Αναφορά 2ης φάσης Εργασίας

Αλέξανδρος Βεντούρας AM: 3160013 email: alvent98@gmail.com

Στην παρούσα αναφορά περιγράφονται τα βήματα που ακολουθήθηκαν μέχρι και κατά την δεύτερη φάση (Επέκταση ερωτήματος με WordNet) της εργασίας του μαθήματος «Συστήματα Ανάκτησης Πληροφοριών».

Α. Πηγές κώδικα:

Οι πηγές που χρησιμοποιήθηκαν σε αυτό το στάδιο, είναι οι ίδιες με αυτές της πρώτης φάσης, δηλαδή το overview του Documentation της συγκεκριμένης έκδοσης του Lucene[1] (χρησιμοποιήθηκε η έκδοση 7.7.2.), το οποίο περιέχει δύο αρχεία, τα IndexFiles.java και SearchFiles.java, που αποτέλεσαν τη βάση για τη συγγραφή των παραδιδόμενων προγραμμάτων. Επιπλέον, αξιοποιήθηκε και ένα βίντεο στο youtube[2], βάσει των οδηγιών του οποίου προστέθηκαν ως external JAR's ορισμένα αρχεία της βιβλιοθήκης Lucene.

Επιπροσθέτως, όσον αφορά τον κώδικα που προστέθηκε στη 2η φάση, αυτός προέρχεται πρωτίστως από τον κώδικα που παρουσιάστηκε σε ένα σχετικό βίντεο, το οποίο αναρτήθηκε στο eclass του μαθήματος. Αναφορικά με την επεξεργασία του αρχείου wn_s.pl, αυτή έγινε με βάση ένα συγκεκριμένο άρθρο στο stackoverflow.com[3]. Ο tokenizer και τα tokenFilters τα οποία χρησιμοποιήθηκαν για την δημιουργία του CustomAnalyzer, επιλέχθηκαν με βάση το προαναφερθέν βίντεο της διδάσκουσας. Οι όποιες μετατροπές, έγιναν έπειτα από μελέτη της περιοχής συζητήσεων του eclass, καθώς και του documentation του Lucene των αντίστοιχων κλάσεων[4].

---

[1] https://lucene.apache.org/core/7_7_2/demo/overview-summary.html (τελευταία επίσκεψη στις 28-4-2020)

[2] https://www.youtube.com/watch?v=pVDVURw_AJQ (τελευταία επίσκεψη στις 28-4-2020)

[3] https://stackoverflow.com/questions/45401338/java-copy-from-a-file-to-another-line-by-line-with-an-interval (τελευταία επίσκεψη στις 22-5-2020)

[4] https://lucene.apache.org/core/7_4_0/analyzers-common/org/apache/lucene/analysis/core/WhitespaceTokenizer.html (τελευταία επίσκεψη στις 22-5-2020)
https://lucene.apache.org/core/7_3_1/analyzers-common/org/apache/lucene/analysis/en/EnglishPossessiveFilter.html (τελευταία επίσκεψη στις 22-5-2020)
https://lucene.apache.org/core/7_4_0/core/org/apache/lucene/analysis/LowerCaseFilter.html (τελευταία επίσκεψη στις 22-5-2020)
https://lucene.apache.org/core/7_2_0/core/org/apache/lucene/analysis/StopFilter.html (τελευταία επίσκεψη στις 22-5-2020)
https://lucene.apache.org/core/7_0_0/analyzers-common/index.html?org/apache/lucene/analysis/en/PorterStemFilter.html (τελευταία επίσκεψη στις 22-5-2020)
https://lucene.apache.org/core/7_0_0/analyzers-common/org/apache/lucene/analysis/synonym/SynonymGraphFilter.html(τελευταία επίσκεψη στις 22-5-2020)

Β. Περιγραφή υλοποίησης τριών πρώτων βημάτων – ανάλυση κώδικα:

Όσον αφορά τα τρία πρώτα βήματα, έγινε parsing του εγγράφου documents.txt κατά τον τρόπο που περιγράφεται στις ακόλουθες παραγράφους. Τα δύο πρώτα βήματα κατουσίαν παρέμειναν αναλλοίωτα, σε σχέση με την πρώτη φάση της εργασίας.

