Προγραμματιστική Εργασία

Φάση 1 Baseline – Κλασική ανάκτηση

Συστήματα Ανάκτησης Πληροφοριών Ακαδημαϊκό

Έτος 2025-2026

ΦΑΤΣΕΑ ΑΝΘΙΠΠΗ 3190209

1. Εισαγωγή

Στο πλαίσιο της παρούσας εργασίας, αναπτύχθηκε, βελτιστοποιήθηκε και αξιολογήθηκε ένα σύστημα ανάκτησης πληροφορίας με χρήση της μηχανής **Elasticsearch**. Το αντικείμενο μελέτης αφορά μια συλλογή εγγράφων από ερευνητικά έργα της Ευρωπαϊκής Ένωσης και ένα σύνολο ερωτημάτων φυσικής γλώσσας.

Κύριος στόχος της υλοποίησης ήταν η επίτευξη υψηλής **Ακρίβειας** *(Precision)* στα πρώτα αποτελέσματα, διατηρώντας παράλληλα υψηλή **Ανάκληση** *(Recall)* στο σύνολο, με ιδιαίτερη έμφαση στη μετρική **Mean Average Precision** *(MAP)*. Η μεθοδολογία βασίστηκε στον πιθανοτικό αλγόριθμο **BM25**, ενισχυμένο με προχωρημένες τεχνικές **Επεξεργασίας Φυσικής Γλώσσας** *(NLP)* μέσω της γλώσσας προγραμματισμού **Python**, ενώ η αξιολόγηση των αποτελεσμάτων πραγματοποιήθηκε με το πρότυπο εργαλείο **trec_eval**.

Για τη μεγιστοποίηση της απόδοσης, διεξήχθη εκτενής πειραματισμός με ποικίλες τεχνικές ανάλυσης και στρατηγικές αναζήτησης. Συγκεκριμένα, εξετάστηκαν η επίδραση της **ενίσχυσης πεδίων** *(Boosting)*, η χρήση **διγραμμάτων** *(Bigrams)*, η εφαρμογή **Λημματοποίησης** *(Lemmatization)* και **ριζικής ετυμολογίας** *(Stemming)*, καθώς και διαφορετικά **μοντέλα ερωτημάτων**. Παράλληλα, πραγματοποιήθηκε συστηματικός έλεγχος για τον εντοπισμό των πιο αποδοτικών τιμών *b* και *k1* του αλγορίθμου BM25, ώστε να προσαρμοστεί η κατάταξη στα χαρακτηριστικά της συλλογής. Επιπλέον, υλοποιήθηκε ο διαχωρισμός και η εξαγωγή των τίτλων από το σώμα των κειμένων.

Σημαντικό μέρος της εργασίας αποτέλεσε η ανάπτυξη εξειδικευμένων, custom μηχανισμών για τη στατιστική ανάλυση του corpus. Μέσω αυτών, εντοπίστηκαν και αφαιρέθηκαν δυναμικά λέξεις και φράσεις υψηλής συχνότητας ***(Stopwords & Boilerplate)*** που αποτελούσαν θόρυβο, βελτιώνοντας δραστικά την ποιότητα του ευρετηρίου και την ταχύτητα ανάκτησης.

2. Μεθοδολογία & Προεπεξεργασία Δεδομένων

Η ποιότητα των αποτελεσμάτων σε ένα σύστημα ανάκτησης εξαρτάται άμεσα από την ποιότητα των δεδομένων εισόδου. Για τον λόγο αυτό, δεν εφαρμόστηκε μια τυπική διαδικασία καθαρισμού, αλλά αναπτύχθηκε ένα εξειδικευμένο pipeline προεπεξεργασίας, προσαρμοσμένο στα ιδιαιτέρα χαρακτηριστικά της συλλογής εγγράφων.

Έγινε εγκατάσταση των απαραίτητων βιβλιοθηκών. **Pandas** για τη διαχείριση και οργάνωση των δεδομένων, **NLTK** (*Natural Language Toolkit*) για την προεπεξεργασία και τον καθαρισμό του κειμένου, **Elasticsearch** *(έκδοση 9.1.5)* για τη σύνδεση και την επικοινωνία με τη μηχανή αναζήτησης, η **Matplotlib** για την οπτικοποίηση των αποτελεσμάτων μέσω γραφημάτων, και η **Scikit-learn** για τον υπολογισμό στατιστικών μετρικών αξιολόγησης. Επιπλέον έγινε λήψη των **datasets stopwords**, **punkt**, **wordnet** και **omw-1.4**, οι οποίοι χρησιμοποιούνται αντίστοιχα για την αφαίρεση κοινών λέξεων χωρίς σημασιολογικό φορτίο, τον διαχωρισμό του κειμένου σε προτάσεις και λέξεις, τη λεμματοποίηση και τη σημασιολογική ανάλυση μέσω συνωνύμων, καθώς και την υποστήριξη πολυγλωσσικής λεμματοποίησης.

2.1. Στατιστική Ανάλυση Συλλογής

Πριν την εφαρμογή οποιουδήποτε φίλτρου, πραγματοποιήθηκε ποσοτική ανάλυση της συχνότητας εμφάνισης όρων στο σώμα κειμένων. Η ανάλυση ανέδειξε δύο σημαντικές κατηγορίες "θορύβου" που θα μπορούσαν να αλλοιώσουν την ακρίβεια του αλγορίθμου **BM25**:

- **Domain-Specific Stopwords (Safe Noise)**: Λέξεις όπως *"project"*, *"aim"*, *"proposal"*, *"consortium"* εμφανίζονταν σε ποσοστό **> 90%** των εγγράφων. Λόγω της καθολικής τους παρουσίας, ο δείκτης **IDF** *(Inverse Document Frequency)* αυτών των όρων τείνει στο μηδέν, καθιστώντας τες μη διακριτικές για την ανάκτηση.
- **Boilerplate Phrases**: Εντοπίστηκαν τυποποιημένες εκφράσεις, κυρίως νομικής ή διοικητικής φύσης. Οι φράσεις αυτές συγκεντρώθηκαν μέσω **δειγματοληπτικής ανάγνωσης** και **άντλησης λιστών από το διαδίκτυο**, και ένα **Python script** που σάρωνε τα κείμενα για τον εντοπισμό επαναλαμβανομένων φράσεων τέτοιου ίδους. Η παρουσία αυτών των φράσεων σε σημασιολογικά ασύνδετα έγγραφα δημιουργεί *false positives*, οδηγώντας σε εσφαλμένη κατάταξη.

