ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Chatbot για Προκηρύξεις Εντεταλμένων Διδασκόντων στα Πανεπιστήμια με Χρήση LLMs»

Των φοιτητών
Τζεπέλογλου Βασίλειος
Αρ. Μητρώου: 164755

Τζεπέλογλου Χρήστος
Αρ. Μητρώου: 518153

Επιβλέπων
Μπράτσας Χαράλαμπος
Επίκουρος Καθηγητής

26 Μαΐου 2025

Ανάπτυξη Chatbot για Προκηρύξεις Εντεταλμένων Διδασκόντων στα Πανεπιστήμια με Χρήση LLMs

Κωδικός Δ.Ε. 25193

Τζεπέλογλου Βασίλειος, Τζεπέλογλου Χρήστος

Ονοματεπώνυμο εισηγητή: Μπράτσας Χαράλαμπος

Ημερομηνία ανάληψης Δ.Ε. 20/03/2025

Ημερομηνία περάτωσης Δ.Ε. 26/05/2025

Βεβαιώνουμε ότι είμαστε οι συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχουμε καταγράψει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνουμε ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Αφιερώνεται στις οικογένειές μας και στους φίλους μας.

Πρόλογος

Επιλέξαμε αυτήν τη διπλωματική εργασία, επειδή θέλαμε να εμβαθύνουμε στη λειτουργία των μεγάλων γλωσσικών μοντέλων (LLMs), τα οποία αποτελούν ένα από τα πιο ενδιαφέροντα πεδία της τεχνητής νοημοσύνης σήμερα. Συγκεκριμένα, μας ενδιέφερε ιδιαίτερα να κατανοήσουμε εις βάθος τον τρόπο με τον οποίο εκπαιδεύονται και χρησιμοποιούνται τα μοντέλα αυτά για την επεξεργασία και ανάκτηση πληροφορίας στην ελληνική γλώσσα, λόγω της σύνθετης μορφολογίας και της σχετικής έλλειψης ελληνικών δεδομένων. Επίσης, μας ενθουσίασε η ιδέα να δημιουργήσουμε ένα chatbot που θα μπορούσε να διευκολύνει σημαντικά την καθημερινότητα των καθηγητών, οι οποίοι συχνά χρειάζεται να αφιερώνουν πολύ χρόνο αναζητώντας πληροφορίες και έγγραφα σχετικά με τις προκηρύξεις τους.

Μέσα από τη διαδικασία υλοποίησης του chatbot αποκτήσαμε πρακτικές γνώσεις για τις σύγχρονες τεχνικές επεξεργασίας φυσικής γλώσσας, όπως τη δημιουργία embeddings, την αποθήκευση και ανάκτηση πληροφορίας μέσω διανυσματικών βάσεων δεδομένων και τη μέτρηση ομοιότητας εγγράφων. Η ενασχόληση με αυτό το έργο μάς έδωσε τη δυνατότητα να αντιληφθούμε καλύτερα τις προκλήσεις αλλά και τις ευκαιρίες που παρέχει η τεχνητή νοημοσύνη στην πράξη. Τέλος, η συνεργασία μας, ως αδέλφια, συνέβαλε στο να μοιραστούμε γνώσεις και να αναπτύξουμε συμπληρωματικές δεξιότητες, αποκτώντας πολύτιμη εμπειρία τόσο σε τεχνικό όσο και σε προσωπικό επίπεδο.

Περίληψη

Η εργασία μας έχει ως αντικείμενο την ανάπτυξη ενός chatbot το οποίο επιτρέπει την εύκολη και γρήγορη αναζήτηση εγγράφων που αφορούν προκηρύξεις θέσεων εντεταλμένων διδασκόντων στα Ελληνικά πανεπιστήμια μέσω ερωτήσεων σε φυσική γλώσσα. Στόχος μας ήταν να δημιουργήσουμε ένα χρήσιμο και εύχρηστο εργαλείο, αξιοποιώντας τις δυνατότητες που παρέχουν τα μεγάλα γλωσσικά μοντέλα (LLMs) εκπαιδευμένα στην ελληνική γλώσσα.

Ξεκινώντας, αντλήσαμε έγγραφα σχετικά με προκηρύξεις θέσεων από τη ΔΙΑΥΓΕΙΑ, την επίσημη πηγή δημόσιων εγγράφων του ελληνικού κράτους. Στη συνέχεια, επεξεργαστήκαμε αυτά τα έγγραφα, εφαρμόζοντας κατάλληλες τεχνικές επεξεργασίας φυσικής γλώσσας ώστε να είναι συμβατά με τα γλωσσικά μοντέλα. Τα μοντέλα που επιλέξαμε — GreekBERT, Meltemi, και Krikri — χρησιμοποιήθηκαν για τη δημιουργία embeddings για κάθε έγγραφο, τα οποία αποθηκεύσαμε σε μια διανυσματική βάση δεδομένων μαζί με βασικά μεταδεδομένα όπως τίτλο και URL του εγγράφου.