Αρχικά (αναφορικά με την δημιουργία του ευρετηρίου) αποθηκεύεται η πρώτη σειρά (σαν StringField με property "id"), η οποία είναι προφανές ότι περιέχει το id του πρώτου document (γραμμές 41 – 43). Έπειτα, γινόταν ανάγνωση κάθε επόμενης σειράς (γραμμή 45), και αν αυτή η σειρά περιείχε τρεις καθέτους συνεχόμενα (δηλαδή την ακολουθία '///' χωρίς τα εισαγωγικά), τότε αυτό σήμαινε ότι από την επόμενη γραμμή ξεκινάει καινούργιο document. Συνεπώς, αποθηκευόταν οι ήδη διαβασμένες γραμμές σε ένα TextField με property "contents" (γραμμή 51), και αν υπήρχε ήδη το document, τότε έπρεπε να γίνει append σε αυτό το νέο property, αλλιώς αν δεν υπήρχε, προστίθεντο στον writer ως νέο document (γραμμή 58). Αν προς το παρόν μόνο create γίνεται, η περίπτωση του append παρέμεινε, μήπως χρειαστεί σε επόμενη φάση της εργασίας. Έπειτα, διαβάζεται η επόμενη γραμμή και αποθηκεύεται το id του επόμενου κειμένου (γραμμές 79-81).

Σε διαφορετική περίπτωση, δηλαδή αν η διαβασμένη γραμμή δεν περιείχε την ακολουθία των τριπλών καθέτων, απλά περνούσαμε στην ανάγνωση της επόμενης γραμμής ,αφού πρώτα είχε προστεθεί η τρέχουσα γραμμή σε ένα string (γραμμή 83).

Τα παραπάνω γίνονται στην μέθοδο indexDocument της κλάσης IndexCreator. Στην main του συγκεκριμένου αρχείου (IndexCreator.java), καταχωρείται το path του txt που περιέχει τα κείμενα, τα οποία περιέχουν τις ζητούμενες πληροφορίες (γραμμή 97). Έπειτα, «ανοίγεται» το directory το οποίο περιέχει το txt με τις πληροφορίες (γραμμή 104), και αρχικοποιείται ο αναλυτής (Analyzer), ως στιγμιότυπο της ειδικότερης κλάσης EnglishAnalyzer (γραμμή 105). Επιλέχθηκε ο συγκεκριμένος Analyzer[5], προκειμένου να εκμεταλλευτούμε το γεγονός ότι εκτελεί αυτόματα την απαραίτητη γλωσσολογική επεξεργασία, όπως την αφαίρεση των stopwords, ή την αποκοπή των παραγωγικών καταλήξεων, με τη βοήθεια του Porter Stemmer.

Έπειτα, δημιουργείται ένας εγγραφέας ευρετηρίου (IndexWriter – γραμμή 112), που κατ' ουσίαν παραμετροποιείται με todirectory, το οποίο περιέχει το path του txt, καθώς και τον IndexWriterConfig, που εμπερικλείει τον EnglishAnalyzer, τον οποίο δημιουργήσαμε προηγουμένως. Τέλος, εκτελείται η μέθοδος indexDocument (γραμμή 115), που περιγράφεται παραπάνω.

Όσον αφορά την αναζήτηση, αυτή εκτελείται μέσω της μεθόδου search της κλάσης Searcher, του ομώνυμου αρχείου. Σε αυτή, αρχικά μέσω της κλήσης της μεθόδου search της κλάσης IndexSearcher (του Lucene), γίνεται η αναζήτηση ενός από τα queries, στο ευρετήριο που έχει ήδη δημιουργηθεί (γραμμή 39).

[5] Βάσει του συγκεκριμένου άρθρου, το οποίο αναλύει τα features κάθε διαθέσιμου Analyzer στη Lucene: https://www.baeldung.com/lucene-analyzers (τελευταία επίσκεψη στις 29-4-2020)

Έπειτα, αφού αποθηκευτούν τα αποτελέσματα της αναζήτησης (γραμμή 40), αποθηκεύονται σε ένα αρχείο τα στοιχεία κάθε document που επιτυγχάνει πάνω από ένα συγκεκριμένο αριθμό ευρέσεων (γραμμές 58-60 και 62).

Όσον αφορά το parsing των queries, αυτό γίνεται στη μέθοδο getQueries του αρχείου Searcher. Σε αυτή, διαβάζεται γραμμή-γραμμή το αρχείο που τα περιέχει, και αποθηκεύονται μόνο οι γραμμές που περιέχουν μόνο γράμματα ή και παύλες, κάθε μία γραμμή σε μία ξεχωριστή λίστα από Strings (γραμμές 77-78).