2.2. Στάδια Καθαρισμού Κειμένου

Για την αντιμετώπιση των παραπάνω, υλοποιήθηκε η συνάρτηση analyze_and_clean_safe, η οποία εφαρμόζει τα ακόλουθα στάδια καθαρισμού σε κάθε έγγραφο ΚΑΙ σε κάθε ερώτηση:

- **Lowercasing**: Μετατροπή όλων των χαρακτήρων σε πεζά για την αποφυγή διπλότυπων εγγραφών στο ευρετήριο.

- **Κανονικοποίηση Κωδικοποίησης**: Μετατροπή HTML entities (π.χ. *&* σε *&*) και κανονικοποίηση Unicode χαρακτήρων (**NFKC Normalization**) για την εξάλειψη ασυμβατοτήτων στην κωδικοποίηση.
- **Boilerplate Removal**: Μέσω χρήσης Regular Expressions, αφαιρέθηκαν δυναμικά οι τυποποιημένες φράσεις που εντοπίστηκαν στο στάδιο της ανάλυσης, "απελευθερώνοντας" το ουσιαστικό περιεχόμενο των εγγράφων.
- **Domain-Specific Stopwords**: Διερευνήθηκαν δύο στρατηγικές:
    - **Aggressive Cleaning**: Αφαίρεση συχνών ουσιαστικών (π.χ. *project, system, development*).
    - **Minimal Cleaning**: Διατήρηση των λέξεων αυτών και αφαίρεση μόνο των *stop_phrases* και των συμβόλων.

    Παρατήρηση: Η στρατηγική Minimal Cleaning απέδωσε καλύτερα αποτελέσματα. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος BM25 διαχειρίζεται εγγενώς τις συχνές λέξεις μέσω του παράγοντα IDF, **μειώνοντας αυτόματα** τη βαρύτητά τους. Η χειροκίνητη διαγραφή τους αφαιρούσε χρήσιμο context από τα ερωτήματα.
- **Αφαίρεση Θορύβου**: Διατηρήθηκαν αποκλειστικά αλφαριθμητικοί χαρακτήρες, ενώ αφαιρέθηκαν σημεία στίξης και σύμβολα.

2.3. Ευρειστική Εξαγωγή Τίτλου

Δεδομένου ότι τα κείμενα εισόδου δεν διέθεταν διακριτό πεδίο τίτλου, αναπτύχθηκε η συνάρτηση **extract_title** για την εξαγωγή **"υπονοούμενων" τίτλων**. Ο αλγόριθμος βασίζεται σε δομικά χαρακτηριστικά του κειμένου:

- **Εντοπισμός**: Εντοπίζει την πρώτη εμφάνιση του διαχωριστικού άνω-κάτω τελεία (:).
- **Απομόνωση**: Απομονώνει το τμήμα του κειμένου που προηγείται του διαχωριστικού.
- **Φιλτράρισμα**: Εφαρμόζει **φίλτρο μήκους** (3 έως 30 λέξεις) για να απορρίψει πολύ σύντομες ετικέτες ή πολύ μεγάλα τμήματα κειμένου.

Στόχος αυτής της διαδικασίας ήταν η δημιουργία ενός νέου πεδίου (*extracted_title*) στο ευρετήριο, το οποίο έλαβε **αυξημένη βαρύτητα** (*boosting*) κατά την αναζήτηση, βελτιώνοντας την ακρίβεια.

2.4. Γλωσσική Επεξεργασία (NLP Transformation)

Μετά τον αρχικό καθαρισμό, το κείμενο υποβλήθηκε σε περαιτέρω επεξεργασία μέσω της βιβλιοθήκης **NLTK**, με στόχο τη μεγιστοποίηση της **Ανάκλησης** (Recall):

1. **Φιλτράρισμα Stopwords**: Αφαίρεση των κοινών αγγλικών *stopwords*, εμπλουτισμένων με τη λίστα του “Safe Noise” που δημιουργήθηκε στο στάδιο 2.1.
2. **Παράλληλη Μορφολογική Ανάλυση**:

    a. **Λημματοποίηση** (Lemmatization): Χρήση του *WordNet Lemmatizer* για την αναγωγή των λέξεων στη λεξικολογική τους βάση (π.χ. *"better"* σε *"good"*), διατηρώντας τη σημασιολογική ορθότητα.
    b. **Ριζική Ετυμολογία** (Stemming): Χρήση του *Porter Stemmer* για την επιθετική αποκοπή καταλήξεων (π.χ. *"simulation"* σε *"simul"*), ώστε να ομαδοποιηθούν λέξεις με κοινή ρίζα.
3. **Παραγωγή Διγραμμάτων** (Bigrams): Δημιουργία ζευγών διαδοχικών λέξεων (π.χ. *"artificial_intelligence"*) για την ενίσχυση της ακρίβειας σε φραστικές αναζητήσεις.

Η παραπάνω διαδικασία μετέτρεψε το ακατέργαστο κείμενο και τα ερωτήματα σε **καθαρή**, **πυκνή** μορφή πληροφορίας, έτοιμη για εισαγωγή στο ευρετήριο του Elasticsearch.

