Ανάκτηση Πληροφορίας - 2η Φάση

Scientific Article Search Engine

4651: Γιακουμάκη Ελένη , 4680: Καϊλίδης Κύριλλος

https://github.com/elenigki/Sci-ArticleSearch

Εισαγωγή:

Ο στόχος της εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός συστήματος αναζήτησης πληροφορίας (λέξεων - φράσεων) σε επιστημονικά άρθρα. Η μηχανή αναζήτησης αυτή, χρησιμοποιεί την βιβλιοθήκη ανοικτού κώδικα, Lucene.

Συλλογή άρθρων:

Η συλλογή που χρησιμοποιήθηκε, προέρχεται από την ιστοσελίδα Kaggle, και πιο συγκεκριμένα:

https://www.kaggle.com/datasets/rowhitswami/nips-papers-1987-2019-updated/data?select=papers.csv

Η συλλογή περιέχει 9.406 επιστημονικά άρθρα. Κάθε ένα από τα άρθρα περιέχουν τα πεδία: *source_id, year, title, abstract και full_text.*

Ανάλυση κειμένου και κατασκευή ευρετηρίου:

Για την ανάλυση του κειμένου και την κατασκευή του ευρετηρίου χρησιμοποιήθηκαν διάφορες λειτουργίες που προσφέρει η Lucene. Συγκεκριμένα, χρησιμοποιήθηκε ο ***StandardAnalyzer***, ο οποίος παρέχει ένα πλήθος λειτουργιών για την επεξεργασία και την κανονικοποίηση του κειμένου, πριν αυτό εισαχθεί στο ευρετήριο.

Οι λειτουργίες του ***StandardAnalyzer*** που χρησιμοποιήθηκαν είναι:

(I) *Ανάλυση κειμένου σε λέξεις* ***(Tokenization)***: το κείμενο διαχωρίζεται σε μεμονωμένες λέξεις (tokens), για να μπορούν να αναγνωριστούν και να αποθηκευτούν στο ευρετήριο.

(II) *Μετατροπή σε πεζά γράμματα* ***(Lowercasing)***: τα γράμματα όλων των λέξεων μετατρέπονται σε πεζά, ώστε η αναζήτηση να μην επηρεαστεί από την διάκριση πεζών-κεφαλαίων γραμμάτων.

(III) *Αφαίρεση κοινών λέξεων* ***(Stop Word Removal)***: οι κοινές λέξεις (όπως οι "the", "is", "at" κλπ) αφαιρούνται, καθώς δεν έχουν ιδιαίτερη σημασία στην αναζήτηση, και έτσι μειώνεται το μέγεθος του ευρετηρίου και βελτιώνεται η απόδοση της αναζήτησης.

(IV) *Διαχείριση σημείων στίξης* ***(Punctuation Handling)***: τα σημεία στίξης και οι ειδικοί χαρακτήρες αγνοούνται ή επεξεργάζονται κατάλληλα για να μην επηρεάσουν την αναζήτηση.

(V) *Κανονικοποίηση κειμένου* ***(Text Normalization)***: εκτελείται βασική κανονικοποίηση του κειμένου για να εξασφαλιστεί η συνέπεια των δεδομένων.

Για την κατασκευή του ευρετηρίου, τα δεδομένα από τα άρθρα επεξεργάζονται και εισάγονται ως *documents*. Κάθε άρθρο αναπαρίσταται με τα πεδία: ***sourceId***, ***year***, ***title***, ***abstractText*** και ***fullText***. Αυτά τα πεδία ευρετηριάζονται και αποθηκεύονται με τον ακόλουθο τρόπο:

- ***sourceId:*** αποθηκεύεται και ευρετηριάζεται ως ***StringField***.
- ***year:*** αποθηκεύεται ως ***StoredField*** και ευρετηριάζεται ως ***IntPoint*** (για ερωτήματα εύρους) και ως ***NumericDocValuesField*** (για ταξινόμηση).
- ***title*** , ***abstractText*** *και* ***fullText:*** αποθηκεύεται και ευρετηριάζεται ως ***TextField*** (για να είναι αναζητήσιμα και ανακτήσιμα).
-

