικά documents. Εδώ ενδεχομένως να βοηθούσε κάποιο pattern matching.

- Τα σχετικά documents δεν σχετικά μεταξύ τους και παράλληλα οι όροι του query δεν εμφανίζονται στα documents. 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 καλύτερο query expansion με χρήση συνωνύμων.
- Δεν υπάρχει επαρκής πληροφορία στα σχετικά documents που να επιτρέπει να κάνουμε σωστή αναζήτηση, πχ. η μόνη πληροφορία που εμφανίζεται είναι η 'This collaboratively edited knowledgebase provides a common source of data for Wikipedia, and everyone else' και 'Wikidata: a free collaborative knowledgebase' με όρο αναζήτησης 'blazegraph' ο οποίος αφορά μια συγκεκριμένη graph database. Εδώ ενδεχομένως να βοηθούσε ένα καλύτερο query expansion αν και στη συγκεκριμένη περίπτωση μάλλον δε θα ήταν χρήσιμο.
- Οι όροι του query δεν εμφανίζονται στα σχετικά documents και μόνο από το περιεχόμενο τους μπορούμε να συμπεράνουμε ότι είναι σχετικά. Πχ ψάχνουμε για 'spirotube' το οποίο είναι το όνομα ενός device για χρήση σε ασθενείς με άσθμα ενώ τα σχετικά documents αναφέρουν απλά 'asthma'. Εδώ ενδεχομένως να βοηθούσε ένα καλύτερο 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 σχετικά documents ανακατεμένοι με άλλα σύμβολα πχ ψάχνουμε για 'Thiahelicene' ενώ στα documents εμφανίζεται ο 'tetrathia[9]helicene' τον οποίο εμείς σπάσαμε σε 'tetrathia', '9', 'helicene'. Εδώ ενδεχομένως να βοηθούσε κάποιο pattern matching ή proximity search.
- Ο γράφος στο Pagerank είναι αραιός και δε βοηθάει όσο θα περίμενε κανείς.
- Τα query expansion, ειδικά το Glove, συχνά επιστρέφουν όρους που μπορεί να μην είναι αρκετά συναφείς με το περιεχόμενο του document. Εδώ χρειάζεται καλύτερη επιλογή των όρων που επιλέγονται για το τελικό query.

D. Άλλα πιθανά σενάρια που δικαιολογούν χαμηλά σκορ:

- Αγνοούμε κάποια split delimiters. Εδώ θα βοηθούσε κάποιο pattern matching ή η αύξηση των split delimiters.
- Χρησιμοποιούμε split delimiters σε υπερβολικό βαθμό. 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.
- Υπάρχουν ορθογραφικά λάθη στα documents. Μια απλή εξέταση των documents δείχνει ότι κάτι τέτοιο είναι συχνό φαινόμενο. Εδώ θα βοηθούσε κάποιο pattern matching.
- 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 προτιμώτερες, πχ. suffix arrays, ώστε να υποστηρίξουμε και άλλες λειτουργίες πάνω σε strings.

- Δε γίνεται σωστό stemming, δηλαδή, λέξεις που δεν σχετίζονται μεταξύ τους καταλήξουν να έχουν την ίδια ρίζα.

Τέλος, θα μπορούσαμε να βελτιώσουμε τα αποτελέσματα και με την εξαγωγή επιπλέον tokens από το κύριο σώμα των documents (s2PdfUrl/s2Url).