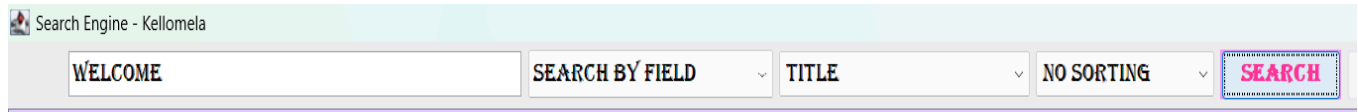


Μηχανή αναζήτησης επιστημονικών άρθρων

Github link

<https://github.com/militsavdr/Information-Retrieval.git>



Επειδή το μέγεθος του αρχείου μας(1.3GB) ξεπερνάει τα 25 MB(ακόμα και σε μορφή .zip)παρόλο που αγοράσαμε και επιπλέον χώρο στο github δυστυχώς δεν ανεβαίνει το project εκεί . Σας παρέχουμε το link στο we transfer το οποίο έχει και συγκεκριμένο χρονικό όριο μιας εβδομάδας. Παρόλα αυτά στο github παρέχουμε τις κλάσεις που δημιουργήσαμε .

link we transfer: <https://we.tl/t-GD3xOqju8w>

Συλλογή εγγράφων - Corpus

Χρησιμοποιούμε επιστημονικά άρθρα από την παρακάτω συλλογή από το Kaggle:

<https://www.kaggle.com/datasets/rowhitswami/nips-papers-1987-2019-update/d/data?select=papers.csv>

Η συλλογή αποτελείται από δύο αρχεία της μορφής csv τα οποία περιέχουν πληροφορίες για επιστημονικά άρθρα καθώς και για τους συγγραφείς τους. Πιο συγκεκριμένα, το πρώτο αρχείο, με την ονομασία papers.csv, περιέχει πληροφορίες σχετικά με τα επιστημονικά άρθρα:

- source_id: μοναδικό αναγνωριστικό πηγής
- year: χρονολογία δημοσίευσης
- title: τίτλος
- abstarct: περίληψη
- full_text: πλήρες κείμενο

ενώ το δεύτερο αρχείο authors.csv περιέχει πληροφορίες σχετικά με τους συγγραφείς :

- source_id: μοναδικό αναγνωριστικό
- first_name: όνομα συγγραφέα
- last_name: επίθετο συγγραφέα
- institution: ινστιτούτο στο οποίο ανήκει ο συγγραφέας

Για να συσχετίσουμε έναν συγγραφέα με τα δημοσιευμένα του άρθρα, χρησιμοποιούμε το κοινό πεδίο `source_id`. Κάθε συγγραφέας έχει ένα μοναδικό `source_id` που τον συσχετίζει με τα άρθρα που έχει γράψει.

Με βάση τις απαιτήσεις του συστήματος, χρησιμοποιούμε όλα τα πεδία για την υλοποίηση των διαφορετικών μορφών αναζήτησης.

Εισαγωγή

Ο κύριος στόχος του συστήματος αναζήτησης είναι η διευκόλυνση της πρόσβασης σε επιστημονικά άρθρα και στους συγγραφείς τους, μέσω ενός αποτελεσματικού και εύχρηστου περιβάλλοντος αναζήτησης. Η λειτουργικότητα του συστήματός μας, περιλαμβάνει αναζήτηση με βάση λέξεις-κλειδιά, κάποιο συγκεκριμένο πεδίο αλλά και συνώνυμα λέξεων. Σκοπός είναι να επικεντρώνεται ο σχεδιασμός στην ακρίβεια, την απλότητα και την ολοκληρωμένη κάλυψη των αποτελεσμάτων. Για την υλοποίηση, χρησιμοποιούμε την βιβλιοθήκη Lucene, η οποία μας παρέχει τα απαραίτητα εργαλεία και λειτουργίες.

Ανάλυση κειμένου και κατασκευή ευρετηρίου

Προεπεξεργασία

Απαραίτητο κρίνουμε τα άρθρα και οι πληροφορίες των συγγραφέων να υποστούν ενός είδους προεπεξεργασίας για την αφαίρεση της περιττής πληροφορίας και την εξασφάλιση ομοιομορφίας στην δομή των δεδομένων. Συγκεκριμένα χρησιμοποιούμε το API της Lucene

org.apache.lucene.analysis για την επεξεργασία των κειμένων, μέσω του οποίου επιλέγουμε ως αναλυτή για την μετατροπή των λέξεων σε tokens, τον **Standard Analyzer** καθώς και τον **Synonym Analyzer**. Ο Standard Analyzer είναι υπεύθυνος για την αφαίρεση των συχνών λέξεων (stop words), την αφαίρεση σημείων στίξης, καθώς και την μετατροπή όλων των γραμμάτων σε lowercase. Ο Synonym Analyzer είναι ένα εργαλείο που χρησιμοποιείται σε προγράμματα ανάλυσης κειμένου για να αναγνωρίζει και να διαχειρίζεται συνώνυμα λέξεις.

