Περίοδος: Χειμερινό Εξάμηνο 2020 - 2021

Μάθημα: Ανάκτηση Πληροφοριών - Μηχανές Αναζήτησης

Συντάκτης: Μάρτσης Βασίλειος

ΑΜ: 154489

Θέμα:

Μηχανή Αναζήτησης Terrier (Εργασία 2)

# περιγραφη του περιβαλλοντος

## Η μηχανή αναζήτησης Terrier

Για την υλοποίηση της εργασίας γίνεται χρήση της μηχανής αναζήτησης Terrier (Version Number: 5.2). Το Terrier είναι μία πολύ ευέλικτη, αποτελεσματική και αποδοτική μηχανή αναζήτησης, Ανοιχτού κώδικα, που αναπτύσσεται γρήγορα για μεγάλες συλλογές εγγράφων. Υλοποιεί προηγμένες λειτουργίες ευρετηρίασης και ανάκτησης δεδομένων, και αποτελεί ιδανική πλατφόρμα για γρήγορη ανάπτυξη και αξιολόγηση αιτημάτων ανάκτησης μεγάλης κλίμακας. Είναι λογισμικό ανοιχτού κώδικα (ΕΛ/ΛΑΚ) και είναι μια περιεκτική και ευέλικτη πλατφόρμα για έρευνα και πειραματισμό στην ανάκτηση κειμένου. Η έρευνα διεξάγεται εύκολα σε standard TREC (Text Retrieval Conference) και CLEF συλλογές δοκιμής. Είναι υλοποιημένο σε Java, και έχει αναπτυχθεί στο School of Computing Science, University of Glasgow.

# προετοιμασία για τη δημιουργία των ευρετηρίων (index)

Μετά την εγκατάσταση της μηχανής στο σύστημα, μέσο του command line των windows, που είναι και το λειτουργικό σύστημα που χρησιμοποιείται για την υλοποίηση της εργασίας, γίνεται χρήση της εντολής “trec\_setup.bat” η οποία παίρνει ως όρισμα τη διαδρομή του φακέλου προορισμού των αρχείων από τα οποία θέλουμε να δημιουργήσουμε το ευρετήριο. Η εντολή αυτή θα δημιουργήσει ορισμένα αρχεία απαραίτητα για προχωρήσουμε στην κατασκευή του ευρετηρίου. Τα αρχεία που θα χρησιμοποιηθούν για την υλοποίηση της εργασίας αυτής είναι μία συλλογή 53.006 εγγράφων (documents), που ακολουθούν τη μορφή TREC (trec format). Για τον σκοπό αυτής της εργασίας θα δημιουργηθούν βασισμένα στη συλλογή εγγράφων αυτή, πέντε ξεχωριστά ευρετήρια.

# δημιουργία ευρετηρίων

Για τη δημιουργία των 5 ευρετηρίων, πραγματοποιήθηκαν για το κάθε ένα από αυτά, διαφορετικές παραμετροποιήσεις στα αρχεία του Terrier. Αρχικά θα χρησιμοποιηθεί η εντολή “terrier bi’’ (Batch Indexing) η οποία θα κατασκευάσει ένα ευρετήριο βασισμένο στα πρώτα δέκα χιλιάδες αρχεία από τη συλλογή που προαναφέρθηκε.

Ακολουθούν η περιγραφή των 5 ευρετηρίων, οι παραμετροποιήσεις που πραγματοποιήθηκαν πάνω σε αυτά και τα αποτελέσματά τους:

## Δημιουργία του πρώτου ευρετηρίου

Για τη δημιουργία του πρώτου ευρετηρίου χρησιμοποιήθηκαν οι προκαθορισμένες ρυθμίσεις του Terrier οι ρυθμίσεις αυτές είναι οι εξής:

### Αρχείο terrier.properites

#### TrecDocTags.doctag=DOC

Η ετικέτα που είναι ορισμένη ως η κύρια ετικέτα των αρχείων.

#### TrecDocTags.idtag=DOCNO

Η ετικέτα που περιέχει το νούμερο το οποίο αντιστοιχεί στο κάθε αρχείο.

