Ανάκτηση Πληροφορίας Αναφορά Εργασίας 2021 “Δημιουργία Μηχανής Αναζήτησης”

Λυσσούδη Πασχαλίνα – 3116 Μάριος Νίκας - 3065

Περιεχόμενα:

1. Εισαγωγή ……………………………………….. 2
2. Crawler ………………………………………….. 3
3. Indexer ………………………………………….. 4
4. Query Processor ……………………………. 4
5. Αλληλεπίδραση χρήστη ………………… 5
6. Επίλογος ……………………………………….. 6

*Εισαγωγή*

Στα πλαίσια της εργασίας του μαθήματος Ανάκτησης Πληροφορίας, δημιουργήθηκε μια μηχανή αναζήτησης. Η δική μας “Google” αποτελείται από 3 βασικά τμήματα, τον Crawler, τον Indexer και τον Query Processor.

Στην πρώτη εκτέλεση του προγράμματος από την αρχή, τα επιμέρους τμήματα πυροδοτούνται με την σειρά που αναφέρθηκαν, καθώς η μηχανή αναζήτησης πρέπει πρώτα να αποκτήσει πρόσβαση σε πληροφορίες και ακολούθως να τις διαχειριστεί και να τις αποθηκεύσει. Μόνο τότε θα είναι σε θέση να μπορεί να απαντά σε ερωτήματα του χρήστη, και να του επιστρέφει ως απάντηση links σχετικά με την αναζήτηση που αυτός έκανε.

Η main ασχολείται κυρίως με την προετοιμασία του βασικού UI που χρησιμοποιείται για να δώσει ο χρήστης τις πληροφορίες εκκίνησης, και αργότερα να μεταβεί στη σελίδα της μηχανής αναζήτησης. Πιο συγκεκριμένα, για αρχή ζητείται από τον χρήστη να εισάγει ένα url, από το οποίο θα ξεκινήσει την αναζήτηση ο crawler, καθώς και ο αριθμός των σελίδων που πρέπει να προσθελαθούν για την δημιουργία του λεξικού. Επίσης, του ζητείται ο αριθμός threads που επιθυμεί να χρησιμοποιηθούν, καθώς η υλοποίηση υποστηρίζει multithreading. Στην συνέχεια, μετά από μια μικρή αναμονή, ο χρήστης ενημερώνεται οτι η διαδικασία του crawling έχει τελειώσει, και καλείται να μεταβεί στο url που αναγράφεται στην κονσόλα ώστε να μεταφερθεί στην ιστοσελίδα της μηχανής αναζήτησης.

Text

Description automatically generated

Εικόνα 1: Console log, στην τελευταία σειρά φαίνεται το link που οδηγεί

στην μηχανή αναζήτησης.

Για να γίνει η εργασία πιο φιλική προς τον χρήστη, πέρα από τις πληροφορίες για τον Crawler που δίνονται μέσω command line, έχει υλοποιηθεί μια απλή ιστοσελίδα σε html. Ο χρήστης παίρνει σύνδεσμο για την ιστοσελίδα ως απάντηση από το console, και μπορεί να χρησιμοποιήσει από εκεί την μηχανή αναζήτησης μας.

Η λειτουργία και οι δυνατότητες της ιστοσελίδας, όπως και των επιμέρους υποσυστημάτων περιγράφονται αναλυτικότερα στην συνέχεια.

*Crawler*

Ο Crawler είναι το πρώτο υποσύστημα που υλοποιήθηκε. Σκοπός του είναι να ξεκινάει από μια δοθείσα ιστοσελίδα και να μαζεύει όλες τις λέξεις που εμφανίζονται σε αυτή, με στόχο την δημιουργία ενός λεξικού. Παράλληλα, μαζεύει άλλα urls που εμφανίζονται στην σελίδα, ώστε να μεταβεί στην επόμενη, όταν τελειώσει με αυτή. Το πόσες μεταβάσεις θα κάνει από ιστοσελίδα σε ιστοσελίδα επίσης καθορίζεται από τον χρήστη για επιτυχία περατότητας, καθώς η υλοποίηση των πρακτικών των crawlers μεγάλων μηχανών αναζήτησης είναι αδύνατες στα πλαίσια μιας εργασίας, λόγω περιορισμένου χώρου αποθήκευσης και υπολογιστικής ισχύος.

Ο carawler, λοιπόν, λαμβάνει τις απαραίτητες πληροφορίες από την main, και εκτελείται σύμφωνα με 4 μεθόδους.

