Εργασία Ανάκτησης Πληροφορίας

Μάθημα: Ανάκτηση Πληροφορίας (ICE1-7102)

Μαθητές: Γεώργιος Θεοχάρης (19390283), Γεώργιος Σάντος (20390204)

Καθηγήτρια: Παναγιώτα Τσελέντη

Gitlab Εργασίας: https://github.com/geotheo01/anakthsh

Δημιουργία μηχανής αναζήτησης

Εισαγωγή

Η εργασία αυτή επικεντρώνεται στην πρακτική εξάσκηση και την ανάπτυξη μιας απλής μηχανής αναζήτησης, με σκοπό την κατανόηση των θεμελιωδών εννοιών της ανάκτησης πληροφορίας, της ευρετηρίασης, της κατάταξης και της αναζήτησης, καθώς και την απόκτηση πρακτικών δεξιοτήτων στην επεξεργασία φυσικής γλώσσας και στην εφαρμογή αλγορίθμων αναζήτησης. Η εργασία περιλαμβάνει τα παρακάτω στάδια: συλλογή δεδομένων, προεπεξεργασία κειμένου, δημιουργία ευρετηρίου, υλοποίηση της μηχανής αναζήτησης και αξιολόγηση της απόδοσής της.

Κατά το πρώτο βήμα, συλλέχθηκαν έγγραφα από την Wikipedia με τη χρήση κατάλληλου web crawler γραμμένου σε Python, ο οποίος αξιοποίησε τη βιβλιοθήκη BeautifulSoup. Ο web crawler ανέσυρε περιεχόμενο από συγκεκριμένα URLs που σχετίζονται με θέματα όπως "Natural Language Processing", "Machine Learning" και "Artificial Intelligence” και άλλες σελίδες οι οποίες δεν έχουν σχέση άμεση με αυτά τα θέματα όπως σελίδες για ασθένειες, φαγητά και το πανεπιστήμιό μας. Κάθε άρθρο που συλλέχθηκε περιλαμβάνει τον τίτλο, το περιεχόμενο και τη διεύθυνση URL του. Τα δεδομένα αποθηκεύτηκαν σε μορφή JSON για εύκολη πρόσβαση και διαχείριση.

Ερώτημα 1

Το πρώτο βήμα της εργασίας αφορά τη συλλογή δεδομένων από το Wikipedia, το οποίο υλοποιείται μέσω ενός web crawler σε Python, με χρήση της βιβλιοθήκης BeautifulSoup. Ο κώδικας αναλαμβάνει να συλλέξει δεδομένα από άρθρα του Wikipedia σχετικά με συγκεκριμένα θέματα, όπως τα "Natural language processing", "Machine learning" και "Artificial intelligence".

Ο web crawler λειτουργεί με την αποστολή αιτημάτων HTTP στους διακομιστές του Wikipedia για κάθε άρθρο που επιλέγεται. Η διεύθυνση URL του κάθε άρθρου δημιουργείται δυναμικά, βασισμένη στον τίτλο του θέματος. Για παράδειγμα, το θέμα "Natural language processing" μετατρέπεται σε URL https://en.wikipedia.org/wiki/Natural_language_processing. Ο κώδικας ελέγχει αν η αίτηση ήταν επιτυχής (HTTP status code 200) και στη συνέχεια χρησιμοποιεί τη βιβλιοθήκη BeautifulSoup για την ανάλυση της HTML σελίδας και την εξαγωγή των παραγράφων του κειμένου.

Η συνάρτηση soup.find_all('p') συλλέγει όλα τα παραγράφους από τη σελίδα και, στη συνέχεια, οι πρώτες παραγράφοι αυτών συνδυάζονται για να δημιουργηθεί το περιεχόμενο του άρθρου. Για κάθε άρθρο, αποθηκεύεται το περιεχόμενο, ο τίτλος και το URL σε μια λίστα από dictionaries, το καθένα από τα οποία αντιπροσωπεύει ένα άρθρο με την πληροφορία του.