Μονάδα εγγράφου και πεδία

Η μονάδα εγγράφου αποτελείται από όλη την συλλογή μας καθώς συγκεντρώνει όλες τις σχετικές πληροφορίες για ένα άρθρο και τον συγγραφέα του, οργανώνοντάς τις σε ένα ενιαίο σύνολο δεδομένων.

Ευρετήρια

Η δημιουργία των ευρετηρίων βασίζεται σε κείμενο για τους τίτλους, τις περιλήψεις και τα πλήρη κείμενα των επιστημονικών άρθρων, αλλά και στα ονόματα, επώνυμα και ινστιτούτα συγγραφέων με την χρήση του API **org.apache.lucene.index** το οποίο παρέχει τις κλάσεις που αφορούν την δημιουργία και την διαχείριση των ευρετηρίων.

Για κάθε άρθρο και πληροφορία συγγραφέα, δημιουργούμε ένα αντικείμενο Document χρησιμοποιώντας το API **org.apache.lucene.document**. Αυτό το έγγραφο περιέχει πολλά πεδία (fields) που αντιπροσωπεύουν τα διάφορα μέρη του άρθρου ή τις πληροφορίες του συγγραφέα.

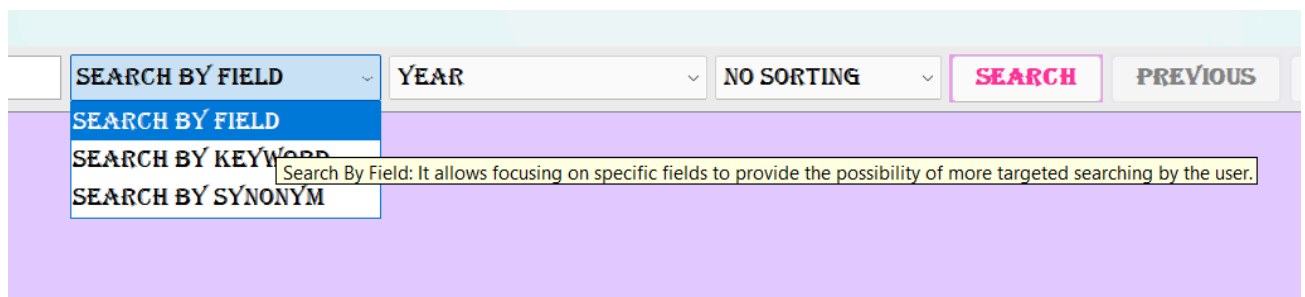
Αρχικά, η χρήση **ανεστραμμένου ευρετηρίου** είναι απαραίτητη για την αναζήτηση συγκεκριμένων λέξεων σε συγκεκριμένα πεδία, όπως ο τίτλος, η περίληψη κ.λπ. Αυτό επιτυγχάνεται με τον συσχετισμό κάθε λέξης με το αντίστοιχο έγγραφο που την περιλαμβάνει. Με αυτόν τον τρόπο, μπορούμε να επιτύχουμε πιο γρήγορη και αποδοτική αναζήτηση σε συγκεκριμένα πεδία των εγγράφων.

Αναζήτηση

Χρησιμοποιούμε το **org.apache.lucene.search** καθώς παρέχει τις κλάσεις που αφορούν την εκτέλεση των αναζητήσεων στο ευρετήριο και την επεξεργασία αποτελεσμάτων. Το σύστημα υποστηρίζει 3 βασικά είδη αναζήτησης εγγράφων:

(α) Αναζήτηση Πεδίου (Fielded Search)

Επιτρέπει την επικέντρωση σε συγκεκριμένα πεδία προκειμένου να προσφέρει τη δυνατότητα πιο στοχευμένης αναζήτησης από τον χρήστη (π.χ. αναζήτηση με βάση τη χρονολογία δημοσίευσης ενός άρθρου (πεδίο: year)). Αυτό επιτρέπει τη διευκόλυνση της εύρεσης συγκεκριμένων πληροφοριών και την προβολή των αποτελεσμάτων σε σχέση με τα συγκεκριμένα κριτήρια που επιλέγει ο χρήστης.

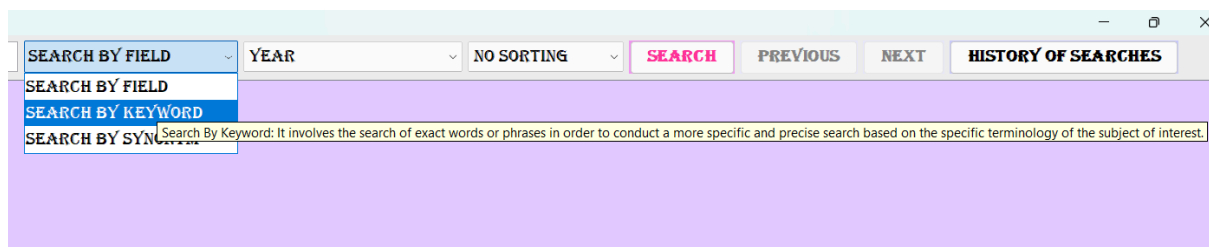


Παράδειγμα αναζήτησης :