#### TrecDocTags.skip=DOCHDR

Οι ετικέτες που θα παραλειφθούν κατά τη διαδικασία της ευρετηρίασης.

#### TrecDocTags.casesensitive=false

false σημαίνει ότι δεν παίζει ρόλο το αν στις ετικέτες χρησιμοποιούνται κεφαλαία ή πεζά γράμματα.

#### TrecQueryTags.doctag=TOP

Το όνομα της ετικέτας που ορίζει την αρχή του TREC αρχείου.

#### TrecQueryTags.idtag=NUM

Το όνομα της ετικέτας που σημαδεύει το id του TREC αρχείο.

#### TrecQueryTags.process=TOP,NUM,TITLE

Οι ετικέτες μέσα στο TREC αρχείο που θα υποστούν επεξεργασία.

#### TrecQueryTags.skip=DESC,NARR

Οι ετικέτες μέσα στο TREC αρχείο που θα παραληφθούνε κατά τη διάρκεια των ερωτημάτων.

Χρησιμοποιώντας τις παραπάνω default ρυθμίσεις, το Terrier θα επεξεργαστεί τους τίτλους των TREC αρχείων, καθώς θα παραλείψει τα descriptions και τα narratives.

#### termpipelines=Stopwords,PorterStemmer

Χρησιμοποιώντας τη ρύθμιση αυτή το σύστημα αγνοεί τις λέξεις των ερωτημάτων που περιέχονται στο αρχείο Stopwords και μετά την ανάγει σε μία ποιο βασική της μορφή χρησιμοποιώντας τον προκαθορισμένο Stemmer, ο οποίος ακολουθεί τον αλγόριθμο αναγωγής λέξεων της αγγλικής γλώσσας του Porter (PorterStemmer).

### Χρόνος εκτέλεσης της εντολής «terrier bi»

Ο χρόνος που χρειάστηκε για να δημιουργηθεί το ευρετήριο στο σύστημα που πραγματοποιείται η εργασία 359 δευτερόλεπτα.

Για τη δημιουργία των υπόλοιπων ευρετηρίων θα αναφερθούν μόνο οι ρυθμίσεις που τροποποιήθηκαν έτσι ώστε να διαφέρουν από τις προκαθορισμένες.

## Δημιουργία του δεύτερου ευρετηρίου

### Αρχείο terrier.properites

#### TrecDocTags.skip=DOCHDR,CLASSIFICATIONS

Επειδή η ετικέτα CLASSIFICATIONS περιέχει μόνο αριθμούς των διαφόρων πατεντών, πιθανών η παράληψή της να ελαφρύνει το σύστημα, επηρεάζοντάς το ελάχιστα.

#### terrier.index.path=C:/Terrier/terrier-project-5.2/var/index2

Η διαδρομή του φακέλου στον οποίο θα αποθηκευτεί το ευρετήριο. Ο λόγος χρήσης αυτής της παραμέτρου είναι η δημιουργία 5 διαφορετικών ευρετηρίων, έτσι ώστε να μην χρειαστεί να διαγραφεί το προηγούμενο προκειμένου να δημιουργηθεί το καινούριο. Για να χρησιμοποιηθεί αυτή η μεταβλητή, θα πρέπει να τεθεί ως όρισμα στην εντολή “terrier bi” (π.χ. terrier bi terrier.index.path).

### Χρόνος εκτέλεσης της εντολής «terrier bi»

Ο χρόνος που χρειάστηκε για να δημιουργηθεί το ευρετήριο στο σύστημα που πραγματοποιείται η εργασία 81 δευτερόλεπτα.

## Δημιουργία του τρίτου ευρετηρίου

### Αρχείο terrier.properites

#### TrecDocTags.skip=DOCHDR,CLASSIFICATIONS

#### stopwords.filename=stopword-list2.txt

Η διαδρομή του φακέλου στον οποίο βρίσκεται το αρχείο με τα stopwords (Λέξεις που θα παραληφθούν κατά τη διάρκεια των ερωτημάτων). Η διαδρομή αυτή που εισήχθη είναι ένα προσαρμοσμένο το οποίο περιέχει ως stopword μόνο τη λέξη «the».

