Πανεπιστήμιο Κρήτης

Τμήμα Επιστήμης Υπολογιστών

HY463 Συστήματα Ανάκτησης Πληροφοριών

Εξάμηνο: Άνοιξη 2023

**Γραπτή Αναφορά Έργου**

Στοιχεία Φοιτητών

|  |  |
| --- | --- |
| Μέλος | 1ο |
| Ονοματωπώνυμο | Ισίδωρος Χατζηχρυσός |
| ΑΜ | 4338 |
| Email | csd4338@csd.uoc.gr |
|  |  |
| Μέλος | 2ο |
| Ονοματωπώνυμο | Καραγιαννάκη Αντιγόνη |
| ΑΜ | 4642 |
| Email | csd4642@csd.uoc.gr |

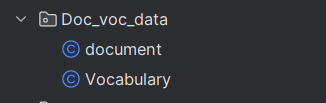
**ΦΑΣΗ Α:**

**Η ΥΛΟΠΟΙΗΣΗ ΜΑΣ ΒΑΣΙΣΤΗΚΕ ΣΤΟ MiniCollection !**

1. **(B1)**

**Υλοποιήθηκε στην:** Doc\_voc\_data/**document.java**

Η συνάρτηση **compute\_occurrences**

****

Η δομή παρακάτω δομή αποθηκεύει όλες τις απαραίτητες πληροφορίες για τα term frequencies όπως αυτά υπολογίζονται για το κάθε ταγκ του nxml αρχείου.

**HashMap<String, Map<Integer,Integer>> termFrequencies;**

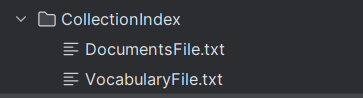
**// mapping: < word , tag\_id, tf>**

Το πρόγραμμα διαβάζει τα περιεχόμενα των ετικετών ενός XML αρχείου UTF-8 κωδικοποίηση (ώστε να υπάρχει υποστήριξη πολυγλωσσικότητας) και τυπώνει το πλήθος των διαφορετικών λέξεων, και την κάθε διαφορετική λέξη συνοδευόμενη από τις ετικέτες (tags) στις οποίες εντοπίστηκε και το πλήθος εμφανίσεών της στην κάθε ετικέτα.

1. **(B2 , B4, B5 )**

**Υλοποιήθηκε στην:** src/main/java/xmlReader**/folderReader.java**

**Διαβάζει φακέλους του Mini Collection και παράγει τα παρακάτω:**

****

Που κρατάνε τις πληροφορίες για τα documents & το Vocabulary , δηλ .

Το DocumentsFile.txt περιέχει πληροφορίες σχετικά με τα έγγραφα που χρησιμοποιούνται στο πρόγραμμα. Κάθε γραμμή στο αρχείο αυτό αντιστοιχεί σε ένα έγγραφο. Η γραμμή περιέχει το id του εγγράφου, την απόλυτη διαδρομή του εγγράφου και ένα δείκτη που δείχνει στο αρχείο Posting.

Το VocabularyFile.txt περιέχει πληροφορίες σχετικά με το λεξιλόγιο που χρησιμοποιείται στο πρόγραμμα. Κάθε γραμμή στο αρχείο αυτό αντιστοιχεί σε μια λέξη από το λεξιλόγιο. Η λέξη ακολουθείται από τη συχνότητα εμφάνισης της λέξης (df) και ένα δείκτη που δείχνει στο αρχείο Posting.

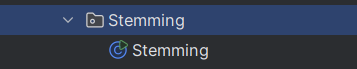
Η υλοποίηση για την μινι συλλογή καταφέρνει :

Για συλλογή 🡪 Vocabulary Size: 18381

Με αρχεία 🡪 Docs read 54

Χρόνο 🡪 Execution time in milliseconds: 79351

1. **(B3)**



Το πακέτο Stemming είναι υπεύθυνο για την διαδικασία που περιγράφεται στο Β3. Με την βοήθεια των ευρεθέντων λέξεων από Β1, Β2 .

O Stemmer παράγει ένα αρχείο στο Stemming package με τις παραχθείσες λέξεις.

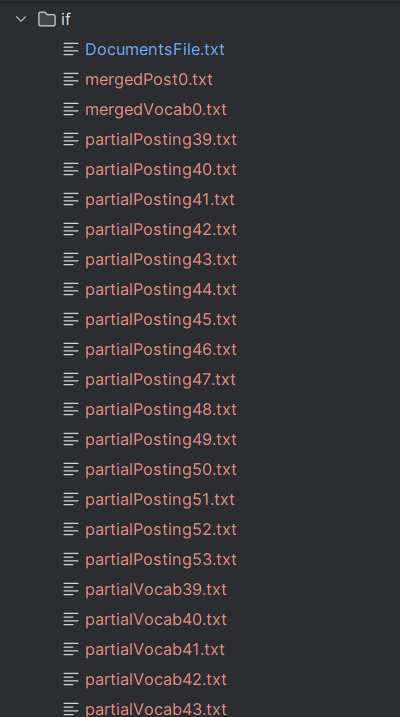
1. **ΑΝΑΦΕΡΕΤΑΙ ΑΠΟ ΠΑΝΩ**
2. **ΑΝΑΦΕΡΕΤΑΙ ΑΠΟ ΠΑΝΩ**
3. **(B6)** src/main/java/pIndexing/**pindexing**.java

Η δομική επιλογή που ακολουθήσαμε ήταν η 2η , οπότε και προβαίνουμε στην δημιουργία ενός Inverted Index File . Η χρήση του Partial Indexing ήταν πολύ βοηθητική , και με αυτήν καταφέραμε χρόνο ευρετηρίασης :

Execution time in milliseconds: ~ 20 seconds

Επεκτείνοντας το σύστημά μας χρειάστηκε να ξαναδημιουργήσουμε πολλές νέες δομές hash map που χρησιμοποιούνται για να συνδυάσουν τα terms με τις μετρικές.

**Παραγόμενα:**



**Εδώ Παράγονται και ευρετηριάζονται αρχεία επίι του Vocabulary & του Posting.**

**Extras:**

* Υλοποιήσαμε **Partial Indexing**
* Ξεκινήσαμετο **GUI** στοαντίστοιχο **package**

Στην παρούσα υλοποίηση χρησιμοποιήθηκαν :

* **Software : Intellij IDEA**
* **SDK version : openjdk 21.0.2**
* & Δόθηκε 4GB RAM