1. Η get\_text διαβάζει το κείμενο μιας σελίδας, οργανωμένο σε παραγράφους, με την βοήθεια της βιβλιοθήκης BeautifulSoup.
2. H thread\_get\_text καλεί την get\_text και γράφει τα αποτελέσματά της σε ένα προϋπάρχον αρχείο, όπου συσχετίζεται το κείμενο με το url από όπου προέρχεται, ενώ παραλείπονται ήδη υπάρχουσες λέξεις.
3. Η συνάρτηση get\_links\_from διαχειρίζεται επίσης μια “soup” από την οποία διαχωρίζει τα links και τα βάζει σε έναν πίνακα. Ο πίνακας έχει μέγιστο μέγεθος τον αριθμό των σελίδων που έχει εισάγει ο χρήστης ως σελίδες προς προσπέλαση.
4. Τέλος, η συνάρτηση crawler έχει πρόσβαση σε όλες τις προηγούμενες πληροφορίες, και με βάση αυτές δημιουργεί ένα λεξικό, στο οποίο τα urls χρησιμοποιούνται ως κλειδιά, και οι λέξεις που εμφανίζονται σε αυτά ως τιμές τους. Αυτό προϋποθέτει οτι οι μεμονωμένες λέξεις έχουν προεπεξεργαστεί για να αφαιρεθούν “stop words”, σημεία στίξης, απόστροφοι κλπ, να γραφούν όλες οι λέξεις με πεζά και να βρεθεί η ρίζα τους. Αυτό συμβαίνει ώστε να μην θεωρούνται διαφορετικές λέξεις ρήματα σε άλλο πρόσωπο, ουσιαστικά σε άλλη πτώση ή αριθμό, επίθετα σε άλλο γένος κλπ.

Επίσης, όπως προαναφέρθηκε, ο crawler υποστηρίζει  multithreading. Αυτό σημαίνει οτι σε περίπτωση που ο χρήστης εισάγει οτι επιθυμεί να χρησιμοποιηθούν περισσότερα από 1 threads, τότε δημιουργείται ένα λεξικό σε κάθε thread, και στο τέλος ενώνονται μεταξύ τους.

Το λεξικό αυτό αποθηκεύεται στο αρχείο data.csv .

*Indexer*

Ο Indexer είναι το δεύτερο υποσύστημα τη μηχανής αναζήτησης . Σκοπός του είναι να παράγει τον αντεστραμμένο κατάλογο, όπως λέγεται, που με την σειρά του χρησιμοποιείται στην διαδικασία των queries.

Στην υλοποίησή μας, αποτελείται από δύο μεθόδους.

1. Η Index χρησιμοποιείται για την δημιουργία του αντεστραμμένου καταλόγου. Αυτό το επιτυγχάνει διαβάζοντας το αρχείο data.csv και βάσει αυτού φτιάχνοντας ένα λεξικό που έχει ως κλειδιά τα url και ως τιμές τις συχνότητες εμφανίσεων των λέξεων σε κάθε url ξεχωριστά.
2. Η count\_words δημιουργεί ένα παρόμοιο λεξικό, με την διαφορά οτι η τιμή για κάθε κλειδί είναι ο συνολικός αριθμός λέξεων που εμφανίζονται στο συγκεκριμένο url. Οι τιμές αυτές μπορούν μελλοντικά να χρησιμοποιηθούν και για εύρεση σημαντικότητας μιας λέξης σε ένα url, ώστε να προκύψει η προτεραιότητα των urls στην λίστα εμφάνισης αποτελεσμάτων με βάση τη σχετικότητά τους.

*Query Processor*

Ο Query Processor είναι το υποσύστημα που διαχειρίζεται τις αναζητήσεις χρήστη. Το σκεπτικό είναι οτι δέχεται μια ή περισσότερες λέξεις και έναν αριθμό αποτελεσμάτων Κ που περιμένει ο χρήστης σαν απάντηση. Με βάση τον αντεστραμμένο κατάλογο που έχει κατασκευαστεί από τον Indexer, βρίσκει τα Κ url που έχει προσπελάσει κάποια στιγμή ο Crawler που είναι πιο σχετικά με τις λέξεις που ψάχνει ο χρήστης.

Για την επίτευξη της διαδικασίας αυτής, χρησιμοποιούνται 9 συναρτήσεις.