Κατασκευάσαμε ένα φιλικό προς το χρήστη interface, στο οποίο ο χρήστης εισάγει ερωτήσεις στα ελληνικά. Κάθε ερώτηση κωδικοποιείται από το μοντέλο σε διανυσματική μορφή και, συγκρίνοντας το διάνυσμα αυτό με τα embeddings των εγγράφων, επιστρέφονται τα πιο σχετικά αποτελέσματα.

Τέλος, πραγματοποιήσαμε συγκριτικά πειράματα (benchmarks) για να αξιολογήσουμε την απόδοση διαφόρων γλωσσικών μοντέλων. Τα αποτελέσματα έδειξαν ότι και τα τρία μοντέλα είχαν υψηλές επιδόσεις, το καθένα με διαφορετικά πλεονεκτήματα: το KriKri πέτυχε την καλύτερη συνολική σχετικότητα, το Meltemi εντόπισε πιο άμεσα το πιο σχετικό έγγραφο, ενώ το GreekBERT συνδύασε ποιοτική κατάταξη με υψηλή υπολογιστική αποδοτικότητα. Η εργασία αυτή μας βοήθησε να κατανοήσουμε σε βάθος τις δυνατότητες αλλά και τις ιδιαιτερότητες των LLMs στην ελληνική γλώσσα, επιβεβαιώνοντας παράλληλα τη χρησιμότητα τέτοιων συστημάτων στον ακαδημαϊκό τομέα.

«Development of a Chatbot for University Teaching Staff Announcements Using Large Language Models (LLMs)»

Tzepeloglou Vasileios

Tzepeloglou Christos

Abstract

Our thesis focuses on the development of a chatbot designed to facilitate the easy and efficient retrieval of documents related to the announcement of appointed teaching staff positions at Greek universities, through natural language queries. Our goal was to create a useful and user-friendly tool by leveraging the capabilities of large language models (LLMs) trained in the Greek language.

To begin with, we collected relevant documents from Diavgeia, the official portal for public administrative documents in Greece. These documents were then preprocessed using appropriate natural language processing (NLP) techniques to ensure compatibility with the selected language models. We utilized three Greek-oriented LLMs — GreekBERT, Meltemi, and Krikri — to generate dense vector embeddings for each document. These embeddings were stored in a vector database, along with key metadata such as the document title and URL.

We developed a user-friendly interface where users can submit queries in Greek. Each query is encoded into a vector representation by the language model, and by comparing this vector to those of the stored documents, the system returns the most relevant results.

Finally, we conducted benchmarks to evaluate the performance of various language models. The results showed that all three models performed well, each with different advantages: KriKri achieved the best overall relevance, Meltemi more directly identified the most relevant document, while GreekBERT combined qualitative ranking with high computational efficiency. This thesis helped us to understand in depth the capabilities and specificities of LLMs in the Greek language, while confirming the usefulness of such systems in the academic sector.

Ευχαριστίες

Θα θέλαμε να εκφράσουμε τις θερμότερες ευχαριστίες μας στους καθηγητές μας για τη συνεχή και πολύτιμη καθοδήγησή τους, που ήταν καθοριστική σε κάθε βήμα της ακαδημαϊκής μας πορείας. Η στήριξη και οι συμβουλές τους υπήρξαν ανεκτίμητες, βοηθώντας μας να ξεπεράσουμε τις δυσκολίες και να πετύχουμε τους στόχους μας.

Ευχαριστούμε θερμά τον επιβλέποντα καθηγητή κ. Μπράτσα Χαράλαμπο, και την αμέτρητη συνεισφορα της ομαδας του και κυριως τους κ. Αγγελίδη Αλέξανδρο και κ. Αναστασιάδη Ευστάθιο Κωνσταντίνο.

Ένα μεγάλο ευχαριστώ οφείλουμε επίσης στην οικογένειά μας, που στάθηκε αδιάκοπα δίπλα μας με υπομονή, αγάπη και εμπιστοσύνη. Η παρουσία τους ήταν πολύτιμη όχι μόνο στις στιγμές επιτυχίας, αλλά κυρίως στις πιο δύσκολες και απαιτητικές περιόδους της εργασίας μας. Ευχαριστούμε επίσης θερμά τους φίλους μας για την καθημερινή ενθάρρυνση και τη συμπαράστασή τους, που μας βοήθησε να προχωρήσουμε μπροστά με αισιοδοξία.