Τα συλλεχθέντα δεδομένα περιλαμβάνουν τον τίτλο του άρθρου, το περιεχόμενό του και τη διεύθυνση URL. Όλα τα δεδομένα αποθηκεύονται σε ένα αρχείο JSON με το όνομα collected_data.json. Αυτό το αρχείο επιτρέπει την εύκολη αποθήκευση και διαχείριση των

δεδομένων για τα επόμενα στάδια της εργασίας, όπως η προεπεξεργασία και η δημιουργία ευρετηρίου. Αυτό το αρχείο JSON περιέχει τη δομή των άρθρων με τις αντίστοιχες παραγράφους και τα URLs τους. Σε περίπτωση σφάλματος, ο κώδικας χειρίζεται καταστάσεις αποτυχίας σύνδεσης ή αποτυχίας λήψης δεδομένων, αποτρέποντας την απώλεια δεδομένων.

Αυτό το βήμα εξασφαλίζει τη συλλογή ενός επαρκούς και οργανωμένου συνόλου δεδομένων για την προεπεξεργασία και την επόμενη φάση της δημιουργίας του ανεστραμμένου ευρετηρίου, η οποία είναι απαραίτητη για τη λειτουργία της μηχανής αναζήτησης.

Ερώτημα 2

Στο δεύτερο βήμα της εργασίας, το οποίο αφορά την προεπεξεργασία των δεδομένων, ο στόχος είναι να καθαρίσουμε και να ετοιμάσουμε τα κείμενα για τη διαδικασία ευρετηρίασης και αναζήτησης. Ο κώδικας για την προεπεξεργασία περιλαμβάνει αρκετές βασικές τεχνικές επεξεργασίας φυσικής γλώσσας, όπως η αφαίρεση ειδικών χαρακτήρων, η κατάτμηση (tokenization), η αφαίρεση των stopwords και η εκτέλεση stemming.

Αρχικά, η συνάρτηση preprocess_text αναλαμβάνει την προετοιμασία του κάθε κειμένου. Χρησιμοποιεί τη βιβλιοθήκη re για την αφαίρεση οποιωνδήποτε μη-αλφαβητικών χαρακτήρων (όπως αριθμοί ή σύμβολα), ώστε να παραμείνει μόνο το λεκτικό περιεχόμενο του κειμένου. Αυτό το βήμα είναι σημαντικό, καθώς οι μη-αλφαβητικοί χαρακτήρες ή τα σημεία στίξης δεν προσφέρουν πληροφορίες για την αναζήτηση και μπορούν να προκαλέσουν λάθη κατά την ανάλυση του κειμένου.

Μετά την αφαίρεση των ειδικών χαρακτήρων, η συνάρτηση χρησιμοποιεί τη βιβλιοθήκη nltk για την κατάτμηση του κειμένου σε επιμέρους λέξεις ή "tokens". Η συνάρτηση word_tokenize του NLTK επιτρέπει την ανάλυση του κειμένου σε λέξεις, αναγνωρίζοντας επίσης λέξεις που συνδέονται με άλλα σύμβολα (όπως ', ", κ.λπ.), δημιουργώντας πιο ακριβή tokens. Αυτή η κατάτμηση είναι το πρώτο βήμα για την επεξεργασία κειμένου, καθώς καθιστά το κείμενο κατάλληλο για περαιτέρω αναλύσεις, όπως η αφαίρεση των stopwords και το stemming.

Η αφαίρεση των stopwords είναι ένα κρίσιμο βήμα στην επεξεργασία κειμένου. Οι stopwords είναι λέξεις που δεν προσφέρουν σημαντική πληροφορία για την αναζήτηση, όπως "και", "το", "ή", κ.λπ. Στη συνέχεια, η συνάρτηση φιλτράρει αυτές τις λέξεις από το σύνολο των tokens χρησιμοποιώντας τη λίστα stopwords της βιβλιοθήκης NLTK. Το αποτέλεσμα είναι μια μικρότερη και πιο σημαντική λίστα λέξεων για περαιτέρω επεξεργασία.

Το επόμενο βήμα είναι το stemming, που εκτελείται με τη χρήση του Porter Stemmer της NLTK. Ο stemmer είναι ένας αλγόριθμος που μετατρέπει τις λέξεις σε βασική μορφή ή "ριζικό" σχήμα. Για παράδειγμα, η λέξη "running" μπορεί να μετατραπεί στην ρίζα "run", και η λέξη "better" στη ρίζα "better". Αυτό βοηθά στην εξομάλυνση των λέξεων και στη μείωση των παραλλαγών που ενδέχεται να προκύψουν από τη χρήση συνωνύμων ή παραλλαγμένων μορφών λέξεων.

