Εργασία στο μάθημα Ανάκτηση Πληροφορίας

Λάλας Σταύρος: 2964
Ιακωβίδης Βασίλειος: 2965

Ολόκληρη η υλοποίηση της εργασίας βρίσκεται σε ένα αρχείο python, το myCrawler.py. Η εκτέλεση γίνεται με την εντολή myCrawler.py https://mypage.gr 200 1 8, όπου το 200 αντιπροσωπεύει τον αριθμό των sites που θα ψάξει ο crawler, το 1 προοριζόταν για την διατήρηση των παλαιότερων δεδομένων ή την εκίννηση από την αρχή, ωστόσο δεν χρησιμοποιήθηκε στον κώδικα, και το 8 δηλώνει το πλήθος των threads που ασχολούνται με την συλλογή των σελίδων και στην συνέχεια στην δημιουργία του καταλόγου (Indexer). Ολόκληρη η εκτέλεση της εργασίας γίνεται στο τερματικό, η εκίννηση του crawler, η δημιουργία του καταλόγου, η ανάθεση του ερωτήματος και η επιστροφή των αποτελεσμάτων.

Ο κώδικας αρχικά διαβάζει τις παραμέτρους εκτέλεσης και δημιουργεί τρία dict και μία λίστα με την βοήθεια manager για να μην υπάρχει θέμα στην ενημέρωση τιμών με την χρήση των πολλών threads. Τα dict είναι: ο Indexer, με τον ρόλο του αντεστραμμένου καταλόγου καθώς έχει το λεξικό και τις λίστες εμφάνισεις τις κάθε λέξεις σε μορφή (λέξη:[έγγραφα που εμφανίζεται, συνολικές φορές εμφάνισης, άρθρο1, εμφανίσεις στο άρθρο1, άρθρο2, εμφανίσεις στο άρθρο2, κ.ο.κ.], ο sites_to_crawl που αποθηκεύει τα sites τα οποία στην συνέχεια θα διαβάσουμε και θα χρησιμοποιήσουμε για την δημιουργία του Indexer, μια λίστα max_nx η οποία αποθηκεύει την μέγιστη τιμή για τις εμφανίσεις ενός όρου στο λεξικό (χρησιμοποιείται για τον υπολογισμό του idf των όρων) καθώς και το articlesDict το οποίο αποθηκεύει τα link μαζί με τις συνολικές λέξεις που διαθέτει το άρθρο και το μέγιστο πλήθος εμφανίσεων ενός όρου στο άρθρο(για τον υπολογισμό του update score). Υπάρχει επίσης μια λίστα η οποία περιέχει ένα πλήθος stopwords (στα ελληνικά) και χρησιμοποιείται για βελτίωση της αποτελεσματικότητας καθώς και μια μεταβλητή για την μέγιστη τιμή εμφάνισης λέξης στο ερώτημα.

Αρχικά όσο δεν έχει συμπληρωθεί το πλήθος των sites που θέλουμε να ψάξουμε καλούμε την linkFinder η οποία παίρνει ένα url και βρίσκει όλα τα link που υπάρχουν μέσα σε αυτό, προσθέτωντας τα σε αυτά που πρέπει να ψάξουμε αργότερα. Έπειτα, καλείται η article και δέχεται ως όρισμα τα sites αυτά, όπου για κάθε site προσπαθεί να ξεχωρίσει το άρθρο. Μετά καλεί την add_article_to_dict στέλνοντας το site μαζί με το άρθρο που έχει ξεχωρίσει. Εκεί για κάθε λέξη που υπάρχει μέσα στο άρθρο, κοιτάμε να αφαιρέσουμε κόμματα και τελείες αν ακολουθούν μια λέξη, κάνουμε όλες τις λέξεις στα μικρά γράμματα και αφαιρούμε τους τόνους για καλύτερα αποτελέσματα. Ελέγχουμε αν η λέξη δεν είναι στα stopwords και αν όχι, ψάχνουμε αν υπάρχει ήδη στον Indexer, αν ναι αυξάνουμε τις συνολικές φορές εμφάνισης και ελέγχουμε αν υπάρχει και στο συγκεκριμένο url. Αν ναι, αυξάνουμε τον αριθμό εμφάνισης της λέξης στο συγκεκριμένο άρθρο, αλλιώς προσθέτουμε το άρθρο στην λίστα εμφάνισης. Αν η λέξη δεν υπάρχει καν στο λεξικό, δημιουργείται η λίστα εμφάνισης της και μπαίνει και αυτή στον Indexer. Παράλληλα ενημερώνεται το articlesDict.