Ιδιαίτερα, θέλουμε να εκφράσουμε τη χαρά μας που είχαμε την ευκαιρία να δουλέψουμε μαζί ως αδέλφια. Η συνεργασία αυτή μας επέτρεψε να γνωριστούμε ακόμα καλύτερα, να ανακαλύψουμε τις δυνατότητες και τα όριά μας, αλλά και να δημιουργήσουμε κάτι που μας κάνει χαρούμενους. Τέλος, ευχαριστούμε όλους όσους πίστεψαν σε εμάς και μας βοήθησαν να φτάσουμε μέχρι εδώ, δίνοντάς μας δύναμη και έμπνευση να συνεχίσουμε.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
LLM	Large Language Model
NLP	Natural Language Processing (Επεξεργασία Φυσικής Γλώσσας)
RAG	Retrieval-Augmented Generation (Ανάκτηση Επαυξημένης Παραγωγής)
POS	Part of Speech (Μέρη του Λόγου – κατηγοριοποίηση λέξεων)
NER	Named Entity Recognition (Αναγνώριση Ονοματικών Οντοτήτων)
NLI	Natural Language Inference (Συμπερασμός Φυσικής Γλώσσας)
OCR	Optical Character Recognition (Οπτική Αναγνώριση Χαρακτήρων)
TF-IDF	Term Frequency - Inverse Document Frequency
HNSW	Hierarchical Navigable Small World (αλγόριθμος ευρετηρίασης)
CLS	[CLS] token – ειδικό token της αρχιτεκτονικής BERT
MRR	Mean Reciprocal Rank (Μέση Αντιστρόφως Κατατακτική Θέση)
CG@k	Cumulative Gain at rank k
DCG@k	Discounted Cumulative Gain at rank k
IDCG@k	Ideal Discounted Cumulative Gain at rank k
nDCG@k	Normalized Discounted Cumulative Gain at rank k
RR	Reciprocal Rank
GCS	Google Cloud Storage
VRAM	Video Random Access Memory
GPU	Graphics Processing Unit
CPU	Central Processing Unit
JSON	JavaScript Object Notation
PDF	Portable Document Format
URL	Uniform Resource Locator
DB	DataBase

Κεφάλαιο 1ο: Εισαγωγή

1.1 Παρουσίαση Θέματος και Προβληματισμός

Η σύγχρονη εποχή χαρακτηρίζεται από έναν πρωτοφανή όγκο ψηφιακά διαθέσιμης πληροφορίας. Ειδικότερα, ο δημόσιος τομέας παράγει και διακινεί καθημερινά πλήθος εγγράφων, τα οποία είναι κρίσιμα για τη διαφάνεια, την ενημέρωση των πολιτών και την εύρυθμη λειτουργία των κρατικών μηχανισμών. Στην Ελλάδα, πλατφόρμες όπως η "ΔΙΑΥΓΕΙΑ" έχουν συμβάλλει καθοριστικά στην προσβασιμότητα αυτών των εγγράφων, ωστόσο, η αποτελεσματική αναζήτηση και ανάκτηση συγκεκριμένων πληροφοριών μέσα από αυτά παραμένει συχνά μια χρονοβόρα και πολύπλοκη διαδικασία για τον μέσο χρήστη.

Η παρούσα διπλωματική εργασία εστιάζει στην ανάπτυξη ενός ευφυούς συστήματος, το οποίο αποσκοπεί στη διευκόλυνση της αλληλεπίδρασης των χρηστών με δημόσια έγγραφα του ελληνικού τομέα. Συγκεκριμένα, η έρευνα επικεντρώνεται στην κατηγορία των προκηρύξεων για θέσεις εντεταλμένων διδασκόντων στα Πανεπιστήμια. Η αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs), ειδικά εκπαιδευμένων ή προσαρμοσμένων στην ελληνική γλώσσα, σε συνδυασμό με την αρχιτεκτονική Ανάκτησης Επαυξημένης Παραγωγής (Retrieval-Augmented Generation - RAG), προσφέρει νέες δυνατότητες για την κατανόηση ερωτημάτων σε φυσική γλώσσα και την παροχή σχετικών απαντήσεων βασισμένων στο περιεχόμενο των εν λόγω εγγράφων και όχι μόνο.

1.1.1 Η Πρόκληση της Αναζήτησης σε Ελληνικά Δημόσια Έγγραφα

Η αναζήτηση πληροφοριών εντός των ελληνικών δημοσίων εγγράφων, όπως οι προκηρύξεις που δημοσιεύονται στη "ΔΙΑΥΓΕΙΑ", παρουσιάζει σημαντικές προκλήσεις. Τα έγγραφα αυτά συχνά χαρακτηρίζονται από εκτενές περιεχόμενο, εξειδικευμένη ορολογία και τυποποιημένη, αλλά ενίοτε πολύπλοκη, δομή. Η μορφή τους, κυρίως PDF, δεν διευκολύνει την αυτοματοποιημένη εξαγωγή και ευρετηρίαση του περιεχομένου με παραδοσιακές μεθόδους [1].