Επιπλέον, στη δεύτερη φάση της εργασίας, δημιουργήθηκε ένας CustomAnalyzer (στη μέθοδο που υπάρχει στις γραμμές 95 – 109), ο οποίος δημιουργεί τα tokens απομονώνοντας μία ακολουθία χαρακτήρων, η οποία περιέχεται μεταξύ δύο spaces (με τη βοήθεια του WhitespaceTokenizerFactory – γραμμή 101). Από κάθε ένα από αυτά τα tokens, αφαιρείται το κτητικό ‘s που υπάρχει σαν κατάληξη στην αγγλική γλώσσα (με τη βοήθεια του EnglishPossessiveFilterFactory – γραμμή 102), μετατρέπονται όλα τα γράμματα που περιέχει σε μικρά (με τη βοήθεια του LowerCaseFilterFactory – γραμμή 103), αν είναι τετριμμένη λέξη αφαιρείται (με τη βοήθεια του StopFilterFactory – γραμμή 104), αποκόπτεται η κατάληξη του token, βάσει του αλγόριθμου αποκοπής Porter Stemmer (με τη βοήθεια του PorterStemFilterFactory – γραμμή 105), και τέλος προστίθενται συνώνυμα σε αυτό (με τη βοήθεια του SynonymGraphFilterFactory και ενός HashMap, το οποίο περιέχει τα δεδομένα του αρχείου wn_s.pl – γραμμή 106).

Επιπροσθέτως, πάλι στη δεύτερη φάση της εργασίας, το αρχείο wn_s.pl επεξεργάζεται έτσι ώστε να απαλοιφούν από αυτό όλες οι εγγραφές που αφορούν ρήματα. Ειδικότερα, ανοίγεται το παλιό αρχείο που περιέχει όλες ανεξαιρέτως τις εγγραφές, και διαβάζεται κάθε γραμμή του (γραμμή 124). Αν η γραμμή αυτή περιέχει την ακολουθία ‘,v,’, η οποία υποδεικνύει πως η εγγραφή αυτή αφορά ρήμα, τότε η γραμμή αυτή δεν εγγράφεται στο νέο αρχείο, αλλιώς εγγράφεται (γραμμή 126). Με αυτόν τον τρόπο, βελτιώνονται σημαντικά οι επιδόσεις που καταγράφονται μέσω του trec_eval.exe. Η παραπάνω υπόδειξη έγινε στις συζητήσεις του eclass του μαθήματος, και η συζητούμενη βελτίωση επαληθεύτηκε από τον γράφοντα.

Έπειτα, στη main του συγκεκριμένου αρχείου, δημιουργείται το αρχείο στο οποίο θα αποθηκευτούν τα αποτελέσματα, αφού ανακτηθούν από το αρχείο στο οποίο είναι γραμμένα (γραμμή 151), και πάλι με τη χρήση του EnglishAnalyzer αυτά επεξεργάζονται κατάλληλα, προκειμένου οι λέξεις που περιέχουν να είναι στην ίδια μορφή με αυτές των documents (γραμμή 164). Έπειτα εκτελείται η μέθοδος search, της οποίας η λειτουργία έχει ήδη αναλυθεί. Η δημιουργία των τριών αρχείων, results20.txt, results30.txt και results50.txt,έγινε με την αλλαγή της τιμής της μεταβλητής hitsInPage (γραμμή 153), καθώς και με την αλλαγή του ονόματος του εκάστοτε αρχείου στις γραμμές 48 και 142.

Τέλος, είναι πολύ σημαντικό να τονιστεί ότι σε περίπτωση που μεταβληθεί το path των σχετικών αρχείων (εκτέλεση σε άλλον υπολογιστή κλπ), πρέπει να ενημερωθούν τα paths που υπάρχουν στις γραμμή 94 του αρχείου IndexCreator.java, και στις γραμμές 48, 142 και 151 του αρχείου Searcher.java, με τα αντίστοιχα paths στα οποία βρίσκονται τα αρχεία της συλλογής IR2020. Επιπλέον, πρέπει να ενημερωθούν τα paths που βρίσκονται στις γραμμές 115, 116 και 117 του αρχείου Searcher.java, με το αντίστοιχο path, στο οποίο βρίσκεται το

αρχείο wn_s.pl (αλλά αφού πρώτα μετονομαστεί, ώστε να έχει όνομα διάφορο του wn_s.pl), με την αρχική μορφή του (γραμμή 115), και το path στο οποίο θα δημιουργηθεί το νέο αρχείο wn_s.pl (με αυτό ακριβώς το όνομα), χωρίς τα ρήματα (γραμμές 116 – 117). Να σημειωθεί ότι αν χρησιμοποιηθεί το IDE Eclipse, ο φάκελος που πρέπει να περιέχει τα δύο αρχεία είναι ο /bin φάκελος μέσα στο project.