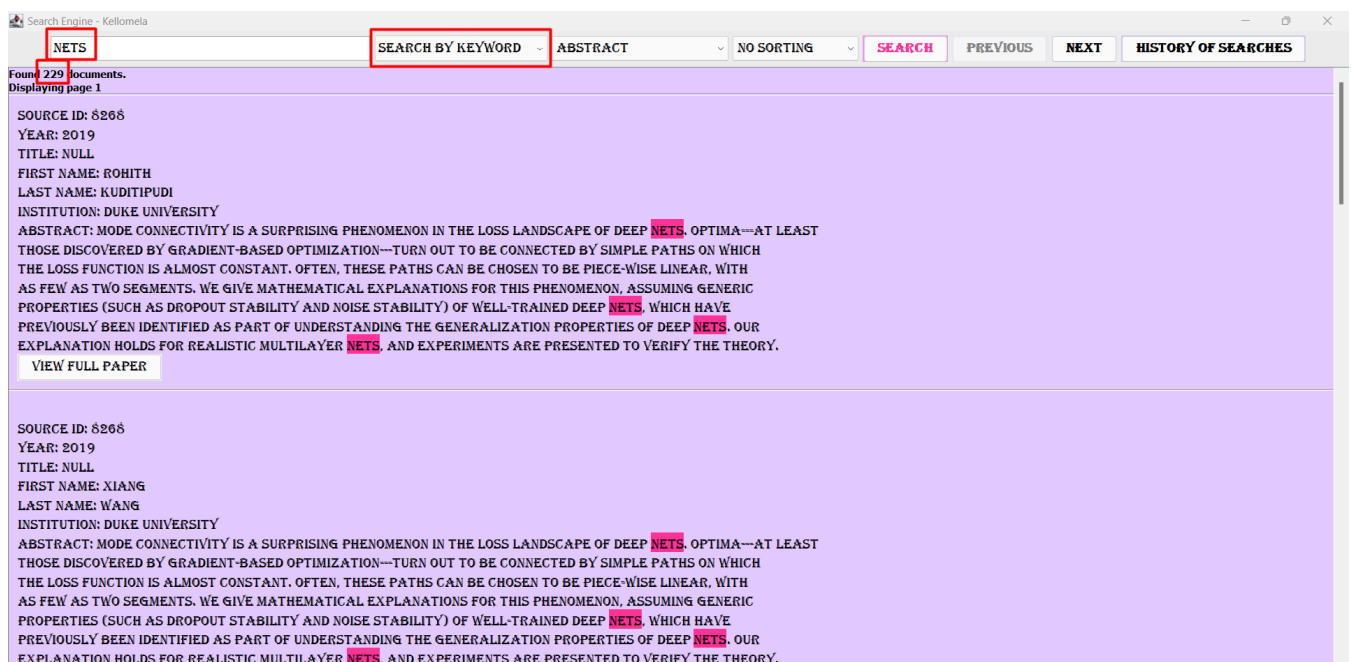


(β) Αναζήτηση με λέξεις-κλειδιά (Keyword Search)

Αφορά την αναζήτηση αυτούσιων λέξεων ή φράσεων προκειμένου να γίνει μία πιο ειδική και ακριβής αναζήτηση με βάση συγκεκριμένη ορολογία του αντικειμένου ενδιαφέροντος. Η αναζήτηση αυτή μπορεί να γίνει σε όλα τα πεδία εξασφαλίζοντας ότι κανένα σημαντικό στοιχείο δεν παραβλέπεται και αυξάνοντας σημαντικά τη πιθανότητα ο χρήστης να βρει ακριβώς αυτό που ψάχνει.



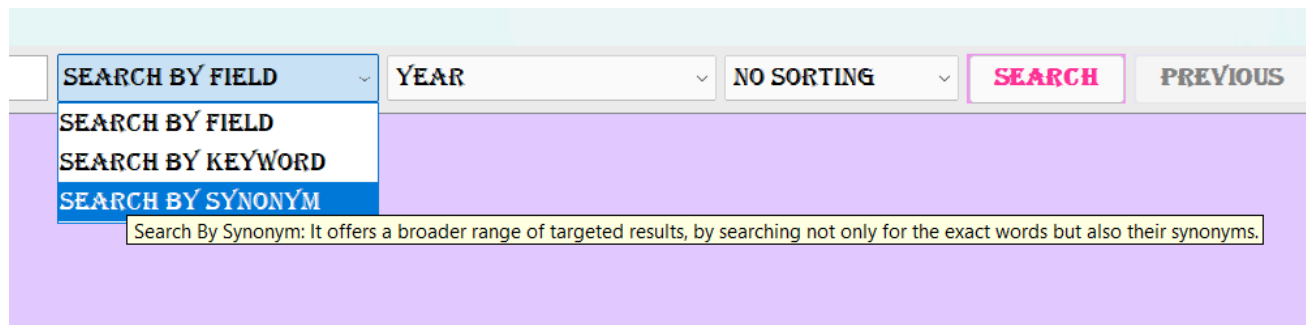
Παράδειγμα αναζήτησης :