Οι χρήστες που αναζητούν συγκεκριμένες πληροφορίες, όπως προθεσμίες, απαιτούμενα προσόντα ή λεπτομέρειες για μια θέση, μπορεί να χρειαστεί να ανατρέξουν σε πολλαπλά έγγραφα και να δαπανήσουν σημαντικό χρόνο στην ανάγνωση και τον εντοπισμό των κρίσιμων στοιχείων. Οι υπάρχουσες μηχανές αναζήτησης, αν και χρήσιμες, συχνά βασίζονται σε αντιστοίχιση λέξεων-κλειδιών και ενδέχεται να μην κατανοούν πλήρως τη σημασιολογική διάσταση του ερωτήματος του χρήστη, οδηγώντας σε λιγότερο σχετικά αποτελέσματα ή στην ανάγκη για διαδοχικές, τροποποιημένες αναζητήσεις. Επομένως, καθίσταται επιτακτική η ανάγκη για πιο ευφυή και ευέλικτα εργαλεία που θα επιτρέπουν την υποβολή ερωτημάτων σε φυσική ελληνική γλώσσα και την ανάκτηση ακριβών εγγράφων με αποτελεσματικό τρόπο. Τελος, τα δημόσια έγγραφα συχνά είναι δύσκολα στη χρήση επειδή δημοσιεύονται ως σαρωμένα PDF με περίεργη μορφοποίηση, ενώ το περιεχόμενό τους περιλαμβάνει δύσκολη γλώσσα με νομικούς όρους και ορισμένες φορές παλαιότερες μορφές ελληνικών που μπερδεύουν τον μέσο αναγνώστη.

1.2 Στόχοι της Διπλωματικής Εργασίας

Ο κύριος στόχος της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός, η υλοποίηση και η αξιολόγηση ενός διαλογικού βοηθού (chatbot) ικανού να αναζητά κατάλληλες πληροφορίες από ένα σύνολο ελληνικών δημοσίων εγγράφων – συγκεκριμένα, προκηρύξεων για θέσεις εντεταλμένων διδασκόντων από τη "ΔΙΑΥΓΕΙΑ", επιστρέφοντας στον χρήστη τα πιο σχετικά αποτελέσματα.

Για την επίτευξη του παραπάνω στόχου, τίθενται οι ακόλουθοι στόχοι:

- Διερεύνηση και επιλογή κατάλληλων προεκπαιδευμένων Μεγάλων Γλωσσικών Μοντέλων (LLMs) για την ελληνική γλώσσα (GreekBert, Meltemi, KriKri).
- Συλλογή, προεπεξεργασία και οργάνωση ενός σώματος εγγράφων (dataset) από προκηρύξεις σε μορφή PDF.
- Υλοποίηση μιας αρχιτεκτονικής Ανάκτησης Επαυξημένης Παραγωγής (RAG) για τον συνδυασμό των δυνατοτήτων των LLMs με την ανάκτηση πληροφοριών από το προ επεξεργασμένο σώμα εγγράφων.
- Δημιουργία διανυσματικών αναπαραστάσεων (embeddings) για τα τμήματα των εγγράφων και αποθήκευσή τους σε κατάλληλη διανυσματική βάση δεδομένων.
- Ανάπτυξη μιας βασικής διεπαφής χρήστη (user interface) για την υποβολή ερωτημάτων και την παρουσίαση των ανακτηθέντων εγγράφων ή πληροφοριών.
- Διεξαγωγή συγκριτικών πειραμάτων (benchmarking) για την αξιολόγηση της απόδοσης των διαφορετικών επιλεγμένων γλωσσικών μοντέλων στην εργασία της αναζήτησης πληροφοριών από τα συγκεκριμένα έγγραφα.

1.3 Μεθοδολογική Προσέγγιση

Η μεθοδολογία που θα ακολουθηθεί για την υλοποίηση των στόχων της εργασίας περιλαμβάνει τα παρακάτω βασικά στάδια:

- Βιβλιογραφική Επισκόπηση: Αρχικά, θα πραγματοποιηθεί μελέτη της υπάρχουσας βιβλιογραφίας σχετικά με Μεγάλα Γλωσσικά Μοντέλα για την ελληνική γλώσσα, αρχιτεκτονικές RAG, τεχνικές επεξεργασίας φυσικής γλώσσας για δημόσια έγγραφα, και συστήματα διαλογικών βοηθών.
- Συλλογή και Προεπεξεργασία Δεδομένων: Θα γίνει συλλογή προκηρύξεων για θέσεις εντεταλμένων διδασκόντων από την πλατφόρμα "ΔΙΑΥΓΕΙΑ". Τα έγγραφα (σε μορφή PDF) θα υποβληθούν σε κατάλληλη προεπεξεργασία, η οποία θα περιλαμβάνει την εξαγωγή κειμένου, τον καθαρισμό του από περιττούς χαρακτήρες ή μορφοποιήσεις, και την τμηματοποιηση του (chunking) σε μικρότερα, διαχειρίσιμα κομμάτια κειμένου. Ο κώδικας για την επεξεργασία αυτή θα αναπτυχθεί σε περιβάλλον Google Colab.
- Δημιουργία Embeddings και Διανυσματικής Βάσης: Τα προ επεξεργασμένα τμήματα κειμένου θα μετατραπούν σε διανυσματικές αναπαραστάσεις (embeddings) χρησιμοποιώντας τα επιλεγμένα ελληνικά LLMs (GreekBert, Meltemi, KriKri) μέσω της πλατφόρμας Hugging Face. Αυτά τα embeddings, μαζί με σχετικά μεταδεδομένα (όπως ο τίτλος του εγγράφου και το URL πηγής), θα αποθηκευτούν σε μια διανυσματική βάση δεδομένων ChromaDB.
- Υλοποίηση Αρχιτεκτονικής RAG και Chatbot: Θα αναπτυχθεί ο πυρήνας της εφαρμογής RAG. Όταν ένας χρήστης υποβάλλει ένα ερώτημα σε φυσική ελληνική γλώσσα, το ερώτημα θα μετατρέπεται επίσης σε embedding. Στη συνέχεια, θα πραγματοποιείται αναζήτηση ομοιότητας (similarity search) μεταξύ του embedding του ερωτήματος και των