2.5. Ροη Επεξεργασίας & Ειδικός Χειρισμός Δομικών Χαρακτήρων

Η εφαρμογή των σταδίων προεπεξεργασίας δεν έγινε μονολιθικά, αλλά ακολουθήθηκε μια **σταδιακή ροή**. Κάθε pipeline αναλαμβάνει μια συγκεκριμένη εργασία και παραδίδει το αποτέλεσμα στο επόμενο, εξασφαλίζοντας την ακεραιότητα των δεδομένων.

Συγκεκριμένα, η αρχιτεκτονική αποτελείται από τα εξής pipelines:

1. **Sanitization Pipeline** (analyze_and_clean_safe):
    - **Είσοδος**: Ακατέργαστο κείμενο.
    - **Λειτουργία**: Εκτελεί τον "ασφαλή" καθαρισμό. Σε αυτό το στάδιο, το pipeline είναι ρυθμισμένο να αγνοεί την άνω-κάτω τελεία (:), προστατεύοντας τη δομή του εγγράφου, ενώ αφαιρεί τον HTML θόρυβο και τις *boilerplate* φράσεις.
    - **Έξοδος**: Καθαρισμένο κείμενο με διατηρημένη δομή *(temp_text)*.

2. **Extraction Pipeline** (extract_title):
    - **Είσοδος**: Το αποτέλεσμα του προηγούμενου σταδίου *(temp_text)*.
    - **Λειτουργία**: Σαρώνει το κείμενο για δομικούς δείκτες (το σύμβολο :) και εξάγει τον τίτλο σε ξεχωριστό πεδίο.
    - **Έξοδος**: Το πεδίο Τίτλου *(extracted_title)*.

3. **NLP Enrichment Pipeline** (apply_nlp):
    - **Είσοδος**: Το κείμενο μετά τον τελικό καθαρισμό *(clean_text)*.
    - **Λειτουργία**: Εφαρμόζει παράλληλους μετασχηματισμούς χρησιμοποιώντας βιβλιοθήκες **NLTK**.
    - **Έξοδος**: Τρία παράλληλα διανύσματα πληροφορίας: Λήμματα *(lemmas)*, Ρίζες *(stems)* και Διγράμματα *(bigrams)*.

Η παραπάνω αρχιτεκτονική εξασφάλισε ότι τα δεδομένα μετασχηματίζονται σταδιακά, από αδόμητη πληροφορία σε πλούσια, **πολυδιάστατα features** έτοιμα για ευρετηρίαση.

3. Σχεδιασμός & Κατασκευή Ευρετηρίου

Μετά την ολοκλήρωση της προεπεξεργασίας, το επόμενο στάδιο ήταν η κατασκευή του **Ανεστραμμένου Ευρετηρίου** στο Elasticsearch. Ο σχεδιασμός του ευρετηρίου (ir_phase1_showcase) δεν περιορίστηκε στις προεπιλεγμένες ρυθμίσεις, αλλά παραμετροποιήθηκε ώστε να υποστηρίζει **σημασιολογική επέκταση** και πολλαπλά επίπεδα ανάλυσης.

3.1 Αλυσίδα Ανάλυσης

Στο επίπεδο των ρυθμίσεων, ορίστηκαν εξειδικευμένα φίλτρα που επεξεργάζονται τη ροή των όρων πριν την αποθήκευση:

- **Custom Stopwords** (my_custom_stop): Ενσωματώθηκε η λίστα stopwords_list_for_set που δημιουργήθηκε στην Python (περιλαμβάνει τα *NLTK stopwords + Safe Noise*), εξασφαλίζοντας συνέπεια μεταξύ προεπεξεργασίας και ευρετηρίασης.
- **Length Filter**: Αποκλείστηκαν όροι με λιγότερους από **2 χαρακτήρες** για μείωση του θορύβου.
- **Synonym Graph Filter** (my_synonyms): Ορίστηκε ο μηχανισμός χαρτογράφησης συνωνύμων, ο οποίος επιτρέπει την επέκταση των όρων κατά τη διάρκεια της ανάλυσης.
- **Stemming**: Χρησιμοποιήθηκε ο αγγλικός *stemmer* του Elasticsearch για την περαιτέρω κανονικοποίηση των όρων στα πεδία που δεν είχαν επεξεργαστεί από την Python.

3.2 Στρατηγική Συνωνύμων

Για την αντιμετώπιση του προβλήματος της **"αναντιστοιχίας λεξιλογίου"**, όπου οι χρήστες χρησιμοποιούν διαφορετικούς όρους από αυτούς των εγγράφων, ορίστηκε το φίλτρο my_synonyms.

Δημιουργήθηκε ένα λεξικό εννοιών, κατόπιν ανάγνωσης και παρατήρησης δηγμάτων από το σύνολο των κείμενων, προσαρμοσμένο στο πεδίο των Ευρωπαϊκών Έργων, π.χ.:

- **Γεωγραφικοί/Οργανισμικοί**: *"UK"* με *"Great Britain"*, *"H2020"* με *"Horizon 2020"*.
- **Τεχνολογικοί**: *"AI"* με *"Artificial Intelligence"* με *"Deep Learning"*.
- **Επιστημονικοί**: *"PV"* με *"Solar Energy"*, *"Cancer"* με *"Oncology"*.

Αυτή η τεχνική αυξάνει δραματικά την **Ανάκληση** (Recall), καθώς επιτρέπει την ανάκτηση σχετικών εγγράφων ακόμη και αν δεν περιέχουν ακριβώς τις λέξεις του ερωτήματος.