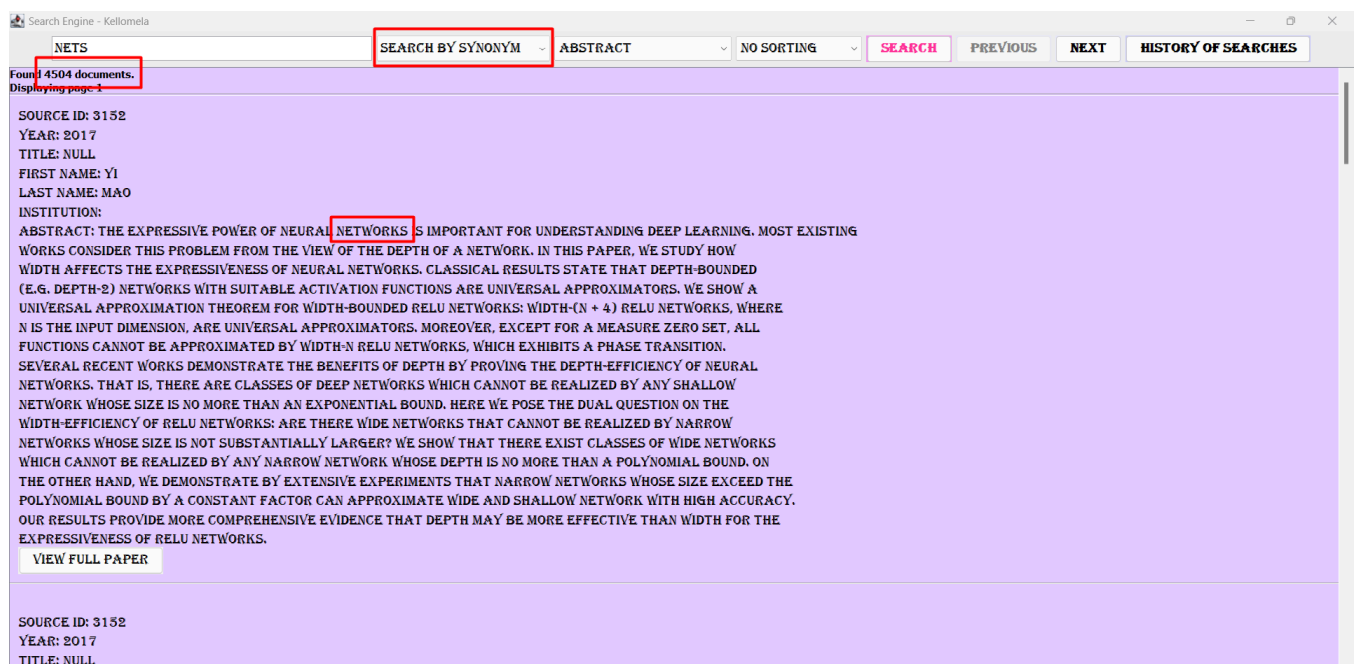
(γ) Αναζήτηση με Συνώνυμα (Synonym Search)

Προσφέρει μεγαλύτερο εύρος στοχευμένων αποτελεσμάτων και την κάλυψη περιπτώσεων διαφορετικής χρήσης ορολογίας ή τρόπου έκφρασης που χρησιμοποιούνται στα επιστημονικά άρθρα. Αυτή η αναζήτηση αφορά πεδία κειμένων (περίληψη και πλήρες κείμενο). Επιτρέπει την εύρεση εγγράφων που ίσως

δεν είχαν εντοπιστεί με ένα πιο περιοριστικό ερώτημα αλλά και βελτιώνει σημαντικά την εμπειρία του χρήστη καθώς καλύπτει και αυτούς που δεν είναι εξοικειωμένοι με την ακριβή επιστημονική ορολογία. Έτσι, το σύστημα γίνεται προσιτό και εύχρηστο για όλους τους χρήστες, επιτρέποντας άτομα εκτός της επιστημονικής κοινότητας, αλλά και σε ερευνητές από άλλους επιστημονικούς τομείς, να χρησιμοποιούν τη μηχανή αναζήτησης με μεγαλύτερη ευκολία και αποτελεσματικότητα. Για την επίτευξη αυτής της αναζήτησης χρησιμοποιούμε το ευρετήριο **SynonymMap** για τον ορισμό των συνωνύμων μέσω του API **org.apache.lucene.analysis.synonym**.