Η διαδικασία της προεπεξεργασίας είναι απαραίτητη για να βελτιωθεί η ποιότητα της αναζήτησης, καθώς επιτρέπει στο σύστημα να εργάζεται με καθαρές και ομοιογενείς μορφές των δεδομένων, αποφεύγοντας ανακριβείς ή περιττές πληροφορίες. Επίσης, η προεπεξεργασία βελτιώνει την ταχύτητα και την αποδοτικότητα των αλγορίθμων κατάταξης, καθώς μειώνει τον αριθμό των λέξεων που πρέπει να αναλυθούν και να ανακτηθούν κατά την αναζήτηση.

Ερώτημα 3

Στο τρίτο βήμα της εργασίας, το οποίο αφορά τη δημιουργία του ευρετηρίου, η διαδικασία επικεντρώνεται στη σχεδίαση και υλοποίηση του ανεστραμμένου ευρετηρίου (inverted index), το οποίο είναι ένα κρίσιμο στοιχείο για την αποτελεσματική αναζήτηση εγγράφων. Το ανεστραμμένο ευρετήριο είναι μια δομή δεδομένων που αντιστοιχεί κάθε λέξη (ή όρο) σε μια λίστα από τα έγγραφα στα οποία εμφανίζεται αυτή η λέξη. Ο σκοπός του είναι να επιτρέπει την ταχεία αναζήτηση λέξεων στα έγγραφα χωρίς την ανάγκη να αναλύεται το πλήρες σύνολο των εγγράφων κάθε φορά που γίνεται μια αναζήτηση.

Η συνάρτηση build_inverted_index είναι υπεύθυνη για τη δημιουργία του ανεστραμμένου ευρετηρίου. Ξεκινώντας από το σύνολο των δεδομένων, η συνάρτηση επεξεργάζεται κάθε έγγραφο ξεχωριστά. Για κάθε έγγραφο, η συνάρτηση εξετάζει το περιεχόμενο και, για κάθε λέξη (ή όρο) στο κείμενο, προσθέτει την αντίστοιχη εγγραφή στο ευρετήριο. Αν η λέξη δεν υπάρχει ήδη στο ευρετήριο, τότε προστίθεται με την αντίστοιχη λίστα των εγγράφων στα οποία εμφανίζεται. Το αποτέλεσμα είναι μια δομή δεδομένων όπου κάθε λέξη έχει μια λίστα από τα έγγραφα στα οποία εμφανίζεται.

Για την υλοποίηση της διαδικασίας αυτής χρησιμοποιείται το defaultdict, μια δομή δεδομένων από τη βιβλιοθήκη collections της Python. Το defaultdict επιτρέπει την εύκολη προσθήκη νέων λέξεων στο ευρετήριο χωρίς την ανάγκη ελέγχου αν η λέξη υπάρχει ήδη. Όταν μια λέξη εντοπιστεί σε ένα έγγραφο, το αντίστοιχο ID του εγγράφου προστίθεται στη λίστα της λέξης. Η χρήση του set διασφαλίζει ότι κάθε έγγραφο εμφανίζεται μόνο μία φορά για κάθε λέξη, αποφεύγοντας τις επαναλήψεις και βελτιώνοντας την αποδοτικότητα του ευρετηρίου.

Μετά τη δημιουργία του ανεστραμμένου ευρετηρίου, η συνάρτηση επιστρέφει το τελικό ευρετήριο, το οποίο είναι ένα λεξικό όπου τα κλειδιά είναι οι λέξεις και οι τιμές είναι λίστες των εγγράφων στα οποία εμφανίζονται. Το ευρετήριο αυτό μπορεί στη συνέχεια να αποθηκευτεί σε αρχείο JSON με το όνομα “inverted_index.json" για μελλοντική χρήση.

