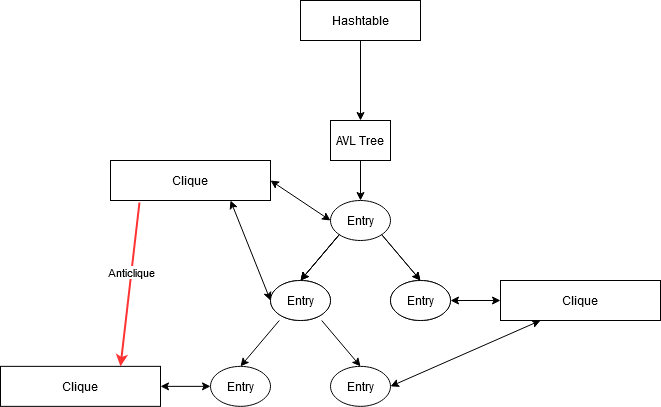
Μέλη της Ομάδας:  
ΑΜ 1115201700119 Θωμάς Παπαθυμιόπουλος

ΑΜ 1115201700106 Νικόλαος Νιώτης



**Σύνοψη**

//ΑΝΑΓΝΩΣΗ - ΚΛΙΚΕΣ - ΑΝΤΙΚΛΙΚΕΣ

Με την λήξη της ανάγνωσης έχουμε το πλήρες λεξιλόγιο σε μορφή AVL δέντρου του οποίου κάθε κόμβος αποτελεί μια λέξη. Το δέντρο αυτό μετατρέπεται σε πίνακα. Κατά την μετατροπή αυτή σε κάθε λέξη του δέντρου φυλάσσεται η θέση που της αντιστοιχεί στον πίνακα για να έχουμε ταχύτερες αναζητήσεις ( η αναζήτηση σε πίνακα έχει time complexity O(n) ενώ σε AVL δέντρο O(nlogn)).  
  
Έπειτα δημιουργούμε ένα sparse matrix για όλα τα entries του προγράμματος. Το sparse matrix είναι ένας πίνακας από λίστες λέξεων. Κάθε θέση του πίνακα αντιστοιχεί σε κάποιο entry του dataset μας και η λίστα περιέχει τις μοναδικές λέξεις των specs του μαζί με τις τιμές bow και tfidf που της αντιστοιχούν. Οι τιμές αυτές υπολογίζονται κατά την διάρκεια της δημιουργίας του sparse matrix με την βοήθεια του πίνακα idf που έχουμε γεμίσει για κάθε λέξη κατά την ανάγνωση.   
  
Κατά δημιουργία του sparse matrix φυλάμε σε έναν πίνακα το άθροισμα των τιμών tfidf για κάθε λέξη σε έναν πίνακα. Αυτός ο πίνακας ταξινομείται παράλληλα με τον πίνακα των λέξεων και έτσι γνωρίζουμε τις καλύτερες λέξεις του προγράμματος. Εισάγουμε τις 1000 καλύτερες λέξεις σε ένα νέο λεξιλόγιο AVL δέντρο και αφαιρούμε από το sparse matrix τις λέξεις που δεν ανήκουν στο νέο μας λεξιλόγιο.  
  
Πλέων από το sparse matrix μπορούμε εύκολα για κάποιο αρχείο να πάρουμε το tfidf vector των λέξεων του και να το χρησιμοποιήσουμε στη λογιστική παλινδρόμηση.

O Job Scheduler δημιουργεί το πλήθος των thread που έχουμε ορίσει και τα φυλάει σε ένα thread pool σε μορφή πίνακα και αρχικοποιεί την ουρά προτεραιότητας που θα περιέχει τα jobs. Jobs προστίθενται στην ουρά με την χρήση της συνάρτησης **submit\_job().**Υπάρχει μια γενική κλαση Job η οποία κληρονομείται από και εξειδικεύεται σε lr\_train\_job, lr\_retrain\_job και lr\_test\_job.

Με την συνάρτηση **execute\_all\_jobs()** ο Job Scheduler αναθέτει σε όλα τα threads του thread pool την συνάρτηση thread\_f τα οποία αμέσως την εκτελούν. Κάθε thread εξάγει το πρώτο Job από την ουρά προτεραιότητας και εκτελεί την συνάρτηση του. Τα threads επαναλαμβάνουν αυτή την διαδικασία μέχρι να αδειάσει η ουρά.  
  
Με την συνάρτηση **wait\_for\_all\_tasks\_to\_finish()** ο Job Scheduler περιμένει όλα τα threads να τερματίσουν την εκτέλεση της thread\_f.

//LR

**Πειράματα**