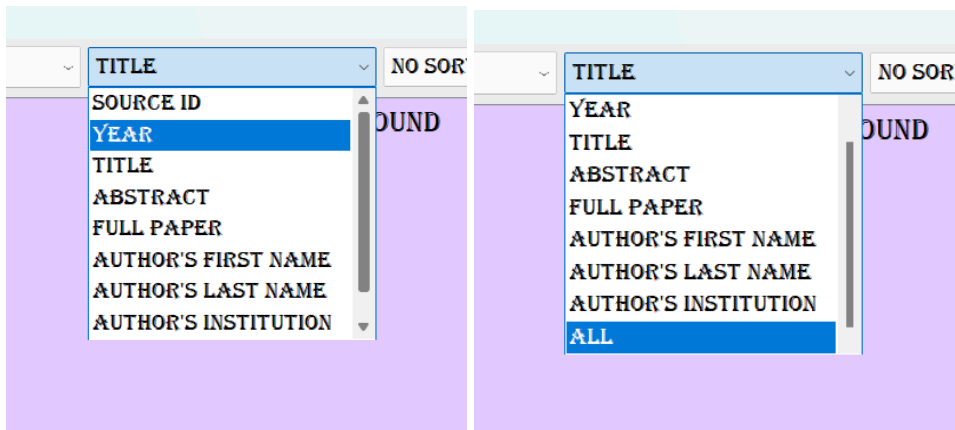


Παράδειγμα αναζήτησης :



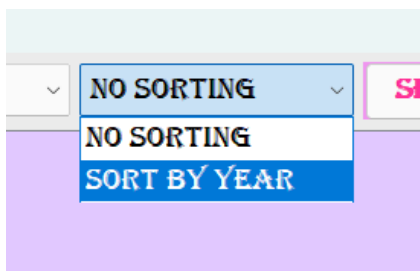
Επιλογή πεδίου αναζήτησης

Οποιοδήποτε είδος αναζήτησης μπορεί να γίνει σε οποιοδήποτε πεδίο επιλέξει ο χρήστης, έχοντας επιπλέον τη δυνατότητα να εκτελεστεί η αναζήτηση σε όλα τα πεδία ταυτόχρονα επιλέγοντας το πεδίο 'ALL'.

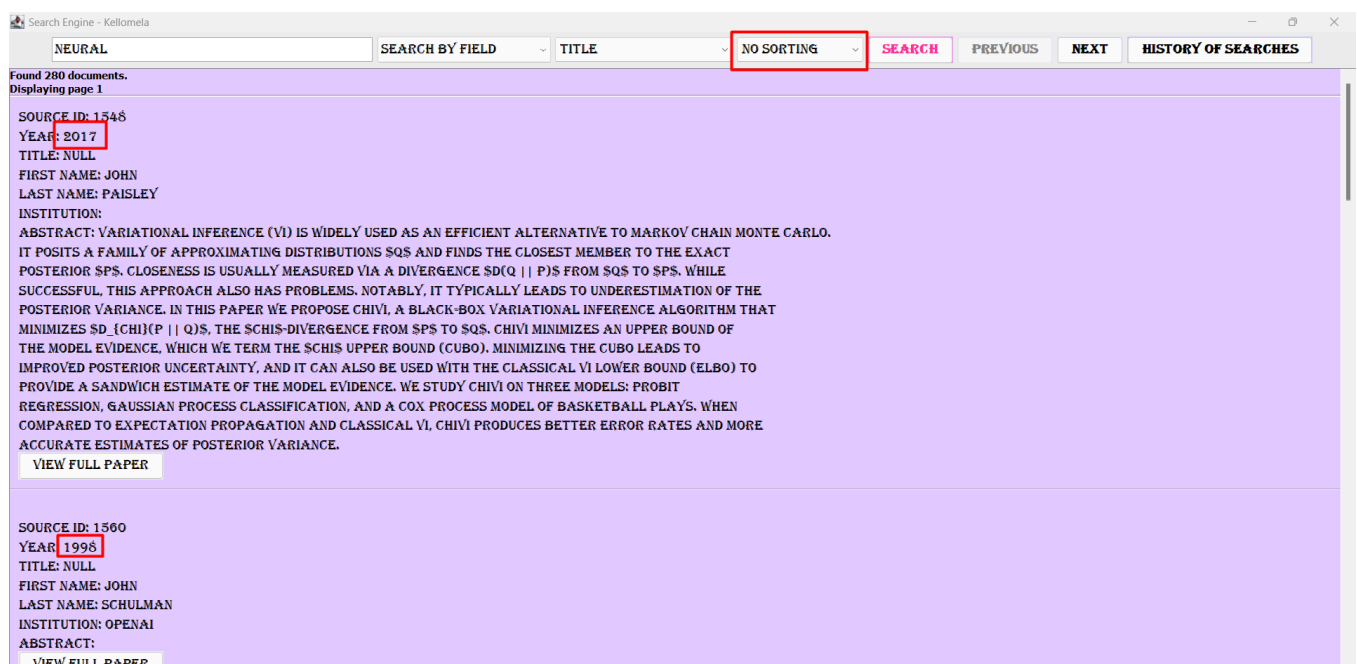


Ταξινόμηση αποτελεσμάτων

Παρέχεται η δυνατότητα αναδιάταξης των αποτελεσμάτων με βάση τη χρονολογία που δημοσιεύτηκε το κάθε άρθρο. Με αυτόν τον τρόπο, η εφαρμογή παρέχει μια ευέλικτη και χρήσιμη διεπαφή για τους χρήστες που αναζητούν πληροφορίες στο σύστημά μας.