Η δημιουργία του ανεστραμμένου ευρετηρίου είναι ένα θεμελιώδες βήμα στην ανάπτυξη μιας μηχανής αναζήτησης, καθώς επιτρέπει την ταχεία αναγνώριση των εγγράφων που περιέχουν συγκεκριμένες λέξεις-κλειδιά. Χωρίς αυτή τη δομή, η αναζήτηση θα απαιτούσε την ανάλυση του πλήρους κειμένου κάθε εγγράφου για κάθε ερώτημα, κάτι που θα ήταν ιδιαίτερα αργό και αναποτελεσματικό. Το ευρετήριο επιτρέπει τη γρήγορη ανεύρεση των σχετικών εγγράφων και αποτελεί τη βάση για την αποτελεσματική κατάταξη των αποτελεσμάτων αναζήτησης.

Ερώτημα 4

Στο τέταρτο βήμα της εργασίας, η ανάπτυξη της μηχανής αναζήτησης περιλαμβάνει τη δημιουργία μιας διεπαφής χρήστη για την υποβολή ερωτημάτων και την ανάκτηση σχετικών εγγράφων. Το κύριο μέρος της διαδικασίας αυτής είναι η ανάπτυξη του επεξεργαστή ερωτημάτων (Query Processor) και η κατάταξη των αποτελεσμάτων (Ranking). Η μηχανή αναζήτησης θα πρέπει να είναι ικανή να επεξεργάζεται ερωτήματα με χρήση λεκτικών όρων και να επιστρέφει τα έγγραφα που είναι πιο σχετικά με το ερώτημα του χρήστη, χρησιμοποιώντας διαφορετικούς αλγορίθμους κατάταξης.

Επεξεργασία Ερωτημάτων (Query Processing)

Η διαδικασία επεξεργασίας του ερωτήματος είναι καθοριστική για την αποτελεσματικότητα του συστήματος αναζήτησης. Ο επεξεργαστής ερωτημάτων, που βρίσκεται στο αρχείο beautifulsoup.ipynb, αναλαμβάνει να δέχεται το ερώτημα από τον χρήστη, να το κατατμήσει σε επιμέρους λέξεις (tokens), και να το αναλύσει για την αναγνώριση των όρων που θα χρησιμοποιηθούν για την αναζήτηση.

Η συνάρτηση process_query_boolean εκτελεί βασικές λειτουργίες Boolean, όπως AND, OR και NOT. Αν το ερώτημα περιλαμβάνει τη λέξη "OR", η συνάρτηση αναζητά όλα τα έγγραφα που περιέχουν τουλάχιστον έναν από τους όρους του ερωτήματος. Εάν το ερώτημα περιλαμβάνει τη λέξη "AND", η συνάρτηση αναζητά έγγραφα που περιέχουν όλους τους όρους του ερωτήματος. Τέλος, αν το ερώτημα περιλαμβάνει τη λέξη "NOT", η συνάρτηση αναζητά όλα τα έγγραφα που δεν περιέχουν τον όρο που ακολουθεί.

Αυτός ο τύπος επεξεργασίας επιτρέπει στους χρήστες να διατυπώσουν πιο πολύπλοκα ερωτήματα και να περιορίσουν ή να διευρύνουν την αναζήτησή τους, ενισχύοντας την ακρίβεια των αποτελεσμάτων.

Αλγόριθμοι Κατάταξης (Ranking Algorithms)

Η κατάταξη των αποτελεσμάτων είναι ένα άλλο βασικό στοιχείο της μηχανής αναζήτησης. Η κατάταξη των εγγράφων επιτρέπει στην μηχανή να επιστρέψει τα πιο σχετικά έγγραφα στην κορυφή της λίστας αποτελεσμάτων. Στην εργασία, χρησιμοποιούνται τρεις κύριοι αλγόριθμοι για την κατάταξη:

1. **TF-IDF (Term Frequency-Inverse Document Frequency)**: Ο αλγόριθμος TF-IDF χρησιμοποιείται για να υπολογίσει τη σημασία κάθε όρου σε σχέση με το σύνολο των εγγράφων. Όροι που εμφανίζονται συχνά σε ένα έγγραφο, αλλά σπάνια σε άλλα, θεωρούνται πιο σημαντικοί. Η συνάρτηση calculate_tfidf στο αρχείο beautifulsoup.ipynb υπολογίζει την τιμή TF-IDF για κάθε όρο και για κάθε έγγραφο της βάσης δεδομένων.
2. **Vector Space Model (VSM)**: Το VSM μετατρέπει τα έγγραφα και τα ερωτήματα σε διανύσματα και υπολογίζει τη σχετικότητα μεταξύ τους μέσω της συνάρτησης συνημιτόνου (cosine similarity). Ο συντελεστής συσχέτισης δείχνει πόσο κοντά είναι τα