αποθηκευμένων embeddings των εγγράφων στη διανυσματική βάση. Τα πιο σχετικά τμήματα κειμένου θα ανακτώνται.

- Ανάπτυξη Διεπαφής Χρήστη: Θα δημιουργηθεί μια διεπαφή χρήστη με χρήση Gradio στην πλατφόρμα Hugging Face Spaces όπου ο χρήστης θα μπορεί να εισάγει τα ερωτήματά του και το σύστημα θα επιστρέφει τα ανακτηθέντα σχετικά έγγραφα ή αποσπάσματα.
- Αξιολόγηση και Συγκριτική Ανάλυση: Θα σχεδιαστεί και θα εκτελεστεί μια σειρά πειραμάτων (benchmarks) για την αξιολόγηση της απόδοσης των τριών επιλεγμένων γλωσσικών μοντέλων (GreekBert, Meltemi, KriKri) στο πλαίσιο της συγκεκριμένης εργασίας ανάκτησης πληροφορίας. Θα χρησιμοποιηθούν κατάλληλες μετρικές για τη σύγκριση της αποτελεσματικότητάς τους.

1.4 Δομή της Εργασίας

Η παρούσα διπλωματική εργασία είναι οργανωμένη σε εννέα κεφάλαια, τα οποία καλύπτουν τη θεωρητική θεμελίωση, τη μεθοδολογία υλοποίησης, την πειραματική αξιολόγηση και τα συμπεράσματα της έρευνας.

Το Κεφάλαιο 2 επικεντρώνεται στη σχετική έρευνα και το θεωρητικό υπόβαθρο. Παρουσιάζονται τα ελληνικά Μεγάλα Γλωσσικά Μοντέλα, GreekBERT, Meltemi-7B-Instruct και KriKri, οι βασικές τεχνικές Επεξεργασίας Φυσικής Γλώσσας (NLP) για την ελληνική γλώσσα, οι διανυσματικές αναπαραστάσεις (vector embeddings) και τα μέτρα ομοιότητας. Επιπλέον, γίνεται αναφορά στα υβριδικά πληροφοριακά συστήματα αναζήτησης και σε υφιστάμενα έργα που προωθούν τη διαφάνεια στον δημόσιο τομέα.

Στο Κεφάλαιο 3 περιγράφεται η διαδικασία συλλογής δεδομένων και η κατασκευή του συνόλου δεδομένων dataset. Αναλύεται η πηγή των προκηρύξεων ΔΙΑΥΓΕΙΑ, η χρήση του Colab script για τη συλλογή, η τελική δομή του dataset σε μορφή JSON και η αποθήκευσή του στο Google Drive με διαχείριση εκδόσεων.

Το Κεφάλαιο 4 εστιάζει στην προεπεξεργασία των αρχείων PDF. Αναλύονται οι τεχνικές εξαγωγής κειμένου με χρήση της βιβλιοθήκης PyPDF2, η αφαίρεση επικεφαλίδων και υποσέλιδων, ο καθαρισμός του κειμένου (αφαίρεση συνδέσμων, προστασία ημερομηνιών, κανονικοποίηση στίξης), η χρήση εκτενούς λίστας ελληνικών stop-words και η εφαρμογή ενός απλού ελληνικού stemmer. Ολοκληρώνεται με την παρουσίαση του pipeline επεξεργασίας process_document() και παραδείγματα ποιοτικού ελέγχου.

Το Κεφάλαιο 5 αναφέρεται στη δημιουργία των διανυσματικών αναπαραστάσεων (embeddings) και την κατασκευή της διανυσματικής βάσης δεδομένων. Γίνεται αναφορά στην επιλογή των μοντέλων, στην τμηματοποίηση (chunking) του κειμένου, στην τεχνική mean pooling για τα decoder-only LLMs και στη χρήση της ChromaDB για την αποθήκευση των embeddings και των σχετικών μεταδεδομένων.