#### termpipelines=WeakPorterStemmer

#### terrier.index.path=C:/Terrier/terrier-project-5.2/var/index3

Θα χρησιμοποιηθεί μία ποιο αδύναμη έκδοση του PorterStemmer

### Χρόνος εκτέλεσης της εντολής «terrier bi»

Ο χρόνος που χρειάστηκε για να δημιουργηθεί το ευρετήριο στο σύστημα που πραγματοποιείται η εργασία 359 δευτερόλεπτα.

## Δημιουργία του τέταρτου ευρετηρίου

### Αρχείο terrier.properites

#### TrecDocTags.skip=DOCHDR,CLASSIFICATIONS

#### Η μεταβλητή «TrecQueryTags.skip» αφαιρέθηκε ώστε να μην παραληφθεί καμία ετικέτα

#### Η μεταβλητή «TrecQueryTags.process» αφερέθηκε ώστε να επεξεργαστούν όλες οι ετικέτες

#### Η μεταβλητή «TrecQueryTags.skip» αφαιρέθηκε ώστε να μην παραληφθεί καμία ετικέτα

#### Η μεταβλητή «termpipelines» αφαιρέθηκε ώστε να μην χρησημοποιηθούν ούτε stopwords ούτε stemmer.

#### terrier.index.path=C:/Terrier/terrier-project-5.2/var/index4

### Χρόνος εκτέλεσης της εντολής «terrier bi»

Ο χρόνος που χρειάστηκε για να δημιουργηθεί το ευρετήριο στο σύστημα που πραγματοποιείται η εργασία 359 δευτερόλεπτα.

## Δημιουργία του πέμπτου ευρετηρίου

### Αρχείο terrier.properites

#### FieldTags.process=TITLE

Οι ετικέτες που θα συμπεριληφθούν στην ευρετηρίαση.

### Χρόνος εκτέλεσης της εντολής «terrier bi»

Ο χρόνος που χρειάστηκε για να δημιουργηθεί το ευρετήριο στο σύστημα που πραγματοποιείται η εργασία 81 δευτερόλεπτα.

# εκτελεση ερωτημάτων

Για την εκτέλεση των ερωτημάτων, θα χρησιμοποιηθεί το πρώτο ευρετήριο που δημιουργήθηκε.

Η ανάκτηση των εγγράφων της συλλογής και η παραγωγή των αποτελεσμάτων έγινε με βάση ένα αρχείο 30 ερωτημάτων (topics list). Το αρχείο των ερωτημάτων δημιουργήθηκε με βάση το αρχείο (MinicollectionTopics.txt) και ακολουθεί τη μορφοποίηση TREC\_format(sgml). Το περιεχόμενο των ερωτημάτων (Topics Content) προέκυψε από τα αντίστοιχα αρχεία .xml που βρίσκονται στο φάκελο PAC\_topics.

Για να παραχθεί το αποτέλεσμα των ερωτημάτων, θα γίνει χρήση της εντολής “terrier br” (Batch Retrieval).

# Αξιολόγηση (Evaluation)

Για την αξιολόγηση των αποτελεσμάτων των ερωτημάτων που εκτελέστηκαν στο προηγούμενο βήμα, θα γίνει χρήση του προγράμματος trec\_eval-9.0.7. Για τον έλεγχο της εγγύτητας των αποτελεσμάτων είναι αναγκαία η χρήση νεώς αρχείου που περιλαμβάνει τα μη αμφισβητήσιμα αποτελέσματα, δηλαδή ποια είναι τα πραγματικά σχετικά έγγραφα σε σχέση με τα ερωτήματα.

Η εντολή που θα πυροδοτήσει την αξιολόγηση των αποτελεσμάτων είναι η εξής: «trec\_eval.exe qrels.txt results.res», όπου qrels.txt το αρχείο των ορθών αποτελεσμάτων και results.res το αρχείο των αποτελεσμάτων μετά την εκτέλεση των ερωτημάτων.