αποτελέσματα στο αρχικό ερώτημα. Αυτός ο αλγόριθμος είναι ιδιαίτερα αποτελεσματικός για ερωτήματα πολλαπλών όρων και παρέχει πιο σχετικές κατατάξεις.

3. **Okapi BM25**: Ο αλγόριθμος BM25 είναι ένα μοντέλο πιθανοτήτων για την ανάκτηση εγγράφων, το οποίο βασίζεται σε δύο παραμέτρους: την συχνότητα εμφάνισης των όρων (tf) και την αντίστροφη συχνότητα εγγράφων (idf). Αυτή η μέθοδος είναι γνωστή για την ακρίβεια της και χρησιμοποιείται ευρέως σε προηγμένα συστήματα αναζήτησης. Η συνάρτηση process_query_bm25 υπολογίζει την τιμή BM25 για κάθε έγγραφο και κάθε όρο.

Παρουσίαση Αποτελεσμάτων

Αφού ολοκληρωθεί η κατάταξη των εγγράφων, η μηχανή αναζήτησης εμφανίζει τα αποτελέσματα στον χρήστη. Για κάθε έγγραφο που ανακτάται, εμφανίζεται ο τίτλος του και η βαθμολογία του, η οποία δείχνει τη συσχέτιση του εγγράφου με το ερώτημα. Ο χρήστης μπορεί να δει τα πιο συναφή αποτελέσματα στην κορυφή και να αποφασίσει ποιο έγγραφο να εξετάσει.

Αξιολόγηση Αποτελεσμάτων

Μετά την κατάταξη, η μηχανή αναζήτησης αξιολογεί την απόδοση των αποτελεσμάτων χρησιμοποιώντας μετρικές όπως η ακρίβεια (precision), η ανάκληση (recall) και ο δείκτης F1 (F1-score). Αυτές οι μετρικές υπολογίζονται από την συνάρτηση evaluate_retrieval, η οποία συγκρίνει τα ανακτηθέντα έγγραφα με τα πραγματικά σχετικά έγγραφα. Επιπλέον, υπολογίζεται η μέση ακρίβεια (MAP) μέσω της συνάρτησης calculate_map, η οποία παρέχει μια συνολική αξιολόγηση της απόδοσης του συστήματος.

Η δημιουργία αυτής της μηχανής αναζήτησης παρέχει στους χρήστες μια ισχυρή και αποδοτική μέθοδο για την ανεύρεση και κατάταξη εγγράφων, με διάφορους αλγορίθμους που μπορούν να επιλέξουν ανάλογα με τις ανάγκες τους, προσφέροντας ευελιξία και ακρίβεια στην αναζήτηση πληροφοριών.

Ερώτημα 5

Στο πέμπτο βήμα της εργασίας, η αξιολόγηση της απόδοσης της μηχανής αναζήτησης είναι κρίσιμη για την εκτίμηση της αποτελεσματικότητας και της ποιότητας των αποτελεσμάτων που επιστρέφονται για τα ερωτήματα των χρηστών. Η αξιολόγηση πραγματοποιείται με τη χρήση κοινών μετρικών απόδοσης όπως η ακρίβεια (precision), η ανάκληση (recall), η μέση ακρίβεια (Mean Average Precision - MAP), καθώς και ο δείκτης F1 (F1-score), οι οποίες παρέχουν σημαντικές πληροφορίες για το πόσο καλά λειτουργεί η μηχανή αναζήτησης και πόσο σχετικά είναι τα αποτελέσματα με τα αναμενόμενα.