3.3. Αρχιτεκτονική Αναλυτών

Για να εξυπηρετηθεί η στρατηγική αναζήτησης, υλοποιήθηκαν τρεις διακριτοί αναλυτές:

1. **synonym_analyzer (Recall-Oriented)**: Εφαρμόζει όλη την αλυσίδα: 1. *Tokenization* 2. *Lowercase* 3. *Synonyms* 4. *Stopwords* 5. *Stemming*. Είναι ο κύριος αναλυτής για τη μέγιστη δυνατή εύρεση σχετικών εγγράφων.
2. **exact_analyzer (Precision-Oriented)**: Εφαρμόζει τα πάντα εκτός από τα συνώνυμα. Χρησιμοποιείται για να ενισχύσει έγγραφα που περιέχουν τους ακριβείς όρους που έγραψε ο χρήστης.
3. **python_analyzer (Passthrough)**: Ένας απλός αναλυτής (**Whitespace Tokenizer** + **Lowercase**) που δεν πειράζει τις λέξεις. Χρησιμοποιείται αποκλειστικά για τα πεδία που έχουν ήδη υποστεί NLP επεξεργασία από την Python (*lemmas, stems, bigram*s), ώστε να μην αλλοιωθεί η δουλειά της προεπεξεργασίας.

3.4. Σχήμα Δεδομένων Mappings

Το σχήμα του εγγράφου σχεδιάστηκε με την τεχνική **Multi-fields**, επιτρέποντας στο ίδιο κείμενο να ευρετηριαστεί με διαφορετικούς τρόπους:

- **text**: Το κύριο πεδίο κειμένου, αναλυμένο με **συνώνυμα**.
    - **text.exact**: Υπο-πεδίο του κειμένου, αναλυμένο αυστηρά (χωρίς συνώνυμα) για ακριβές ταίριασμα.
- **title_extracted**: Το πεδίο του τίτλου (όπως εξήχθη στη Φάση 2.5), αναλυμένο με συνώνυμα. Λαμβάνει ειδική μεταχείριση στην αναζήτηση.
- **text_lemmatized**, **text_stemmed**, **text_bigrams**: Πεδία που αποθηκεύουν τα αποτελέσματα της Python προεπεξεργασίας, αναλυμένα με τον python_analyzer.

3.5. Ρύθμιση Πιθανοτικού Μοντέλου

Η απόδοση του αλγορίθμου **BM25** εξαρτάται καθοριστικά από δύο παραμέτρους: το *k1* (που ελέγχει τον κορεσμό της συχνότητας όρων) και το *b* (που ελέγχει την κανονικοποίηση μήκους). Αντί να βασιστούμε στις προεπιλεγμένες ρυθμίσεις του Elasticsearch, προχωρήσαμε σε εξαντλητική **αναζήτηση πλέγματος** *(Grid Search)* για τον εντοπισμό του βέλτιστου συνδυασμού, δοκιμάζοντας τιμές στα εύρη:

$$b \in \{0.5, 0.6, 0.75, 0.9, 1.0\}$$

$$k1 \in \{1.2, 1.6, 2.0, 3.0\}$$

Τα πειράματα κατέδειξαν ως βέλτιστο τον συνδυασμό $k1 = 2.0$ και $b = 1.0$. Η ερμηνεία αυτής της επιλογής βασίζεται στα ειδικά χαρακτηριστικά της συλλογής μας και αναλύεται παρακάτω.

Α. Κανονικοποίηση Μήκους ($b = 1.0$)

Επιλέχθηκε η μέγιστη δυνατή τιμή ($b = 1.0$ έναντι της *default 0.75*) για την εφαρμογή **αυστηρής κανονικοποίησης μήκους**. Η επιλογή αυτή συνδέεται άμεσα με την προεπεξεργασία που

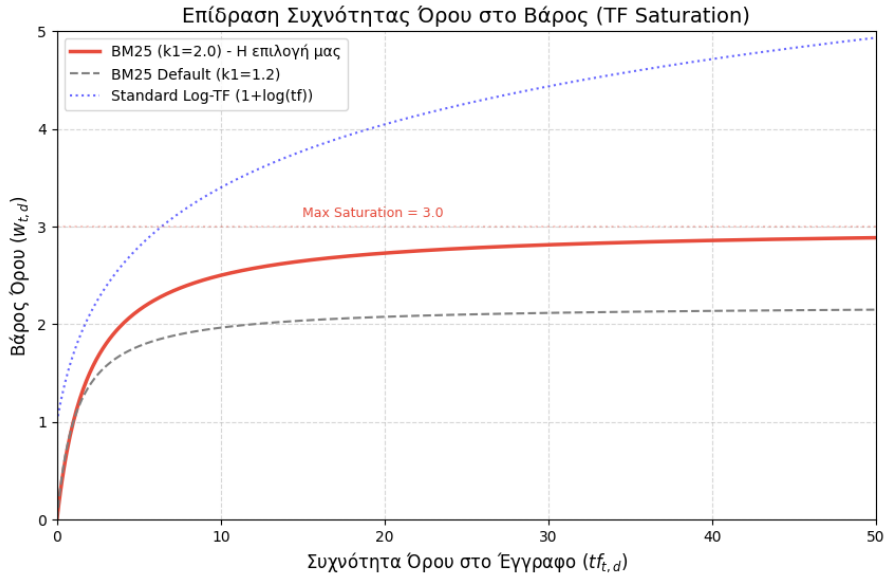
προηγήθηκε: εφόσον είχαμε ήδη αφαιρέσει τον "θόρυβο", το εναπομείναν μήκος του κειμένου αποτελούσε ισχυρή ένδειξη πληροφοριακής πυκνότητας.

Με b = 1.0, τα κείμενα που παρέμεναν αδικαιολόγητα μεγάλα χωρίς να περιέχουν υψηλή συγκέντρωση των όρων αναζήτησης, "**τιμωρήθηκαν**" δικαίως, αποτρέποντας την προκατάληψη υπέρ των μακροσκελών εγγράφων.