Γ. Περιγραφή τέταρτου βήματος (χρήση trec_eval):

Έπειτα από την εκτέλεση των τριών πρώτων βημάτων, μέσα από την χρήση της γραμμής εντολών, εκτελέστηκε η trec_eval, ακριβώς όπως και στην πρώτη φάση της εργασίας, και παρήχθησαν τα αρχεία που συμπεριλαμβάνονται στο zip αρχείο, μέσα στον φάκελο txts, σύμφωνα με την υπόδειξη που υπάρχει στην διαφάνεια 40 του αρχείου trec_eval.pdf. Παρακάτω υπάρχουν τα screenshots από τις διάφορες εκτελέσεις της trec_eval, για κάθε μετρική που ζητήθηκε:

i) Average Precision (AvgPre@k):

```
C:\Users\a\Desktop\txts>trec_eval -q -m map_cut.5 qrels.txt results20.txt > eval_map_cut_05.txt
C:\Users\a\Desktop\txts>trec_eval -q -m map_cut.10 qrels.txt results20.txt > eval_map_cut_10.txt
C:\Users\a\Desktop\txts>trec_eval -q -m map_cut.15 qrels.txt results20.txt > eval_map_cut_15.txt
C:\Users\a\Desktop\txts>trec_eval -q -m map_cut.20 qrels.txt results20.txt > eval_map_cut_20.txt
```

ii) Relative Returned Documents

```
C:\Users\a\Desktop\txts>trec_eval -q -M 5 -m num_rel_ret qrels.txt results20.txt > eval_num_rel_ret_05.txt
C:\Users\a\Desktop\txts>trec_eval -q -M 10 -m num_rel_ret qrels.txt results20.txt > eval_num_rel_ret_10.txt
C:\Users\a\Desktop\txts>trec_eval -q -M 15 -m num_rel_ret qrels.txt results20.txt > eval_num_rel_ret_15.txt
C:\Users\a\Desktop\txts>trec_eval -q -m num_rel_ret qrels.txt results20.txt > eval_num_rel_ret_20.txt
C:\Users\a\Desktop\txts>trec_eval -q -m num_rel_ret qrels.txt results30.txt > eval_num_rel_ret_30.txt
C:\Users\a\Desktop\txts>trec_eval -q -m num_rel_ret qrels.txt results50.txt > eval_num_rel_ret_50.txt
```

iii) Mean Average Precision (MAP@k):

```
C:\Users\a\Desktop\txts>trec_eval -q -m map qrels.txt results20.txt > eval_map_20.txt
C:\Users\a\Desktop\txts>trec_eval -q -m map qrels.txt results30.txt > eval_map_30.txt
C:\Users\a\Desktop\txts>trec_eval -q -m map qrels.txt results50.txt > eval_map_50.txt
```

Στις επόμενες σελίδα περιλαμβάνεται ο πίνακας με τα περιεχόμενα των παραχθέντων αρχείων, καταπώς υποδείχτηκε.

Query ID:	Total Returned Documents k:	Average Precision *AvgPre@k:*	Relative Returned Documents:	Mean Average Precision *MAP@k:*
Q01	k = 5	0.1698	4	
	k = 10	0.3857	8	
	k = 15	0.5307	11	
	k = 20	0.5749	12	0.5749
	k = 30		14	0.6516
	k = 50		15	0.6792
Q02	k = 5	0.0278	1	
	k = 10	0.0516	2	
	k = 15	0.0516	2	
	k = 20	0.0663	3	0.0663
	k = 30		3	0.0663
	k = 50		3	0.0663
Q03	k = 5	0.2536	4	
	k = 10	0.3743	6	
	k = 15	0.3743	6	
	k = 20	0.4323	8	0.4323
	k = 30		11	0.5078
	k = 50		14	0.5745
Q04	k = 5	0.0464	2	
	k = 10	0.0464	2	
	k = 15	0.0464	2	
	k = 20	0.0583	3	0.0583
	k = 30		3	0.0583
	k = 50		4	0.0653
Q05	k = 5	0.0896	3	
	k = 10	0.0896	3	
	k = 15	0.0896	3	
	k = 20	0.1207	5	0.1207
	k = 30		9	0.1911
	k = 50		13	0.2619
Q06	k = 5	0.0000	0	
	k = 10	0.0000	0	
	k = 15	0.0000	0	
	k = 20	0.0000	0	0.0000
	k = 30		0	0.0000
	k = 50		0	0.0000
Q07	k = 5	0.0625	1	
	k = 10	0.0764	2	
	k = 15	0.0920	3	
	k = 20	0.0920	3	0.0920
	k = 30		5	0.1114
	k = 50		10	0.1807
Q08	k = 5	0.3571	5	
	k = 10	0.5714	8	
	k = 15	0.6143	9	
	k = 20	0.6982	11	0.6982
	k = 30		11	0.6982
	k = 50		11	0.6982