Στη συνέχεια καλείται η idf_calculation η οποία διαιρεί το πλήθος των εμφανίσεων σε διαφορετικά άρθρα της κάθε λέξης με το μέγιστο αυτών, για τον υπολογισμό του idf. Ζητείται από τον χρήστη να δώσει το ερώτημα καθώς και το πλήθος των άρθρων που θέλει να του επιστραφούν και καλείται η συνάρτηση που φτιάχνει τον IndexerQ. Στο χτίσιμο αυτού ακολουθείται παρόμοια (αλλά πιο απλή καθώς μόνο ένα ερώτημα έρχεται κάθε φορά) με το χτίσιμο του Indexer.
Τελευταία συνάρτηση που καλείται είναι η similarityCalc. Για κάθε λέξη του query, ψάχνουμε να βρούμε την λίστα εμφάνισής της στον Indexer. Παίρνουμε διαδοχικά τα άρθρα στα οποία εμφανίζεται η λέξη αυτή και αν το συγκεκριμένο άρθρο έχει ήδη μπει στους accumulators τότε αυξάνουμε το σκορ του, διαφορετικά προσθέτουμε στους accumulators το λινκ του συγκεκριμένου άρθρου μαζί με το σκορ. Το σκορ υπολογίζεται πολλαπλασιάζοντας nidf(για url *ntf(για url) *

ntf(για query)*ln(2) (idf για query). Στο τέλος διαιρούμε τα σκορ με το μέγεθος του κάθε άρθρο με τη βοήθεια του articlesDict. Οι accumulators ταξινομούνται ως προς το σκορ σχετικότητας με φθίνουσα διάταξη.

Τέλος, αν ο χρήστης επιθυμεί να δει περισσότερα άρθρα από όσα είναι διαθέσιμα του εμφανίζεται σχετικό μήνυμα καθώς και όσα άρθρα έχουν βρεθεί ως σχετικά, διαφορετικά επιστρέφεται το πλήθος των σχετικών άρθρων που αναζήτησε με τα αντίστοιχα σκορ ομοιότητας στο τέλος.

Παραδείγματα εκτέλεσης:

```
(base) vasilis@vasilis-Inspiron-3576:~/Desktop$ python3 myCrawler.py https://www.sport24.gr 200 1 8
Execution time for crawling and indexing:  144.86386704444885
Enter query: Ολυμπιακός Παναθηναικός
Enter the number of the top documents: 10
1) https://www.sport24.gr/LiveMatches/basket_live/live-valenthia-panathhnaikos.5676248.html, score: 0.021425730635482962
2) https://www.sport24.gr/football/omades/Panathinaikos/panathhnaikos-sto-trifulli-ews-to-2023-o-dioudhs.5676829.html, score: 0.00724693830317806
3) https://www.sport24.gr/football/omades/Olympiakos/olympiakos-pairnei-ton-kafou.5676599.html, score: 0.006387858233395442
4) https://www.sport24.gr/Basket/Euroleague/eb-news/eb_olympiacos/olympiakos-epistolh-sth-euroleague-gia-amesh-arsh-toy-ban.5676699.html, score: 0.00626072365943812
5) https://www.sport24.gr/football/omades/Olympiakos/olympiakos-h-galata-epivevaiwnei-tis-diapragmateuseis-me-mor.5676606.html, score: 0.006121697473670631
6) https://www.sport24.gr/football/omades/Olympiakos/olympiakos-sthn-ellada-o-kafou.5676644.html, score: 0.0060655351115268645
7) https://www.sport24.gr/football/omades/Olympiakos/olympiakos-me-xasan-alla-xwris-lazaro-kontra-sthn-ksanthh.5676814.html, score: 0.005974186691172543
8) https://www.sport24.gr/football/ellada/SuperLeague/o-agkagief-sto-panathhnaikos-paok.5675911.html, score: 0.005971902114584972
9) https://www.sport24.gr/football/omades/Olympiakos/h-koph-pitas-toy-olympiakou-kai-to-video-ths-omilias-marinakh.5676520.html, score: 0.005397088384950434
10) https://www.sport24.gr/football/omades/Olympiakos/olympiakos-anakoinwse-xasan.5675866.html, score: 0.005216120924311071
(base) vasilis@vasilis-Inspiron-3576:~/Desktop$
```

```
(base) vasilis@vasilis-Inspiron-3576:~/Desktop$ python3 myCrawler.py https://news247.gr 100 1 8
Execution time for crawling and indexing:  33.48153471946716
Enter query: κορονοιος
Enter the number of the top documents: 5
1) https://www.news247.gr/ygeia/koronaios-ti-einai-kai-pos-metadidetai-osa-prepei-na-gnorizete.7574543.html, score: 0.0044957864231077345
2) https://www.news247.gr/kosmos/koronaios-se-karantina-195-amerikanoi-poy-epestrepsan-apo-oychan.7575185.html, score: 0.0024545010434981565
3) https://www.news247.gr/koinonia/koronaios-arnitiko-to-deigma-sti-thessaloniki-kinezos-kroysma-achepa-koronoios.7574571.html, score: 0.001768072785570706
4) https://www.news247.gr/kosmos/koronaios-kina-pethainoyn-dromo-kai-den-mazeyoyn.7574793.html, score: 0.0012196278477009472
5) https://www.news247.gr/technologia/, score: 0.0007194227196460113
(base) vasilis@vasilis-Inspiron-3576:~/Desktop$
```