Β. Κορεσμός Συχνότητας (*k1* = 2.0) και Ανάλυση TF

Η τιμή *k1* αυξήθηκε στο 2.0 (έναντι της *default 1.2*) για να επιτρέψει στη συχνότητα εμφάνισης ενός όρου να επηρεάζει περισσότερο το σκορ πριν επέλθει κορεσμός. Στα επιστημονικά κείμενα, η επανάληψη συγκεκριμένων όρων δεν είναι τυχαία, αλλά συνήθως υποδηλώνει **ισχυρή θεματική εστίαση**, κάτι που θέλαμε να επιβραβεύσει ο αλγόριθμος.

Τη συμπεριφορά αυτή επιβεβαιώνει η ανάλυση της καμπύλης κορεσμού στο παρακάτω διάγραμμα:



Ερμηνεία Διαγράμματος:

Το διάγραμμα αναδεικνύει τη θεμελιώδη διαφορά του μοντέλου μας (Κόκκινη Γραμμή) σε σχέση με τις προεπιλογές:

Έλεγχος Κορεσμού: Σε αντίθεση με την μπλε διακεκομμένη γραμμή (**Standard Log-TF**) που αυξάνεται επ' άπειρον, οι καμπύλες του **BM25** συγκλίνουν σε ένα ανώτατο όριο *(ασύμπτωτη)*, προστατεύοντας το σύστημα από το **"keyword stuffing"**.

Αργός Κορεσμός: Συγκρίνοντας την κόκκινη καμπύλη (*k1*=2.0) με την γκρι (*Default k1=1.2*), παρατηρούμε ότι η δική μας καμπύλη **κορέζεται πιο αργά** και σε υψηλότερο επίπεδο $Score_{max} \approx$ 3.0.

Αυτό πρακτικά σημαίνει ότι το σύστημά μας συνεχίζει να "επιβραβεύει" την εμφάνιση μιας λέξης-κλειδί ακόμα και αν έχει ήδη εμφανιστεί 5-10 φορές. Αυτή η συμπεριφορά είναι επιθυμητή για τον εντοπισμό εγγράφων που εμβαθύνουν σε ένα θέμα, διαχωρίζοντάς τα από εκείνα που κάνουν απλώς μια επιφανειακή αναφορά.

3.6. Εποπτεία & Επαλήθευση Μετασχηματισμών

Πριν την εκτέλεση των ερωτημάτων, πραγματοποιήθηκε **δειγματοληπτικός έλεγχος** για την επαλήθευση της ακεραιότητας των δεδομένων. Η παρακάτω εικόνα αποτυπώνει την "πορεία" ενός εγγράφου *(Doc ID: 215866)* μέσα από τα στάδια επεξεργασίας που ορίστηκαν, επιβεβαιώνοντας τη σωστή λειτουργία της **Multi-field αρχιτεκτονικής**.

```
DOC ID: 215866 | Title: a 3d printed affordable myoelectrical prosthetic hand of personalizeable size for optimal comfort and
functionality
--------------------------------------------------------------------------------------------------
 [ORIGINAL]:
    A 3D printed, affordable myoelectrical prosthetic hand of personalizeable size for optimal
    comfort and functionality: The needs of prosthetic hand users are still widely unaddressed, with
    1/5 of end-u...

 [CLEANED]:
    a 3d printed affordable myoelectrical prosthetic hand of personalizeable size for optimal
    comfort and functionality the needs of prosthetic hand users are still widely unaddressed with 1
    5 of end user...

 [LEMMATIZED]:
    3d printed affordable myoelectrical prosthetic hand personalizeable size optimal comfort
    functionality need prosthetic hand user still widely unaddressed end user giving prosthesis use
    altogether disc...

 [STEMMED]:
    3d print afford myoelectr prosthet hand personaliz size optim comfort function need prosthet
    hand user still wide unaddress end user give prosthesi use altogeth discomfort poor function key
    complaint ...

 [BIGRAMS]:
    3d_printed printed_affordable affordable_myoelectrical myoelectrical_prosthetic prosthetic_hand
    hand_personalizeable personalizeable_size size_optimal optimal_comfort comfort_functionality
    functionali...
==================================================================================================
```

Το συγκεκριμένο έγγραφο αφορά τεχνολογία προσθετικής (**"prosthetic hand"**) και αναδεικνύει την αξία των διαφορετικών αναπαραστάσεων:

1 **[CLEANED]**: Παρατηρούμε ότι ειδικοί χαρακτήρες όπως το κλάσμα "*1/5"* καθαρίστηκαν σε *"1 5"*, ενώ αφαιρέθηκαν τα σημεία στίξης, αφήνοντας ένα "επίπεδο" κείμενο έτοιμο για *tokenization*.

2 **[LEMMATIZED]** (Σημασιολογία): Ο αλγόριθμος αφαίρεσε επιτυχώς τα **stopwords** (*"a"*, *"with"*, *"of"*) κρατώντας μόνο την ουσία. Επίσης, εφάρμοσε γραμματική ομαλοποίηση, μετατρέποντας τον πληθυντικό ***"users"*** στον ενικό ***"user"***, διατηρώντας όμως ολόκληρες τις λέξεις για ακριβή αναζήτηση.

3 **[STEMMED]** (Ανάκληση): Εδώ φαίνεται η επιθετική λειτουργία του **Porter Stemmer**. Λέξεις όπως ***"affordable"*** έγιναν ***"afford"*** και ***"myoelectrical"*** έγιναν ***"myoelectr"***. Αυτό είναι κρίσιμο για την Ανάκληση, καθώς θα επιτρέψει στο σύστημα να βρει το έγγραφο ακόμη και αν ο χρήστης αναζητήσει *"myoelectricity"* ή *"affordability"*, καθώς όλα ανάγονται στην ίδια ρίζα.

