Εργασία Ανάκτησης Πληροφορίας

Δημιουργία μηχανής αναζήτησης

Η εργασία αφορά στην πρακτική εξάσκηση και εμπειρία στη δημιουργία μιας απλής μηχανής αναζήτησης με σκοπό την κατανόηση των θεμελιωδών εννοιών της ανάκτησης πληροφορίας, της ευρετηρίασης, της κατάταξης και της αναζήτησης πληροφορίας, καθώς και πρακτικές δεξιότητες στην επεξεργασία φυσικής γλώσσας και την εφαρμογή αλγορίθμων αναζήτησης.

Στόχοι της εργασίας:

- Ο σχεδιασμός και υλοποίηση ενός συστήματος ανάκτησης εγγράφων που μπορεί να ευρετηριάσει και να αναζητήσει αποτελεσματικά σε μια συλλογή εγγράφων κειμένου.
- Η ανάπτυξη αλγορίθμων ανάκτησης και η αξιολόγηση της απόδοσή τους χρησιμοποιώντας κοινές μετρήσεις αξιολόγησης, όπως ακρίβεια, ανάκληση και βαθμολογία F1.
- Η παροχή μιας φιλικής προς το χρήστη διεπαφής για τους χρήστες να εισάγουν ερωτήματα και να ανακτούν σχετικά έγγραφα.
- Η απόκτηση πρακτικής εμπειρίας σε διάφορες τεχνικές ανάκτησης πληροφοριών, όπως Boolean retrieval, Vector Space Model και Probabilistic retrieval models.

Περιγραφή εργασίας
Υλοποιείστε μια μηχανή αναζήτησης που ανακτά έγγραφα από το Wikipedia με βάση τα ερωτήματα των χρηστών.

Βήμα 1. Συλλογή δεδομένων:

α. Συλλέξτε έγγραφα από το Wikipedia
β. Υλοποιήστε έναν web crawler σε Python (π.χ. με BeautifulSoup) για τη συλλογή εγγράφων από την επιλεγμένη πηγή. Εναλλακτικά, βρείτε ένα έτοιμο σύνολο δεδομένων με άρθρα του Wikipedia. Η συλλογή πρέπει να έχει έναν ικανοποιητικό αριθμό εγγράφων χωρίς ωστόσο να είναι απαραίτητο να είναι μακροσκελή. Μπορείτε να βρείτε βοήθεια στα:
    - https://datamam.com/tutorial-how-to-scrape-wikipedia/
    - https://dev.to/admantium/nlp-project-wikipedia-article-crawler-classification-corpus-reader-dik
γ. Αποθηκεύστε τα δεδομένα που συλλέγονται σε αρχείο μορφής JSON ή CSV.

Ορόσημο: Ένα σύνολο δεδομένων με την περιγραφή του και την περιγραφή της μεθοδολογίας συλλογής του.

Βήμα 2. Προεπεξεργασία κειμένου (Text Processing):

Κάντε προεπεξεργασία του κειμενικού περιεχομένου. Αυτό μπορεί να περιλαμβάνει εργασίες όπως tokenization, stemming/lemmatization και stop-word removal και αφαίρεση ειδικών χαρακτήρων (removing special characters).
Ορόσημο: Το «καθαρισμένο» σύνολο δεδομένων και περιγραφή των εργασιών που επιλέχθηκαν.

Βήμα 3. Ευρετήριο (Indexing):

α. Δημιουργήστε μια ανεστραμμένη δομή δεδομένων ευρετηρίου (inverted index) για την αποτελεσματική αντιστοίχιση όρων στα έγγραφα στα οποία εμφανίζονται.

β. Εφαρμόστε μια δομή δεδομένων για την αποθήκευση του ευρετηρίου.

Ορόσημο: Το αντεστραμμένο ευρετήριο με περιγραφή του σχεδιασμού και της υλοποίησής του.

Βήμα 4. Μηχανή αναζήτησης (Search Engine): Αναπτύξτε μια διεπαφή χρήστη για την αναζήτηση όρων χρησιμοποιώντας την Python (π.χ. μια διεπαφή γραμμής εντολών ή μια απλή διεπαφή ιστού εντός του Jupyter Notebook).

α) Επεξεργασία ερωτήματος (Query Processing): Αναπτύξτε ένα module επεξεργασίας ερωτημάτων που θα προεπεξεργάζεται τα ερωτήματα που λαμβάνει από τον χρήστη, τα αναλύει και ανακτά σχετικά έγγραφα χρησιμοποιώντας το ανεστραμμένο ευρετήριο. Μπορείτε να χρησιμοποιήσετε απλά ερωτήματα βάσει λέξεων (όρων). Οι χρήστες θα πρέπει να μπορούν να αναζητούν έγγραφα χρησιμοποιώντας μία ή περισσότερες λέξεις. Το module θα λαμβάνει ερωτήματα χρηστών τα οποία τα γίνονται tokenized και θα εκτελεί απλές λειτουργίες Boolean (AND, OR και NOT).

Ορόσημο: Ένας λειτουργικός επεξεργαστής ερωτημάτων ικανός να χειρίζεται ερωτήματα Boolean.

β) Κατάταξη αποτελεσμάτων (Ranking): Εφαρμόστε έναν βασικό αλγόριθμο κατάταξης. Μπορείτε να ξεκινήσετε με έναν απλό αλγόριθμο κατάταξης TF-IDF (Term Frequency-Inverse Document Frequency) και αργότερα μπορείτε να συμπεριλάβετε πιο προηγμένες τεχνικές κατάταξης Υλοποιήστε πολλαπλούς (τουλάχιστον 3) αλγόριθμους ανάκτησης, όπως Boolean retrieval, Vector Space Model (VSM) και Probabilistic retrieval models (π.χ. Okapi BM25) για να ανακτήσετε σχετικές εργασίες με βάση τα ερωτήματα των χρηστών. Ο χρήστης θα μπορεί να επιλέγει τον αλγόριθμο ανάκτησης.