Στο Κεφάλαιο 6 παρουσιάζεται ο υβριδικός αλγόριθμος αναζήτησης που αναπτύχθηκε. Περιγράφονται τα δύο μονοπάτια αναζήτησης, η μέθοδος στάθμισης των αποτελεσμάτων τους, η ενίσχυση των ακριβών αντιστοιχιών και ο αλγόριθμος υλοποίησης hybrid_search_gradio.

Το Κεφάλαιο 7 καλύπτει τον σχεδιασμό και την υλοποίηση της τελικής εφαρμογής. Παρουσιάζεται η αρχιτεκτονική του συστήματος, η χρήση Hugging Face Space με Persistent Storage, η διαχείριση του μοντέλου Meltemi-7B, η αξιοποίηση του Google Cloud Storage για την παροχή των PDF και η ανάπτυξη της διεπαφής χρήστη με Gradio.

Το Κεφάλαιο 8 επικεντρώνεται στην πειραματική αξιολόγηση και τα benchmarks των υλοποιημένων μοντέλων. Καθορίζεται το σύνολο των ερωτήσεων αξιολόγησης, οι μετρικές απόδοσης (MRR, nDCG@5, CG@5), το hardware που χρησιμοποιήθηκε και παρουσιάζονται τα ποσοτικά και ποιοτικά αποτελέσματα της σύγκρισης των GreekBERT, Meltemi και KriKri.

Τέλος, το Κεφάλαιο 9 συνοψίζει τα κύρια θέματα της εργασίας, αναγνωρίζει τους περιορισμούς της έρευνας και προτείνει κατευθύνσεις για μελλοντική εργασία, όπως η βελτιστοποίηση των μοντέλων (fine-tuning), η χρήση rerankers, βελτιώσεις στη διεπαφή χρήστη και η επέκταση του συστήματος σε άλλες κατηγορίες δημοσίων εγγράφων.

1.5 Επίλογος

Το παρόν εισαγωγικό κεφάλαιο έθεσε το πλαίσιο της διπλωματικής εργασίας, αναδεικνύοντας την πρόκληση της αναζήτησης σε ελληνικά δημόσια έγγραφα και την προοπτική που προσφέρουν οι σύγχρονες τεχνολογίες επεξεργασίας φυσικής γλώσσας για την αντιμετώπισή της. Οι στόχοι που τέθηκαν αποσκοπούν στην ανάπτυξη ενός λειτουργικού πρωτοτύπου και στην εξαγωγή χρήσιμων συμπερασμάτων σχετικά με την απόδοση διαφορετικών ελληνικών γλωσσικών μοντέλων. Η μεθοδολογική προσέγγιση και η δομή της εργασίας, όπως περιγράφηκαν, στοχεύουν στην συστηματική διερεύνηση του θέματος. Στα επόμενα κεφάλαια θα αναλυθούν διεξοδικά το θεωρητικό υπόβαθρο, η διαδικασία υλοποίησης και τα αποτελέσματα της έρευνας.

Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο και Βιβλιογραφική Ανασκόπηση

2.1 Ελληνικά LLMs: GreekBERT, Meltemi, KriKri

Η αποτελεσματική αναζήτηση και ανάκτηση πληροφοριών από εκτενή σώματα κειμένων αποτελεί μια διαρκή πρόκληση στον τομέα της Πληροφορικής. Με την εκρηκτική αύξηση των ψηφιακών δεδομένων, η ανάγκη για ευφυή συστήματα που κατανοούν και επεξεργάζονται τη φυσική γλώσσα καθίσταται ολοένα και πιο επιτακτική. Η παρούσα διπλωματική εργασία τοποθετείται σε αυτό το πλαίσιο, διερευνώντας τη χρήση σύγχρονων τεχνολογιών Τεχνητής Νοημοσύνης για την ανάπτυξη ενός διαλογικού βοηθού ανάκτησης ελληνικών εγγράφων.

Η ραγδαία εξέλιξη των Μεγάλων Γλωσσικών Μοντέλων (LLMs) έχει επιφέρει επανάσταση στον τομέα της Επεξεργασίας Φυσικής Γλώσσας (NLP). Ωστόσο, η πρόοδος αυτή δεν κατανέμεται εξίσου σε όλες τις γλώσσες. Οι γλώσσες με λιγότερους διαθέσιμους ψηφιακούς πόρους, όπως η Ελληνική, αντιμετωπίζουν σημαντικές προκλήσεις. Συχνά, τα μοντέλα που εκπαιδεύονται κυρίως σε γλώσσες με άφθονους πόρους, όπως τα Αγγλικά, μεταφέρουν προκαταλήψεις και παραδοχές που δεν είναι κατάλληλες για άλλες γλώσσες [2]. Αυτό το χάσμα επιδεινώνεται από την έλλειψη ανοιχτά προσβάσιμων, υψηλής ποιότητας γλωσσικών πόρων. Η ανάπτυξη LLMs ειδικά προσαρμοσμένων στις γλωσσικές και πολιτισμικές ιδιαιτερότητες της Ελληνικής γλώσσας είναι, επομένως, κρίσιμης σημασίας. Τέτοιες προσπάθειες όχι μόνο διασφαλίζουν δικαιότερη τεχνολογική πρόοδο αλλά και αποτρέπουν την ψηφιακή περιθωριοποίηση γλωσσικών κοινοτήτων.

Τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) αποτελούν την αιχμή του δόρατος στην έρευνα της Τεχνητής Νοημοσύνης και της Επεξεργασίας Φυσικής Γλώσσας. Πρόκειται για νευρωνικά δίκτυα με δισεκατομμύρια παραμέτρους, εκπαιδευμένα σε τεράστιους όγκους δεδομένων κειμένου, ικανά να κατανοούν, να παράγουν και να χειρίζονται την ανθρώπινη γλώσσα με πρωτοφανή ακρίβεια και ευελιξία. Ενώ η πλειονότητα της έρευνας και των διαθέσιμων μοντέλων επικεντρώνεται στην αγγλική γλώσσα, τα τελευταία χρόνια έχει σημειωθεί σημαντική πρόοδος στην ανάπτυξη LLMs και για γλώσσες με λιγότερους ψηφιακούς πόρους, όπως η ελληνική. Η διαθεσιμότητα τέτοιων μοντέλων είναι κρίσιμη για την ανάπτυξη εφαρμογών που απευθύνονται στο ελληνόφωνο κοινό και χειρίζονται ελληνικό περιεχόμενο. Στην παρούσα ενότητα, θα γίνει επισκόπηση τριών βασικών LLMs που έχουν αναπτυχθεί για την ελληνική γλώσσα και τα οποία θα αξιοποιηθούν στην παρούσα εργασία: το GreekBERT, το Meltemi και το KriKri.

2.1.1 GreekBERT

Το GreekBERT αναπτύχθηκε από τους Γιάννη Κουτσικάκη, Ηλία Χαλκίδη, Πρόδρομο Μαλακασιώτη και Ίωνα Ανδρουτσόπουλο από την Ομάδα Επεξεργασίας Φυσικής Γλώσσας του Οικονομικού Πανεπιστημίου Αθηνών [3]. Πρόκειται για μια μονο γλωσσική έκδοση της αρχιτεκτονικής BERT, αντίστοιχη με το Αγγλικό μοντέλο bert-base-uncased (12 επίπεδα, 768 κρυφές μονάδες, 12 κεφαλές προσοχής, 110 εκατομμύρια παράμετροι). Το σώμα κειμένων προεκπαίδευσης του GreekBERT ανέρχεται σε 29GB Ελληνικού κειμένου. Αυτό περιλαμβάνει το Ελληνικό μέρος της Wikipedia (1.73GB), το Ελληνικό μέρος του European Parliament Proceedings Parallel Corpus (Europarl, 0.38GB) και το Ελληνικό μέρος του OSCAR (27GB) [3]. Τα κείμενα προ επεξεργάστηκαν μετατρέποντάς τα σε πεζά και αφαιρώντας τους τόνους, αν και οι νεότεροι tokenizers ενδέχεται να χειρίζονται εγγενώς αυτή τη διαδικασία [4]. Το GreekBERT σχεδιάστηκε για γενικές εργασίες Ελληνικής NLP και έχει εφαρμοστεί σε επισήμανση μερών του λόγου (POS tagging), αναγνώριση ονοματικών οντοτήτων (NER) και εξαγωγή φυσικής γλώσσας συμπερασμάτων (NLI) [5]. Χρησιμοποιήθηκε ως σημείο αναφοράς ή ως συστατικό σε μεταγενέστερα μοντέλα και μελέτες, όπως το Ancient Greek BERT που αρχικοποιήθηκε από αυτό και σε συγκρίσεις με το Meltemi [6].

Κατά την κυκλοφορία του, το GreekBERT ξεπέρασε τις επιδόσεις των πολύγλωσσων μοντέλων M-BERT και XLM-RoBERTa σε διάφορες εργασίες Ελληνικής NLP και θεωρήθηκε κορυφαίο για πολλές Ελληνικές εργασίες.7 Είναι διαθέσιμο στο Hugging Face (nlpaueb/bert-base-greek-uncased-v1) 8, ενώ και τα αρχικά σημεία ελέγχου του TensorFlow είναι επίσης διαθέσιμα.8 Ως παλαιότερο μοντέλο τύπου BERT, έχει μικρότερο παράθυρο πλαισίου και διαφορετικές αρχιτεκτονικές ιδιότητες σε σύγκριση με νεότερα LLMs όπως το Meltemi ή το Krikri. Η απόδοσή του ενδέχεται να έχει ξεπεραστεί από αυτά τα νεότερα μοντέλα σε πολλές εργασίες, αν και οι τελειοποιημένες εκδόσεις του μπορούν ακόμα να είναι πολύ ισχυρές για συγκεκριμένες εφαρμογές.