4 **[BIGRAMS]** (Φραστική Συνοχή): Το σύστημα αναγνώρισε και συνένωσε σωστά έννοιες-κλειδιά. Το ***"3d printed"*** έγινε ***"3d_printed"*** και το ***"prosthetic hand"*** έγινε ***"prosthetic_hand"***. Αυτά τα *tokens* λειτουργούν ως μοναδικοί όροι στο ευρετήριο, διευκολύνοντας τον εντοπισμό συγκεκριμένων τεχνολογικών όρων χωρίς την ανάγκη πολύπλοκων *query strings*.

3.7. Μαζική Εισαγωγή Δεδομένων

Η διαδικασία ολοκληρώθηκε με την πλήρωση του ευρετηρίου μέσω του **bulk API** της βιβλιοθήκης **Elasticsearch**. Αυτό το στάδιο αποτελεί την έμπρακτη εφαρμογή του σχήματος (**Mapping**) που σχεδιάστηκε στην ενότητα 3.4.

Συγκεκριμένα, πραγματοποιήθηκε η αντιστοίχιση των στηλών του pandas DataFrame (που προέκυψαν από την Python προεπεξεργασία) στα αντίστοιχα πεδία του Elasticsearch:

- row['clean_text'] → _source['text'] **(Κύριο πεδίο)**
- row['extracted_title'] → _source['title_extracted'] **(Δομικό πεδίο)**
- row['lemmatized_text'] → _source['text_lemmatized'] **(NLP πεδίο)**
- row['stemmed_text'] → _source['text_stemmed'] **(NLP πεδίο)**
- row['bigrams_text'] → _source['text_bigrams'] **(NLP πεδίο)**

Η χρήση της μεθόδου **bulk** εξασφάλισε την ταχεία και αποδοτική ευρετηρίαση του συνόλου των εγγράφων, καθιστώντας το σύστημα έτοιμο για την εκτέλεση ερωτημάτων.

4. Στρατηγική Αναζήτησης & Αξιολόγηση Αποτελεσμάτων

Με το ευρετήριο πλήρως δομημένο, το επόμενο βήμα ήταν η σχεδίαση της στρατηγικής ανάκτησης. Η διαδικασία αυτή δεν ήταν στατική, αλλά προέκυψε μέσω πειραματισμού για την εξισορρόπηση των αντικρουόμενων στόχων της Ακρίβειας και της Ανάκλησης.

4.1. Ερώτημα

Η αναζήτηση υλοποιήθηκε μέσω ενός **σύνθετου Boolean Query**, το οποίο επιτρέπει τον συνδυασμό πολλαπλών πεδίων με διαφορετικούς **συντελεστές βαρύτητας** *(Boosting)*. Η λογική της κατανομής βαρών περιγράφεται ως εξής:

1. **Phrase Ranking:**

- **match_phrase (slop: 0) → Boost 10.0:** Αποτελεί τον ισχυρότερο παράγοντα κατάταξης. Αναζητά τις λέξεις του χρήστη ακριβώς με τη σειρά που δόθηκαν. Όπως φάνηκε στα logs, αυτό παράγει πολύ υψηλά σκορ (άνω του 3000), φέρνοντας τα απόλυτα σχετικά έγγραφα στην κορυφή.
- **match_phrase (slop: 2) → Boost 2.0:** Επιτρέπει μικρή απόσταση *(έως 2 λέξεις)* μεταξύ των όρων.

2. **Precision Layer:**
   - **text.exact → Boost 7.0:** Αναζήτηση στο πεδίο χωρίς συνώνυμα. Στόχος είναι να επιβραβευθούν τα κείμενα που περιέχουν την **αυθεντική ορολογία** του χρήστη.
3. **Recall Layer:**
   - **text_lemmatized → Boost 1.5:** Αναζήτηση βάσει λημμάτων για **σημασιολογική ορθότητα.**
   - **text & text_stemmed → Boost 1.0:** Αναζήτηση βάσει ριζών. Λειτουργεί ως "βάση" *(baseline)* για να μην χάνονται έγγραφα που έχουν σπάνιες μορφολογικές παραλλαγές.
4. **Fuzziness Layer**
   - **text (fuzziness: AUTO) → Boost 0.5**: Προστέθηκε ως τελευταίο επίπεδο ένας κανόνας ασαφούς αναζήτησης με πολύ χαμηλό βάρος.
     - **Σκοπός:** Να λειτουργήσει ως "δίχτυ ασφαλείας" για την κάλυψη ορθογραφικών λαθών ή μικρών παραλλαγών σε ονόματα έργων.
     - **Χαμηλό Βάρος:** Του δόθηκε σκόπιμα χαμηλή βαρύτητα ώστε να βοηθάει στα δύσκολα ερωτήματα χωρίς να εισάγει θόρυβο στα αποτελέσματα που έχουν ήδη καλυφθεί από τα επίπεδα ακρίβειας.
5. **Απενεργοποιημένα Πεδία:**

   Κατά τον πειραματισμό, τα πεδία text_bigrams και title_extracted έλαβαν **μηδενικό βάρος (boost: 0.0)** για τους εξής λόγους:

   - **Bigrams**: Η ισχυρή φραστική αναζήτηση (match_phrase) κάλυπτε αποδοτικότερα την ανάγκη εντοπισμού φράσεων από τα στατικά bigrams, τα οποία συχνά εισήγαγαν θόρυβο.
   - **Extracted Titles**: Η ευριστική μέθοδος εξαγωγής τίτλου (βάσει του διαχωριστικού :) αποδείχθηκε **ασταθής** λόγω της δομικής ανομοιογένειας της συλλογής. Σε πολλά έγγραφα, η απουσία σαφούς διαχωριστικού ή η ακανόνιστη μορφοποίηση οδήγησε σε εσφαλμένη τμήματοποίηση, δημιουργώντας **"ψευδο-τίτλους"** με άσχετη πληροφορία που αλλοίωναν την ακρίβεια της κατάταξης.