Ταξινομήστε και παρουσιάστε τα αποτελέσματα αναζήτησης σε φιλική προς το χρήστη μορφή.

Σημείωση: Όλοι οι αλγόριθμοι υπάρχουν υλοποιημένοι σε βιβλιοθήκες όπως στο Scikit-learn, okapi κλπ

Ορόσημο: Μια βελτιωμένη μηχανή αναζήτησης με ταξινομημένα αποτελέσματα βάσει ερωτημάτων χρηστών.

Βήμα 5. Αξιολόγηση συστήματος: Αξιολογήστε την απόδοση της μηχανής αναζήτησης χρησιμοποιώντας τυποποιημένες μετρικές αξιολόγησης όπως ακρίβεια, ανάκληση, F1-score και μέση ακρίβεια (MAP). Δημιουργήστε ένα σύνολο ερωτημάτων δοκιμής για να αξιολογήσετε την απόδοση του συστήματος. Εναλλακτικά χρησιμοποιείστε ένα υπάρχον σύνολο δεδομένων αξιολόγησης όπως π.χ. το **CISI dataset**

Σημείωση: Οι μετρικές αξιολόγησης υπάρχουν υλοποιημένες σε βιβλιοθήκες όπως στο Scikit-learn

Ορόσημο: Αναφορά σχετικά με την απόδοση του συστήματος, προσδιορίζοντας τα δυνατά και αδύνατα σημεία.

Βήμα 6. Αναφορά και τεκμηρίωση:

Γράψτε μια ολοκληρωμένη **αναφορά και τεκμηρίωση** που εξηγεί το σχεδιασμό, την υλοποίηση και την αξιολόγηση της μηχανής αναζήτησης (δεν χρειάζεται επεξήγηση κώδικα). Περιγράψτε αναλυτικά το/τα dataset που χρησιμοποιήσατε και ό,τι επιλογές κάνατε σε κάθε βήμα τις εργασίας χωρίς να συμπεριλάβετε καθόλου κώδικα. Αναφέρατε τις δυσκολίες που αντιμετωπίσατε και τις προτεινόμενες βελτιώσεις. Συμπεριλάβετε μελέτες περιπτώσεων από ερωτήματα χρηστών με

εικόνες screenshots, για επίδειξη της λειτουργικότητας της μηχανής αναζήτησης. Για οδηγό στη δομή της αναφοράς ακολουθείστε τη σειρά των βημάτων που ζητήθηκαν.
Ορόσημο: Τελική έκθεση αναφοράς

Παραδοτέα:

- **Τελική έκθεση αναφοράς:** Αναφορά σε μορφή **.pdf** με όνομα αρχείου **`Surname1 AM1-Surname2 AM2.pdf`** Στην **πρώτη** σελίδα της αναφοράς και κάτω από τα ονοματεπώνυμα, θα πρέπει να αναφέρεται το **link** με τον κώδικα όπως περιγράφεται παρακάτω.
- **Κώδικας και εκτελεσμένα παραδείγματα:** σύνδεσμος (link) με τον κώδικα και τα αποτελέσματα των επιμέρους βημάτων, με **παραδείγματα** και τα σχόλιά σας σε αρχείο **Jupiter Notebook** (ipynb). Το αρχείο θα πρέπει να έχει **εκτελεσμένα** τα επιμέρους βήματα και να είναι ανεβασμένο σε κάποιο repository π.χ. GitHub, Kaggle, Colab etc. Το link θα πρέπει να αναφέρεται στην **πρώτη** σελίδα της αναφοράς και κάτω από τα ονοματεπώνυμα. Το Jupyter Notebook θα πρέπει να έχει δομή αντίστοιχη με τα παραδείγματα που υπάρχουν στο eclass, δηλαδή να έχει επικεφαλίδες, κείμενο που θα περιγράφει τι κάνει το επόμενο block εντολών, και αποτελέσματα κάτω από κάθε block εντολών.

Κριτήρια αξιολόγησης εργασίας:
Η αξιολόγηση της εργασίας θα γίνει με βάση την ποιότητα και πληρότητα τόσο της αναφοράς όσο και της μηχανής αναζήτησης, την αποτελεσματικότητα του αλγορίθμου κατάταξης, τις μετρήσεις αξιολόγησης αλλά και την ικανότητα σας να εξηγήσετε τις έννοιες και τις αποφάσεις που λάβατε κατά τη διάρκεια του εργασίας υπερασπίζοντας τις σχεδιαστικές σας επιλογές και την απόδοση στην αξιολόγηση του συστήματος ανάκτησης.

Εξέταση:
Η εξέταση για το εργαστηριακό μέρος θα γίνει μαζί με την εξέταση του θεωρητικού, στην εξεταστική του Φεβρουαρίου. Ο βαθμός από την εργασία θα προσμετρήσει **μόνο** εφόσον απαντηθούν οι ερωτήσεις που θα αντιστοιχούν στο εργαστηριακό μέρος.

Την εργασία μπορεί να την αναλαμβάνει ομάδα μέχρι 2 άτομα από οποιοδήποτε τμήμα

Ημ/νία κατάθεσης (από το **ένα μόνο μέλος** της ομάδας) στο eclass: **14/01/2025**

χωρίς δυνατότητα εκπρόθεσμης υποβολής

Απορίες θα λύνονται μόνο κατά τη διάρκεια του εργαστηριακού μαθήματος και όχι με e-mail