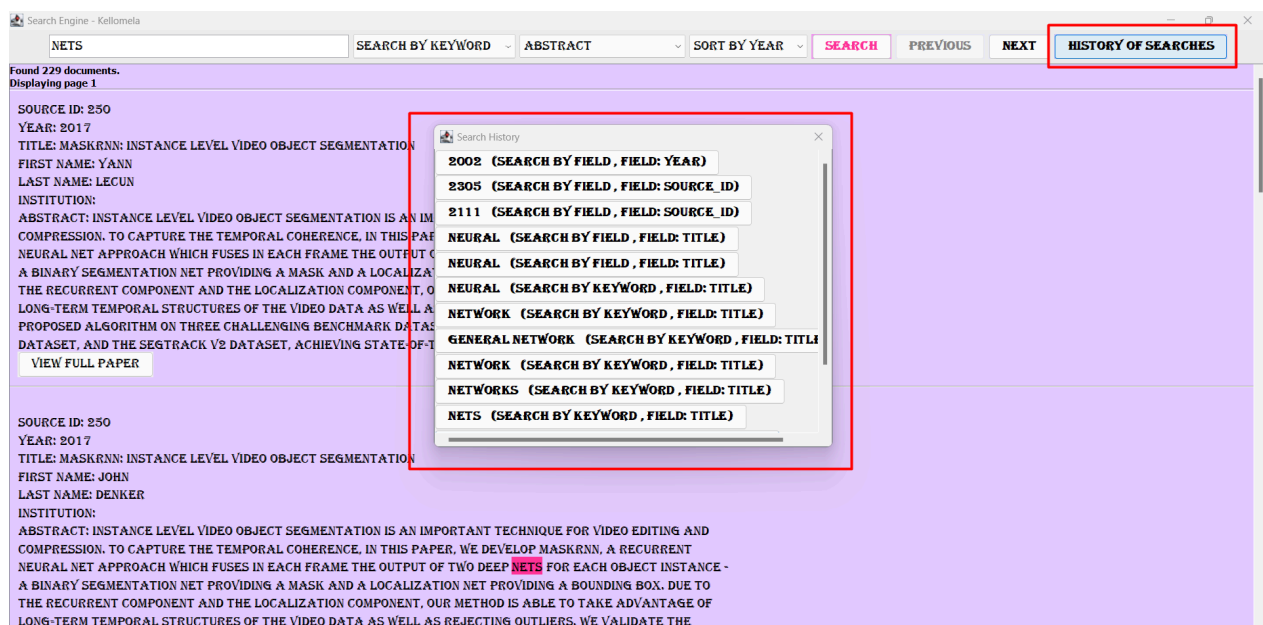
Παραδείγματα των λειτουργιών ταξινόμησης (No Sorting, Sort By Year):





Ιστορικό Αναζήτησης

Σημαντική λειτουργία του συστήματος αποτελεί και η διατήρηση του ιστορικού αναζητήσεων του κάθε χρήστη. Επιλέγοντας το κουμπί “History Of Searches”, ο χρήστης μπορεί να δει με απλό και καθαρό τρόπο την αναζήτηση που έκανε καθώς και τον τύπο αναζήτησης και πεδίου που επέλεξε κατά εκείνη την αναζήτηση. Ακόμα, πατώντας πάνω σε μία προηγούμενη αναζήτηση που έκανε ο χρήστης κατευθύνεται άμεσα στα αποτελέσματα της αναζήτησης εκείνης, χωρίς να επιβαρυνθεί παραπάνω το σύστημα αφού δεν εκτελεί ξανά την ίδια αναζήτηση αλλά την ανακτά από το ιστορικό.