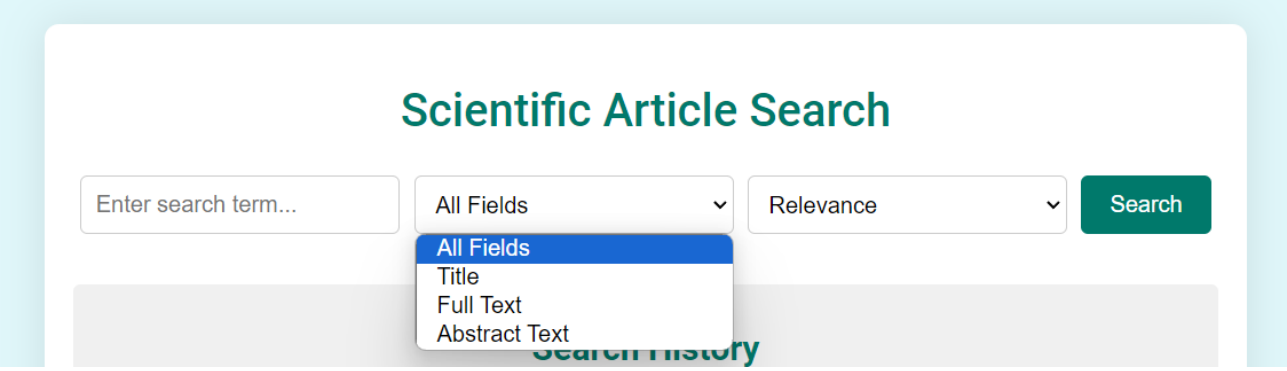
(αναφερόμενος κώδικας από LuceneIndexer.class)

```
20    public LuceneIndexer(String indexPath) throws IOException {
21        this.indexDir = FSDirectory.open(Paths.get(indexPath));
22        this.analyzer = new StandardAnalyzer();
23        this.config = new IndexWriterConfig(analyzer);
24        this.writer = new IndexWriter(indexDir, config);
25    }
26
27    public void indexArticle(ArticleData article) throws IOException {
28        Document doc = new Document();
29        doc.add(new StringField("sourceId", String.valueOf(article.getSourceId()), Field.Store.YES));
30        doc.add(new IntPoint("year", article.getYear())); // Indexed for range queries
31        doc.add(new StoredField("year", article.getYear())); // Stored for retrieval
32        doc.add(new NumericDocValuesField("year", article.getYear())); // Indexed for sorting
33        doc.add(new TextField("title", article.getTitle(), Field.Store.YES)); // Indexed and stored
34        doc.add(new TextField("abstractText", article.getAbstractText(), Field.Store.YES)); // Indexed and stored
35        doc.add(new TextField("fullText", article.getFullText(), Field.Store.YES)); // Indexed and stored
36        writer.addDocument(doc);
37    }
```

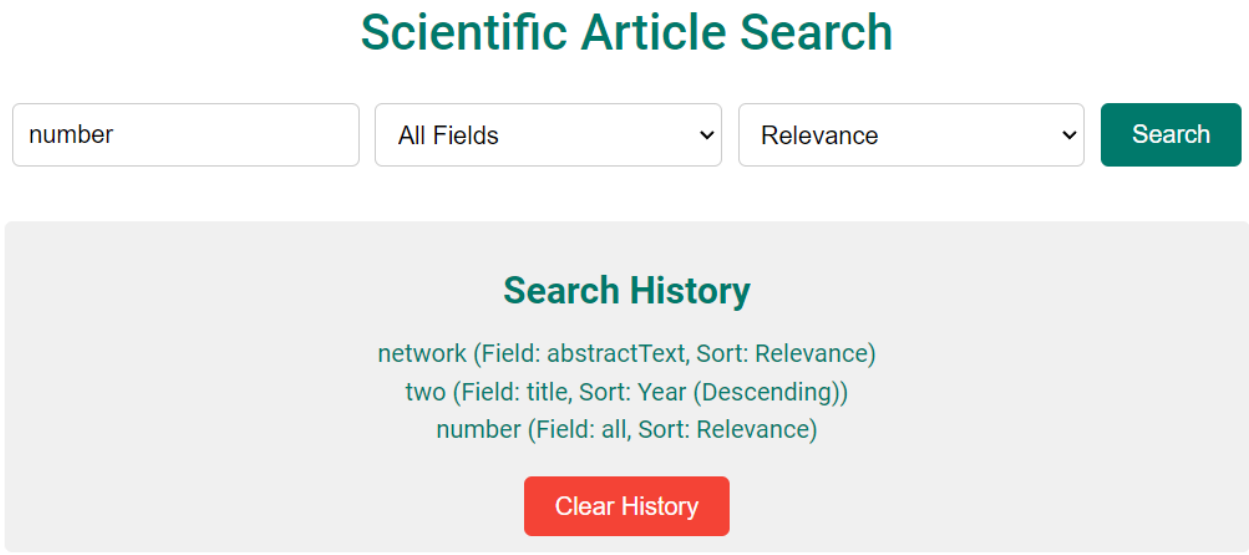
Αναζήτηση:

Η μηχανή αυτή, υποστηρίζει πολλά είδη αναζήτησης. Μπορεί να δεχτεί ως όρο αναζήτησης μια λέξη, τμήμα λέξης ή και ολόκληρη φράση.

Η αναζήτηση επίσης μπορεί να γίνει σε επιλεγμένα πεδία του άρθρου, όπως στον τίτλο(Title), την περίληψη(Abstract Text), το κυρίως κείμενο(Full Text) ή και όλα αυτά μαζί(All Fields).

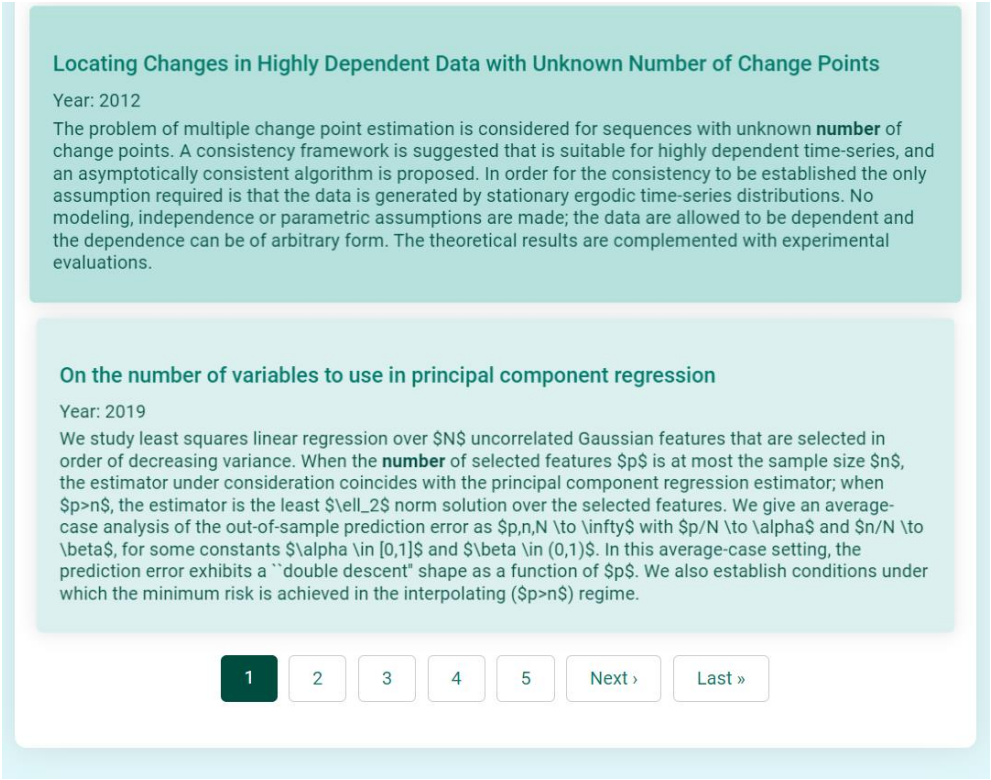


Η μηχανή αναζήτησης περιέχει και αρχείο ιστορίας αναζήτησης (*Search History*), στο οποίο διατηρείται ο όρος που αναζητήθηκε, τα πεδία αρχείου στα οποία αναζητήθηκε και η διάταξη που ζητήθηκε (αναλύεται στην "Παρουσίαση Αποτελεσμάτων"). Ο χρήστης, πατώντας στον επιθυμητό όρο που θέλει από την ιστορία αναζήτησης, οδηγείται αυτόματα στα αποτελέσματα που είχε λάβει και στο παρελθόν. Επιπλέον, υπάρχει και η δυνατότητα διαγραφής ιστορικού *(Clear History)* με την οποία όλες οι προηγούμενες αναζητήσεις διαγράφονται πλέον οριστικά.