4.2. Μεθοδολογία Αξιολόγησης: Bounded Metrics

Για την αξιολόγηση χρησιμοποιήθηκε το πρότυπο εργαλείο *trec_eval*. Ωστόσο, επειδή το πλήθος των σχετικών εγγράφων (R_q) για πολλά ερωτήματα ήταν μικρότερο από το πλήθος των

αποτελεσμάτων που ζητούσαμε (k), η κλασική μετρική **Precision@*k*** παρουσίαζε μια στρεβλή εικόνα.

Για τον λόγο αυτό, εισάγουμε την έννοια της **Bounded Precision**. Ορίζουμε αρχικά το **Εφικτό Παράθυρο Αναζήτησης** (k_{eff}):

$$k_{eff} = \min(k, |R_q|)$$

Όπου $|R_q|$ είναι το σύνολο των υπαρκτών σχετικών εγγράφων. Συνεπώς, η απόδοση υπολογίζεται ως το ποσοστό επιτυχίας εντός του εφικτού στόχου:

$$\text{Bounded Precision@}k = \frac{|Retrieved_k \cap R_q|}{k_{eff}} \times 100$$

Με αυτόν τον τρόπο, το σύστημα δεν "τιμωρείται" όταν δεν μπορεί να γεμίσει και τις 50 θέσεις με σχετικά έγγραφα, εφόσον αυτά δεν υπάρχουν στη συλλογή.

4.3. Πειραματικά Αποτελέσματα

Η αξιολόγηση πραγματοποιήθηκε για τρία σενάρια: ***k*** **∈ {20, 30, 50}.**

Α. Συγκεντρωτική Απόδοση (TREC Metrics)

Τα αποτελέσματα δείχνουν σταθερή βελτίωση του **MAP** καθώς αυξάνεται το παράθυρο αναζήτησης, φτάνοντας σε εξαιρετικά επίπεδα:

Metric	*k=20*	*k=30*	*k=50*	*Ερμηνεία*
MAP	0.7559	0. 8006	0.8137	Εξαιρετικά υψηλή ποιότητα κατάταξης
Precision@5	0.9200	0.9200	0.9200	Σταθερά υψηλή ακρίβεια στην κορυφή.
Recall	0.8789	0. 9589	0.9900	Ανάκτηση του 99% της πληροφορίας στο *k*=50.

Β. Ανάλυση Σκορ & Κατάταξης

Η ανάλυση των logs ανέδειξε το εύρος των σκορ που παράγει το **Elasticsearch**.

Στο ερώτημα ***Q01***, το πρώτο έγγραφο *(Doc ID: 193378)* πέτυχε σκορ **3003.65**, ενώ το δεύτερο **1102.90**. Αυτή η τεράστια διαφορά οφείλεται στο match_phrase (**Boost 10.0**), το οποίο "κλείδωσε" την ακριβή φράση στον τίτλο, διαχωρίζοντας ξεκάθαρα το βέλτιστο αποτέλεσμα από τα υπόλοιπα.

Γ. Ανάλυση ανά Ερώτημα

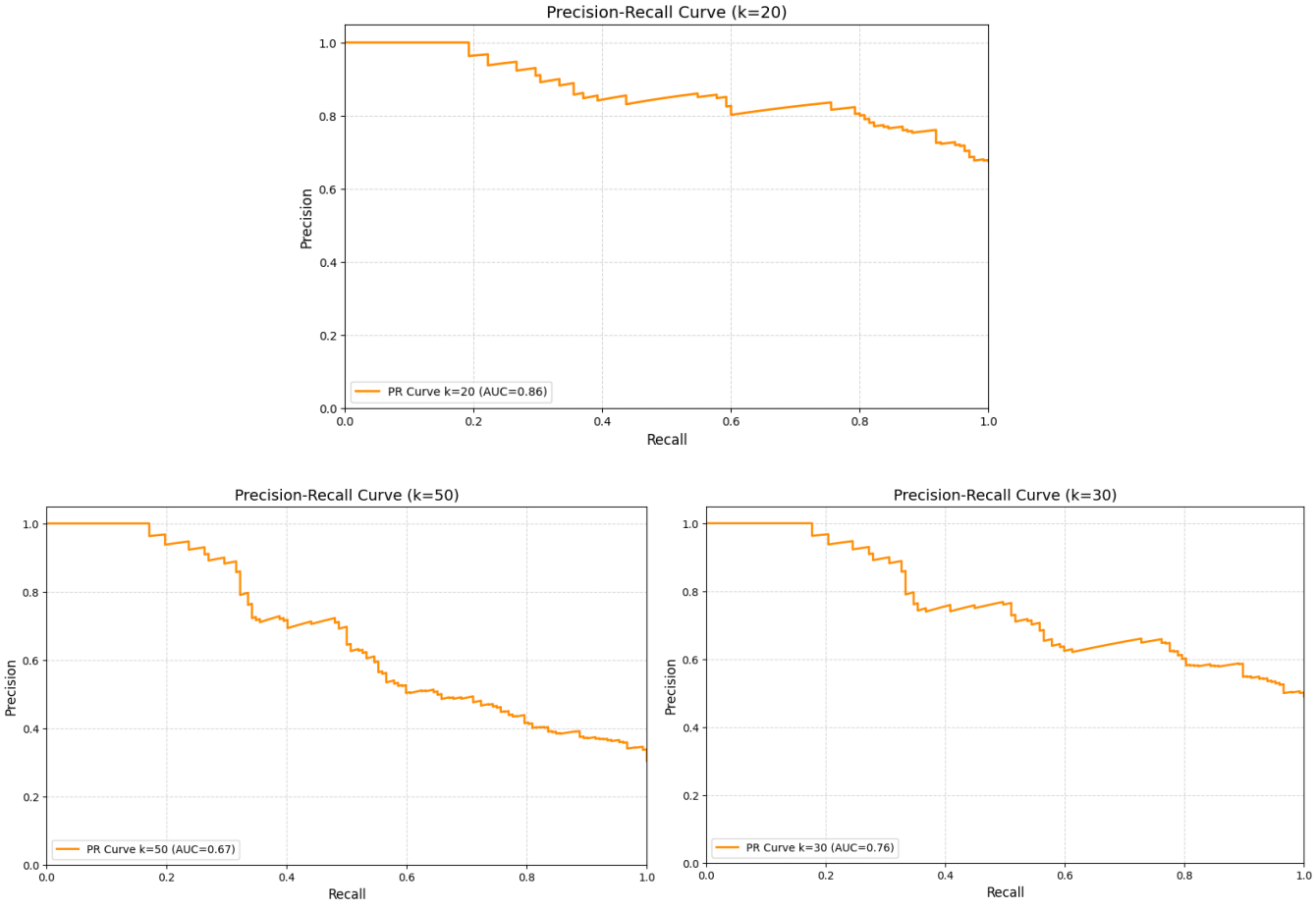
Η απόδοση ανά ερώτημα με βάση τις προσαρμοσμένες μετρικές για k=50:

QUERY ID	KNOWN	FOUND	BOUNDED SCORE	ΣΧΟΛΙΟ
Q01, Q02, Q03	16, 12, 14	All Found	100%	Απόλυτη επιτυχία.
Q04, Q05, Q06	14, 16, 19	All Found	100%	Απόλυτη επιτυχία.
Q7, Q08, Q09	16, 14, 21	All Found	100%	Απόλυτη επιτυχία.
Q10	10	9	90.0%	Δεν ανακτήθηκε ένα έγγραφο.

Συνολικό Αποτέλεσμα: Από το σύνολο των **152** σχετικών εγγράφων, εντοπίστηκαν τα **1**51, επιτυγχάνοντας **Global Recall** 99.34%.

Δ. Ανάλυση Καμπυλών Precision-Recall (PR Curves)

Για τη βαθύτερη κατανόηση της συμπεριφοράς του αλγορίθμου κατάταξης, δημιουργήθηκαν οι καμπύλες Precision-Recall για τα τρία σενάρια (k=20, 30, 50). Οι καμπύλες αυτές απεικονίζουν την ανταλλαγή μεταξύ Ακρίβειας και Ανάκλησης σε διαφορετικά επίπεδα ανάκτησης.



Ερμηνεία Διαγραμμάτων:

Και στα τρία διαγράμματα, η καμπύλη ξεκινά από το σημείο *(0.0, 1.0)* και διατηρείται σε υψηλά επίπεδα **Precision (> 0.8)** για μεγάλο τμήμα του άξονα **Recall**. Τα πρώτα έγγραφα που επιστρέφει το σύστημα είναι σχεδόν πάντα τα σωστά, εξασφαλίζοντας την εμπιστοσύνη του χρήστη.

Για $k = 50$ η καμπύλη παρουσιάζει πτώση στα δεξιά. Αυτό είναι **αναμενόμενο και φυσιολογικό**. Όταν ζητάμε 50 αποτελέσματα για ένα ερώτημα που έχει μόνο 10-15 σχετικά έγγραφα στη βάση, το σύστημα αναγκαστικά θα συμπληρώσει τις υπόλοιπες 35 θέσεις με λιγότερο σχετικά ή άσχετα έγγραφα. Αυτό μειώνει την Ακρίβεια στα χαμηλά ranks, χωρίς όμως να σημαίνει ότι χάσαμε πληροφορία *(αφού το Recall έχει ήδη φτάσει στο 99%)*.

4.4. Συμπεράσματα

Η στρατηγική μας αποδείχθηκε εξαιρετικά αποδοτική, επιτυγχάνοντας στη διαμόρφωση μιας εξαιρετικά αποδοτικής στρατηγικής ανάκτησης

- **Μηχανισμός Ακρίβειας**: Ο συνδυασμός του **υψηλού boosting** στη φραστική αναζήτηση εξασφάλισε ότι τα κορυφαία αποτελέσματα είναι απόλυτα σχετικά με το ερώτημα.
- **Μηχανισμός Ανάκλησης**: Η σημασιολογική επέκταση μέσω **Συνωνύμων**, σε συνδυασμό με τα επίπεδα **NLP (Lemmatization/Stemming) και** με το επίπεδο **Fuzziness**, έπαιξε τον σημαντικότερο ρόλο στην επίτευξη της σχεδόν τέλειας **Ανάκλησης (99%)**. Η προσέγγιση αυτή έλυσε το πρόβλημα της **λεξιλογικής αναντιστοιχίας**, επιτρέποντας στο σύστημα να ανασύρει σχετικά έγγραφα ακόμη και όταν χρησιμοποιούσαν διαφορετική ορολογία από αυτή του χρήστη, χωρίς να εισάγει θόρυβο στην κορυφή της κατάταξης.

5. Γενικά Συμπεράσματα Εργασίας

Το σύστημα που αναπτύχθηκε επιτυγχάνει τους στόχους της εργασίας, προσφέροντας υψηλή ακρίβεια και πληρότητα αποτελεσμάτων, ενώ η μεθοδολογία που ακολουθήθηκε αποτελεί έναν αξιόπιστο οδηγό για την ανάπτυξη μηχανών αναζήτησης σε εξειδικευμένα πεδία *(επιστημονικά και ερευνητικά κείμενα)*.