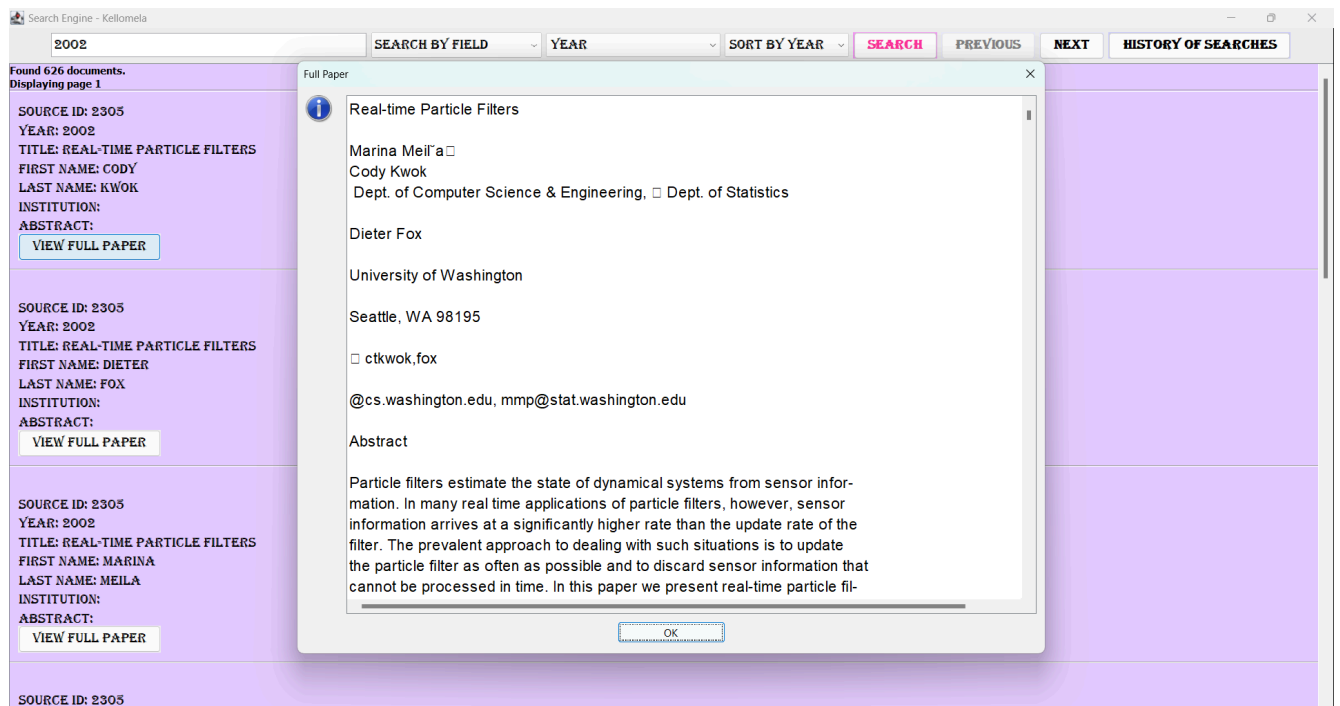


Παρουσίαση Αποτελεσμάτων

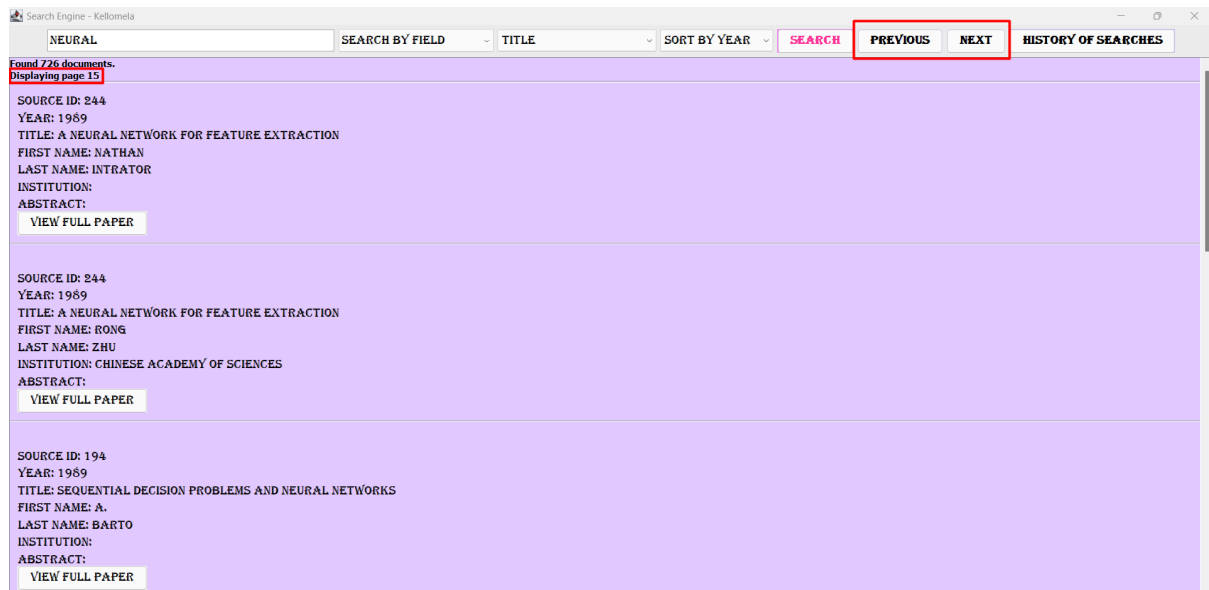
Το σύστημά μας παρουσιάζει τα αποτελέσματα της αναζήτησης διατεταγμένα με βάση τη συνάφειά τους με το ερώτημα του χρήστη. Χρησιμοποιώντας το **Java Swing** ως framework, διαθέτουμε ένα πλήρες σύνολο βιβλιοθηκών και εργαλείων για τη δημιουργία γραφικών διεπαφών χρήστη (GUI).