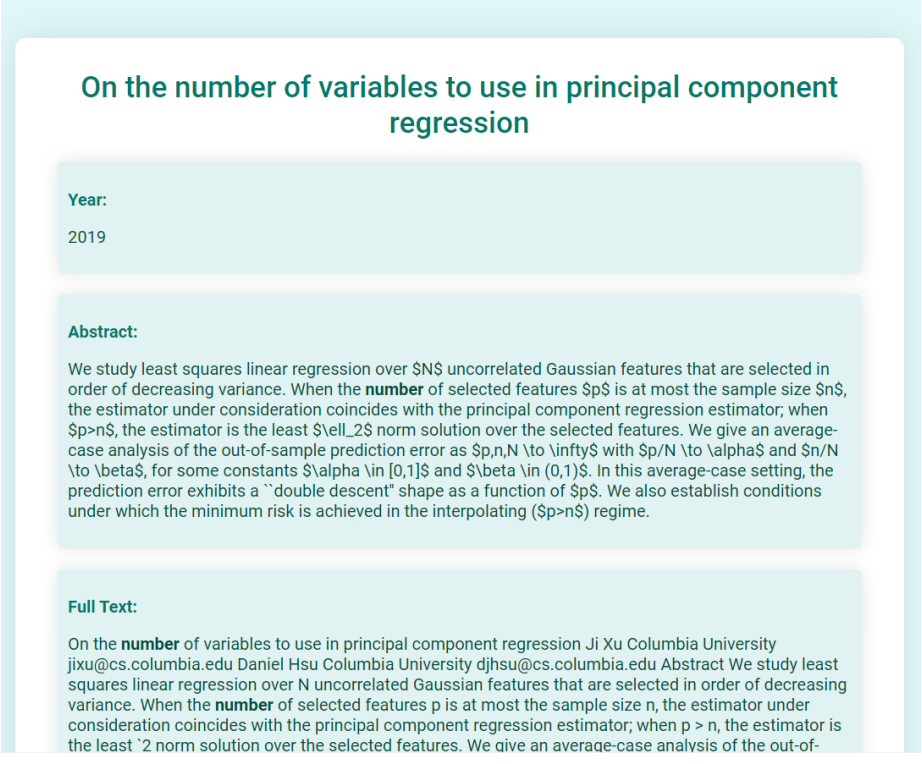


Παρουσίαση Αποτελεσμάτων:

Τα αποτελέσματα της αναζήτησης παρουσιάζονται ανά 10 και υπάρχει η δυνατότητα επιλογής και άλλων σελίδων (οι οποίες παρουσιάζουν επίσης 10 αποτελέσματα η κάθε μία). Σε κάθε άρθρο που εμφανίζεται, υπάρχει ο τίτλος του, η χρονιά την οποία γράφτηκε και από κάτω η περίληψη του (abstract) αν αυτή η υπάρχει, στην οποία φαίνεται τονισμένος ο όρος τον οποίο αναζητήσαμε.



Πατώντας το άρθρο, μεταφερόμαστε σε μια άλλη σελίδα, η οποία περιέχει όλες τις πληροφορίες του άρθρου (τίτλος, χρονολογία, περίληψη και το ίδιο το άρθρο(fullText)). Ο όρος που αναζητήσαμε είναι επίσης τονισμένος σε όλα τα πεδία που τον συναντάμε.



Υπάρχουν δύο επιλογές διάταξης των αποτελεσμάτων. Ο ένας είναι η σχετικότητα (*Relevance)* του όρου που αναζητήσαμε, οπότε τα πρώτα αποτελέσματα που θα πάρουμε θα είναι και αυτά με τις περισσότερες αναφορές και το καλύτερο σκορ της μηχανής αναζήτησης μας. Η σχετικότητα υπολογίζεται αυτόματα από τις βιβλιοθήκες της Lucene. Ο δεύτερος τρόπος είναι η διάταξη των αποτελεσμάτων χρονολογικά (*Year-Descending*), δηλαδή από τα πιο πρόσφατα ως τα παλαιότερα άρθρα, σε περίπτωση που ο χρήστης θέλει να βρει τις νεότερες πληροφορίες πάνω σε ένα θέμα.