Το trec\_eval πρόκειται για ένα πρόγραμμα που αξιολογεί τα αποτελέσματα των ερωτημάτων χρησιμοποιώντας κάποιες μετρικές. Οι μετρικές αυτές αναφέρονται αναλυτικά παρακάτω:

* num\_ret: Συνολικός αριθμός των εγγράφων που ανακτήθηκαν από όλα τα ερωτήματα
* num\_rel: Συνολικός αριθμός σχετικών εγγράφων σε όλα τα ερωτήματα
* num\_rel\_ret: Συνολικός αριθμός σχετικών εγγράφων που ανακτήθηκαν σε όλα τα ερωτήματα
* map: χάρτης μέσης ακρίβειας (MAP)
* gm\_ap: Μέση ακρίβεια. Γεωμετρική μέση τιμή, q\_score=log(MAX(map,.00001))
* R-prec: R-Precision (Ακρίβεια μετά από R (= num-rel for topic) έγγραφα που ανακτήθηκαν)
* bpref: Binary Preference, top R judged nonrel
* recip\_rank: Αμοιβαία κατάταξη των κορυφαίων σχετικών εγγράφων
* ircl\_prn.0.00: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.00 ανάκληση
* ircl\_prn.0.10: Interpolated Recall - Μέσοι όροι ακριβείας στη 0.10 ανάκληση
* ircl\_prn.0.20: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.20 ανάκληση
* ircl\_prn.0.30: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.30 ανάκληση
* ircl\_prn.0.40: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.40 ανάκληση
* ircl\_prn.0.50: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.50 ανάκληση
* ircl\_prn.0.60: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.60 ανάκληση
* ircl\_prn.0.70: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.70 ανάκληση
* ircl\_prn.0.80: Interpolated Recall – Μέσοι όροι ακριβείας στη 0.80 ανάκληση
* ircl\_prn.0.90: Recall – Μέσοι όροι ακριβείας στη 0.90 ανάκληση
* ircl\_prn.1.00: Interpolated Recall – Μέσοι όροι ακριβείας στη 1.00 ανάκληση
* P5: Ακρίβεια μετά από 5 έγγραφα που ανακτήθηκαν
* P10: Ακρίβεια μετά από 10 έγγραφα που ανακτήθηκαν
* P15: Ακρίβεια μετά από 15 έγγραφα που ανακτήθηκαν
* P20: Ακρίβεια μετά από 20 έγγραφα που ανακτήθηκαν
* P30: Ακρίβεια μετά από 30 έγγραφα που ανακτήθηκαν
* P100: Ακρίβεια μετά από 100 έγγραφα που ανακτήθηκαν
* P200: Ακρίβεια μετά από 200 έγγραφα που ανακτήθηκαν
* P500: Ακρίβεια μετά από 500 έγγραφα που ανακτήθηκαν
* P1000: Ακρίβεια μετά από 1000 έγγραφα που ανακτήθηκαν

Για το σκοπό αυτής της εργασίας θα χρησιμοποιηθεί το ίδιο αρχείο ερωτημάτων που δημιουργήθηκε παραπάνω, για τρεις διαφορετικές εκτελέσεις.

## Πρώτη εκτέλεση

Για την πρώτη εκτέλεση των ερωτημάτων χρησιμοποιήθηκε το μοντέλο BM25 (Best Match 25), το οποίο είναι ένα πιθανοτικό το οποίο είναι ένα πιθανοτικό μοντέλο, το οποίο πολλαπλασιάζει δύο μετρικές: πόσες φορές εμφανίζεται μία λέξη μέσα σε ένα έγγραφο, και την αντίστροφη συχνότητα εγγράφου της λέξης σε ένα σύνολο εγγράφων.

Μετά την εκτέλεση της εντολής αξιολόγησης, παραχθήκαν τα παρακάτω αποτελέσματα τα οποία μπορούν να ερμηνευτούν χρησιμοποιώντας την ανάλυση που πραγματοποιήθηκε σε προηγούμενο σημείο του κειμένου:

* runid all BM25
* num\_q all 1
* num\_ret all 1000
* num\_rel all 26
* num\_rel\_ret all 19
* map all 0.1251
* gm\_map all 0.1251
* Rprec all 0.1923
* bpref all 0.7308
* recip\_rank all 1.0000
* iprec\_at\_recall\_0.00 all 1.0000
* iprec\_at\_recall\_0.10 all 0.2500
* iprec\_at\_recall\_0.20 all 0.1842
* iprec\_at\_recall\_0.30 all 0.1667
* iprec\_at\_recall\_0.40 all 0.1279
* iprec\_at\_recall\_0.50 all 0.1197
* iprec\_at\_recall\_0.60 all 0.0323
* iprec\_at\_recall\_0.70 all 0.0231
* iprec\_at\_recall\_0.80 all 0.0000
* iprec\_at\_recall\_0.90 all 0.0000
* iprec\_at\_recall\_1.00 all 0.0000
* P\_5 all 0.2000
* P\_10 all 0.1000
* P\_15 all 0.1333
* P\_20 all 0.2500
* P\_30 all 0.1667
* P\_100 all 0.1100
* P\_200 all 0.0700
* P\_500 all 0.0320
* P\_1000 all 0.0190

## Δεύτερη εκτέλεση

Για την πρώτη εκτέλεση των ερωτημάτων χρησιμοποιήθηκε το μοντέλο TF\_IDF (term frequency–inverse document frequency), To οποίο είναι ένα διανυσματικό μοντέλο. Το συγκεκριμένο μοντέλο χαρακτηρίζεται από μία εμφανή βελτίωση στον τομέα των βαρών σε σχέση µε το Boolean αφού δεν υποστηρίζει δυαδικά βάρη για την σύγκριση ερωτημάτων και κειμένων. Έτσι έχουμε ένα ουσιαστικό μέτρο ομοιότητας μεταξύ κειμένων και ερωτημάτων που ονομάζεται βαθιός ομοιότητας. Μετά τον υπολογισμό αυτού του μέτρου τα κείμενα διατάσσονται µε φθίνουσα σειρά έχοντας ως κριτήριο το βαθιό ομοιότητας µε το ερώτημα του χρήστη.

Μετά την εκτέλεση της εντολής αξιολόγησης, παραχθήκαν τα παρακάτω αποτελέσματα τα οποία μπορούν να ερμηνευτούν χρησιμοποιώντας την ανάλυση που πραγματοποιήθηκε σε προηγούμενο σημείο του κειμένου:

* runid all TF\_IDF
* num\_q all 1
* num\_ret all 1000
* num\_rel all 26
* num\_rel\_ret all 19
* map all 0.1261
* gm\_map all 0.1261
* Rprec all 0.1923
* bpref all 0.7308
* recip\_rank all 1.0000
* iprec\_at\_recall\_0.00 all 1.0000
* iprec\_at\_recall\_0.10 all 0.2632
* iprec\_at\_recall\_0.20 all 0.1892
* iprec\_at\_recall\_0.30 all 0.1818
* iprec\_at\_recall\_0.40 all 0.1224
* iprec\_at\_recall\_0.50 all 0.1040
* iprec\_at\_recall\_0.60 all 0.0319
* iprec\_at\_recall\_0.70 all 0.0198
* iprec\_at\_recall\_0.80 all 0.0000
* iprec\_at\_recall\_0.90 all 0.0000
* iprec\_at\_recall\_1.00 all 0.0000
* P\_5 all 0.2000
* P\_10 all 0.1000
* P\_15 all 0.1333
* P\_20 all 0.2500
* P\_30 all 0.1667
* P\_100 all 0.1200
* P\_200 all 0.0700
* P\_500 all 0.0300
* P\_1000 all 0.0190

## Τρίτη εκτέλεση:

Για την πρώτη εκτέλεση των ερωτημάτων χρησιμοποιήθηκε το μοντέλο DFR\_BM25, το οποίο είναι μία υλοποίηση Divergence From Randomness (DFR) του BM25 μοντέλου. Το Divergence From Randomness (DFR) είναι μια γενίκευση ενός από τα πρώτα μοντέλα ανάκτησης πληροφοριών. Τα μοντέλα DFR βασίζονται σε αυτήν την απλή ιδέα: "Όσο μεγαλύτερη είναι η απόκλιση της συχνότητας όρου εντός του εγγράφου από τη συχνότητά της στη συλλογή, τόσο περισσότερες πληροφορίες μεταφέρονται από τη λέξη t στο έγγραφο d". Με άλλα λόγια, ο όρος-βάρος σχετίζεται αντιστρόφως με την πιθανότητα της συχνότητας όρου μέσα στο έγγραφο d που λαμβάνεται από ένα μοντέλο Μ τυχαιότητας