1. Η read\_query δέχεται ένα string, το επεξεργάζεται για να μην περιορίζεται από γραμματικά φαινόμενα και το αποθηκεύει σε λίστα ΤextBlob μορφής.
2. Η freq βρίσκει την συχνότητα εμφάνισης μιας λέξεις σε μια συγκεκριμένη σελίδα.
3. Η n\_freq βρίσκει την κανονικοποιημένη συχνότητα εμφάνισης ως προ; τι; υπόλοιπες λέξεις του κειμένου της σελίδας.
4. Η idf βρίσκει την αντεστραμμένη συχνότητα εγγράφου.
5. Η n\_idf βρίσκει την κανονικοποιημένη αντεστραμμένη συχνότητα εγγράφου.
6. Η weight\_in\_page χρησιμοποιεί τις προηγούμενες συναρτήσεις για να βρει το βάρος μιας λέξης σε μια σελίδα.
7. Η weight\_in\_query βρίσκει το βάρος μιας λέξης, αλλά μέσα στην αναζήτηση του χρήστη.
8. Η cosine\_similarity βρίσκει την συνημιτονική ομοιότητα ενός query με όλες τις σελίδες ξεχωριστά.
9. Τέλος, η top\_k βρίσκει τις Κ πιο σχετικές σελίδες στο query, ώστε να τις εμφανίσει ως απάντηση στον χρήστη.

Επίσης, υπάρχει η δυνατότητα αλληλεπίδρασης του χρήστη με τα αποτελέσματα, καθώς μπορεί να τα ανακατατάξει ανάλογα με το ποιά πιστεύει αυτός οτι είναι πιο σχετικά. Αυτό γίνεται με την βοήθεια 2 ακόμα μεθόδων.

1. Η feedback χρησιμοποιεί τον αλγόριθμο του Rocchio για συνυπολογισμό της ανατροφοδότησης χρήστη στην κατάταξη των αποτελεσμάτων.
2. Η update\_results εμφανίζει τα ενημερωμένα αποτελέσματα στον χρήστη.

*Αλληλεπίδραση χρήστη*

Όλες οι διαδικασίες που έχουν αναφερθεί μέχρι αυτό το σημείο γίνονται κρυφά, στο background. Ο χρήστης έχει δώσειτο link εκκίνησης της αναζήτησης σελίδων για τον crawler, αλλά ο βασικός του ρόλος είναι να δώσει ένα ερώτημα στην μηχανή αναζήτησης. Αυτό το κάνει μέσω μιας πολύ απλής σελίδας που φτιάχτηκε ως user interface.

Νωρίτερα αναφέρθηκε ότι όταν τελειώνει η διαδικασία του crawling, εμφανίζεται στην κονσόλα ένα link (η ιστοσελίδα τρέχει τοπικά στο μηχάνημα), που οδηγεί στην παρακάτω σελίδα.

Graphical user interface, text, application, chat or text message

Description automatically generated

Εικόνα 2: Αρχική σελίδα μηχανής αναζήτησης.

Ο χρήστης μπορεί να αλληλεπιδράσει με την σελίδα και να εισάγει στα κατάλληλα πεδία το ερώτημά του και πόσα αποτελέσματα θέλει ως απάντηση. Πατώντας υποβολή, θα του εμφανιστεί ο ζητούμενος αριθμός αποτελεσμάτων ως εξής:

Graphical user interface, text, application, chat or text message

Description automatically generatedΕικόνα 3 : αποτελέσματα αναζήτησης

Από εκεί, ο χρήστης μπορεί να ανοίξει τα link που εμφανίζονται και να αποκτήσει πρόσβαση στις πληροφορίες που τον ενδιαφέρουν.

Επίσης, μπορεί, πατώντας την λέξη «Σχετικό» δίπλα από τα links, να δηλώσει αν θεωρεί ότι κάποια ιστοσελίδα είναι πιο σχετική προς την αναζήτηση που έκανε συγκριτικά με κάποια άλλη ιστοσελίδα, ώστε να αναδιαμορφώσει την κατάταξή τους ως προς τη σχετικότητα.

*Επίλογος*

Με το πέρας της εργασίας αυτής, αντιλαμβανόμαστε οτι τελικά δεν είναι τόσο δύσκολο να φτιάξεις την δική σου Google. Αναμφίβολα δεν θα είναι τόσο αποδοτική, κυρίως λόγω υλικών περιορισμών. Δεν έχουμε όλοι τεράστιες αποθήκες δεδομένων για να αποθηκεύουμε αντεστραμμένους καταλόγους, ούτε τόσο μεγάλη υπολογιστική ισχύ ώστε να βγάζουν τα μηχανήματά μας αποτέλεσμα σε ερωτήματα μέσα σε κλάσματα δευτερολέπτου, έχοντας υπολογίσει δισεκατομμύρια σελίδες.

Υπάρχουν, ωστόσο, τρόποι να βελτιώσουμε τους χρόνους και να μειώσουμε την απαιτούμενη μνήμη.

Αν ψάξουμε, όλο και κάτι θα βρούμε...