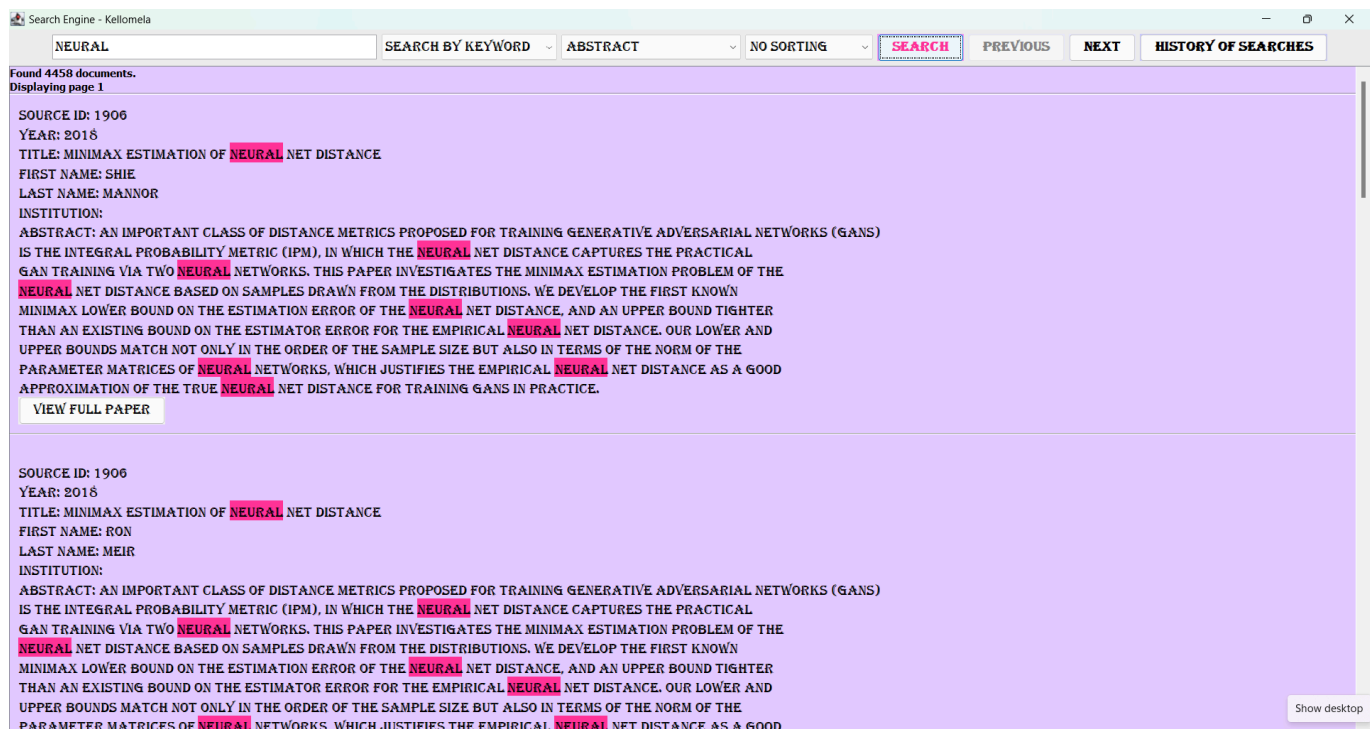
Πατώντας ο χρήστης το κουμπί “View Full Paper” από το άρθρο που θέλει να διαβάσει ανοίγει ένα νέο παράθυρο με το πλήρες κείμενο του άρθρου



Τα αποτελέσματα παρουσιάζονται σε ομάδες των 10, προσφέροντας τη δυνατότητα στον χρήστη να προβεί στην περιήγηση μεταξύ των σελίδων με το πάτημα των αντίστοιχων κουμπιών (next και previous) καθώς και εμφανίζονται στοιχεία για τη σελίδα στην αρχή και στο τέλος της σελίδας όπου φαίνονται και οι συνολικές σελίδες.



Επιπλέον, οι λέξεις-κλειδιά (κατά την εκτέλεση της αναζήτησης με λέξεις-κλειδιά) εμφανίζονται στα αποτελέσματα τονισμένες για να επισημανθεί η συνάφειά τους με το ερώτημα του χρήστη στα πεδία που εμφανίζονται



Οδηγίες εκτέλεσης

Προκειμένου να εκτελεστεί ορθά η μηχανή αναζήτησής μας , θα πρέπει να προστεθούν τα ακόλουθα jar files της βιβλιοθήκης Lucene στο build Path:

1. analysis/common
2. core
3. demo
4. queryparser

Αρχικά θα χρειαστεί να εκτελεστεί η **CsvLoader.java** η οποία είναι υπεύθυνη για το διάβασμα των csv αρχείων, **Indexer.java** η οποία είναι υπεύθυνη για την δημιουργία του ευρετηρίου και στην συνέχεια η **SearchGUI.java** η οποία αποτελεί το γραφικό περιβάλλον αλληλεπίδρασης με τον χρήστη και σε συνδυασμό με την Searcher.java πραγματοποιείται η ζητούμενη αναζήτηση.

Σε περίπτωση που έχει δημιουργηθεί ήδη το ευρετήριο(δηλαδή να έχει εκτελεστεί η Indexer.java μία φορά) εκτελείται απευθείας την **SearchGUI.java**.

Τέλος, για την διεύρυνση αποτελεσμάτων αναζήτησης , υπάρχει η δυνατότητα επιπλέον προσθήκης συνώνυμων λέξεων στο αρχείο synonyms.txt, διαχωρισμένες με κόμμα.