Μετά την εκτέλεση της εντολής αξιολόγησης, παραχθήκαν τα παρακάτω αποτελέσματα τα οποία μπορούν να ερμηνευτούν χρησιμοποιώντας την ανάλυση που πραγματοποιήθηκε σε προηγούμενο σημείο του κειμένου:

* runid all DFR\_BM25
* num\_q all 1
* num\_ret all 1000
* num\_rel all 26
* num\_rel\_ret all 19
* map all 0.1226
* gm\_map all 0.1226
* Rprec all 0.1923
* bpref all 0.7308
* recip\_rank all 1.0000
* iprec\_at\_recall\_0.00 all 1.0000
* iprec\_at\_recall\_0.10 all 0.2222
* iprec\_at\_recall\_0.20 all 0.1842
* iprec\_at\_recall\_0.30 all 0.1636
* iprec\_at\_recall\_0.40 all 0.1264
* iprec\_at\_recall\_0.50 all 0.1120
* iprec\_at\_recall\_0.60 all 0.0315
* iprec\_at\_recall\_0.70 all 0.0210
* iprec\_at\_recall\_0.80 all 0.0000
* iprec\_at\_recall\_0.90 all 0.0000
* iprec\_at\_recall\_1.00 all 0.0000
* P\_5 all 0.2000
* P\_10 all 0.1000
* P\_15 all 0.1333
* P\_20 all 0.2000
* P\_30 all 0.1667
* P\_100 all 0.1100
* P\_200 all 0.0700
* P\_500 all 0.0300
* P\_1000 all 0.0190

# σύγκριση αποτελεσμάτων και συμπεράσματα

Παρατηρούμε ότι οι τιμές που διαφέρουν στις τρεις εκτελέσεις είναι:

* map
* gm\_map
* iprec\_at\_recall\_0.00
* iprec\_at\_recall\_0.10
* iprec\_at\_recall\_0.20
* iprec\_at\_recall\_0.30
* iprec\_at\_recall\_0.40
* iprec\_at\_recall\_0.50
* iprec\_at\_recall\_0.60
* iprec\_at\_recall\_0.70
* P\_20
* P\_100
* P\_500

Δηλαδή παρατηρούμε ότι παρόλο που κατά τη διάρκεια της αξιολόγησης υπήρξαν κάποιες διακυμάνσεις στις τιμές αλλά όσο πλησιάζουμε στις τελευταίες ανακλάσεις και ο αριθμός των εγγράφων που ανακτήθηκαν μεγαλώνει, οι τιμές είναι ίδιες και για τρία μοντέλα ανάκτησης βαρών.

# επίλογος

Το κείμενο αυτό ήταν μία περιγραφή της εκτέλεσης, στην μηχανή αναζήτησης Terrier, διαφόρων ερωτημάτων και της αξιολόγησης των αποτελεσμάτων τους χρησιμοποιώντας το εργαλείο trec\_eval. Κατά τη διαδικασία αυτή υποβλήθηκαν τρία διαφορετικά αποτελέσματα (runs), που προέκυψαν από 3 διαφορετικές εκτελέσεις. Σε κάθε εκτέλεση χρησιμοποιήθηκε διαφορετικό μοντέλο ανάκτησης πληροφορίας, και έπειτα αξιολογήθηκε η ορθότητα των αποτελεσμάτων με τη χρήση του trec eval. Παρατηρήθηκε ότι το αρχείο των αποτελεσμάτων της κάθε εκτέλεσης διέφερε σε μεγάλο βαθμό από τις υπόλοιπες, καθώς επίσης και οι μετρικές της αξιολόγησής τους.

1. <https://github.com/terrier-org/terrier-core/>
2. <http://www.cs.cornell.edu/courses/cs4300/2013fa/lectures/retrieval-models-2-4pp.pdf>
3. <https://www.researchgate.net/publication/45886647_Integrating_the_Probabilistic_Models_BM25BM25F_into_Lucene>
4. <https://ieeexplore.ieee.org/document/7546200>
5. <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>