Q09	k = 5	0.0952	2	
	k = 10	0.1156	3	
	k = 15	0.1664	6	
	k = 20	0.1872	7	0.1872
	k = 30		10	0.2342
	k = 50		12	0.2669
Q10	k = 5	0.0000	0	
	k = 10	0.0125	1	
	k = 15	0.0125	1	
	k = 20	0.0125	1	0.0125
	k = 30		1	0.0125
	k = 50		2	0.0181
All	k = 5	0.1102	22	
	k = 10	0.1724	35	
	k = 15	0.1978	43	
	k = 20	0.2242	53	0.2242
	k = 30		67	0.2531
	k = 50		84	0.2811

Δ. Σύγκριση αποτελεσμάτων δύο πρώτων φάσεων:

Αντιπαραβάλλοντας τα αποτελέσματα των δύο φάσεων, σε γενικές γραμμές είναι δυνατό να πει κανείς ότι δεν υπήρξε κάποια σημαντική βελτίωση των τιμών των μετρικών. Αντιθέτως, σχεδόν παντού παρατηρήθηκε στασιμότητα ή και μια μικρή υποχώρηση των τιμών.

Αναλυτικότερα, στα queries 1, 3, 5 και 8 οι τιμές ήταν οι ίδιες ακριβώς, στο 6^{ο} μηδενίστηκαν πλήρως, μόνο στο 9^{ο} αυξήθηκαν ελαφρώς, ενώ στα queries 2, 4, 7 και 10 σημειώθηκε μία μικρή μείωση των τιμών, κάτι που παρατηρήθηκε και στα συνολικά αποτελέσματα (ομάδα αποτελεσμάτων all). Αρχικά, τα αποτελέσματα ήταν πολύ χειρότερα, αλλά έπειτα από την εισαγωγή ορισμένων τροποποιήσεων που προτάθηκαν στο eclass, οι τιμές βελτιώθηκαν, κυρίως έπειτα από την αφαίρεση των ρημάτων από το .pl αρχείο.

Ίσως τα αποτελέσματα να βελτιωνόντουσαν και άλλο, αν αφαιρούταν και κάποια άλλη ομάδα λέξεων, αλλά αυτή η διαδικασία μείωσης των όρων του αρχείου δεν θα είχε νόημα από ένα σημείο και έπειτα. Και αυτό, γιατί θα καταλήγαμε να είχαμε σχεδόν ακριβώς τα ίδια δεδομένα που είχαμε και στην πρώτη φάση, δηλαδή έναν απλό EnglishAnalyzer, οπότε και τα αποτελέσματα θα πλησίαζαν περισσότερο σε αυτά της πρώτης φάσης.

Σύμφωνα με τα παραπάνω, καταλήγουμε στο συμπέρασμα, πως ίσως περαιτέρω κινήσεις με σκοπό την αύξηση των τιμών των μετρικών, δεν θα έπρεπε να ερευνηθούν στην κατεύθυνση της επιπλέον μείωσης του όγκου του αρχείου, αλλά στην εισαγωγή υπώνυμων όρων, αντί των συνώνυμων όρων, κατά την φάση του εμπλουτισμού των κειμένων των επερωτήσεων. Διότι αυτοί είναι που εννοιολογικά ταυτίζονται με τα συνώνυμα ενός όρου, και όχι τα συνώνυμα που προτείνει η συλλογή, τα οποία ορισμένες φορές έχουν τελείως διαφορετική σημασία από ό,τι η αρχική έννοια.