Μέτρηση Ακρίβειας (Precision) και Ανάκλησης (Recall)

Η ακρίβεια και η ανάκληση είναι δύο από τις βασικές μετρικές που χρησιμοποιούνται για να αξιολογηθεί η ποιότητα των αποτελεσμάτων. Η ακρίβεια υπολογίζεται ως ο λόγος των σχετικών εγγράφων που ανακτήθηκαν προς τον συνολικό αριθμό των εγγράφων που επιστράφηκαν. Επομένως, η ακρίβεια δείχνει το ποσοστό των ανακτηθέντων εγγράφων που είναι πραγματικά σχετικά με το ερώτημα του χρήστη. Η συνάρτηση evaluate_retrieval στο αρχείο evaluation.py υπολογίζει την ακρίβεια συγκρίνοντας τα ανακτηθέντα έγγραφα με το σύνολο των πραγματικά σχετικών εγγράφων. Εάν τα ανακτηθέντα έγγραφα περιλαμβάνουν τα σωστά έγγραφα, η ακρίβεια θα είναι υψηλή.

Η ανάκληση, από την άλλη πλευρά, υπολογίζεται ως ο λόγος των σχετικών εγγράφων που ανακτήθηκαν προς τον συνολικό αριθμό των σχετικών εγγράφων στην αναζήτηση. Η ανάκληση δείχνει πόσα από τα πραγματικά σχετικά έγγραφα καταφέρνει η μηχανή αναζήτησης να ανακτήσει. Αν η μηχανή αναζήτησης επιστρέφει όλα τα σχετικά έγγραφα, τότε η ανάκληση θα είναι 100%. Και οι δύο αυτές μετρικές είναι σημαντικές για να εκτιμηθεί πόσο καλή είναι η αναζήτηση σε όρους τόσο ποσότητας όσο και ποιότητας των αποτελεσμάτων.

Μέσο Δείκτης F1 (F1-Score)

Ο δείκτης F1 είναι μια μετρική που συνδυάζει την ακρίβεια και την ανάκληση σε μία ενιαία τιμή, η οποία υπολογίζεται ως το αρμονικό μέσο της ακρίβειας και της ανάκλησης. Ο δείκτης F1 είναι χρήσιμος όταν υπάρχει ανάγκη για ισορροπία μεταξύ των δύο αυτών μετρικών, για παράδειγμα, όταν είναι σημαντικό να έχουμε τόσο υψηλή ακρίβεια όσο και υψηλή ανάκληση. Η συνάρτηση evaluate_retrieval υπολογίζει επίσης τον δείκτη F1, δίνοντας μια συνολική εικόνα της απόδοσης της μηχανής αναζήτησης σε σχέση με τις δύο βασικές μετρικές.

Μέση Ακρίβεια (MAP)

Ο υπολογισμός της μέσης ακρίβειας (Mean Average Precision - MAP) είναι μια πολύτιμη μέθοδος για την αξιολόγηση της μηχανής αναζήτησης όταν υπάρχει ένα σύνολο ερωτημάτων και επιθυμούμε να μετρήσουμε την ακρίβεια της μηχανής στο σύνολό τους. Η μέση ακρίβεια υπολογίζεται ως ο μέσος όρος της ακρίβειας για κάθε ανακτηθέν έγγραφο σε κάθε ερώτημα. Αυτός ο υπολογισμός προσφέρει μια ακριβέστερη εκτίμηση της γενικής απόδοσης της μηχανής αναζήτησης σε όλα τα ερωτήματα που δοκιμάστηκαν.

Συμπεράσματα και Προτεινόμενες Βελτιώσεις

Αφού γίνει η αξιολόγηση της μηχανής αναζήτησης, είναι σημαντικό να αναλυθούν τα αποτελέσματα και να εντοπιστούν τα δυνατά και τα αδύνατα σημεία του συστήματος. Αν η ακρίβεια είναι χαμηλή ή η ανάκληση δεν είναι ικανοποιητική, μπορεί να χρειαστεί βελτίωση του αλγορίθμου κατάταξης ή της προεπεξεργασίας των δεδομένων. Επιπλέον, μπορεί να εξεταστούν άλλοι αλγόριθμοι αναζήτησης ή τεχνικές βελτιστοποίησης για τη βελτίωση της απόδοσης του συστήματος, όπως η ενσωμάτωση προχωρημένων μεθόδων NLP και η χρήση μεγαλύτερων συνόλων δεδομένων για πιο ρεαλιστική αξιολόγηση.