Η εμφάνιση του GreekBERT σηματοδότησε ένα σημαντικό βήμα για την Ελληνική NLP. Πριν από μοντέλα όπως το GreekBERT, η Ελληνική NLP βασιζόταν συχνά σε πολύγλωσσα μοντέλα που ενδεχομένως δεν αποτύπωναν πλήρως τις Ελληνικές γλωσσικές ιδιαιτερότητες. Η ανάπτυξη ενός αποκλειστικά μόνο γλωσσικού μοντέλου BERT, εκπαιδευμένου σε ένα σημαντικό Ελληνικό σώμα κειμένων, παρείχε σημαντική ώθηση στην έρευνα και τις εφαρμογές της Ελληνικής NLP. Λειτούργησε ως ισχυρό σημείο αναφοράς, επιτρέποντας πιο αυστηρή αξιολόγηση των μεταγενέστερων μοντέλων. Ακόμη και με την εμφάνιση νεότερων, μεγαλύτερων αρχιτεκτονικών, ένα καλά εκπαιδευμένο μοντέλο BERT μπορεί να είναι πολύ αποτελεσματικό, ειδικά όταν τελειοποιείται για συγκεκριμένες εργασίες [6].

Εκτός από τα παραπάνω, αξίζει να αναφερθεί και το Ancient Greek BERT.13 Αν και εστιάζει στην Αρχαία και Μεσαιωνική Ελληνική, είναι σημαντικό καθώς δείχνει την επέκταση των προσπαθειών και σε ιστορικές φάσεις της γλώσσας. Είναι αξιοσημείωτο ότι το μοντέλο αυτό αρχικοποιήθηκε από το GreekBERT της ομάδας NLP του ΟΠΑ και στη συνέχεια εκπαιδεύτηκε σε μονο γλωσσικά δεδομένα από πηγές όπως το First1KGreek Project και η Perseus Digital Library [7].

2.1.2 Meltemi

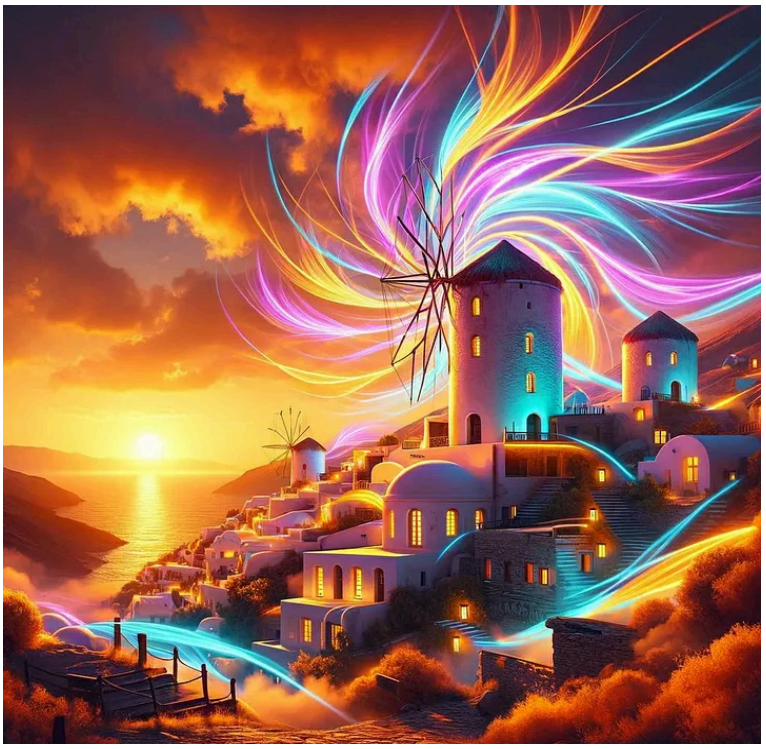
Το Meltemi είναι ένα μεγάλο γλωσσικό μοντέλο για την Ελληνική γλώσσα που αναπτύχθηκε από το Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ) του Ερευνητικού Κέντρου "Αθηνά". Η αρχιτεκτονική του βασίζεται στο ανοιχτού κώδικα LLM Mistral-7B, καθιστώντας το Meltemi ένα μοντέλο 7

δισεκατομμυρίων παραμέτρων. Εκπαιδεύτηκε ως ένα δίγλωσσο μοντέλο, διατηρώντας τις αγγλικές του δυνατότητες ενώ επεκτάθηκε για να κατανοεί και να παράγει άπταιστα κείμενο στη Νέα Ελληνική [8].

Για την προεκπαίδευση του Meltemi χρησιμοποιήθηκε ένα σώμα κειμένων περίπου 40 δισεκατομμυρίων tokens. Από αυτά, τα 28.5 δισεκατομμύρια ήταν Ελληνικά tokens που κατασκευάστηκαν από δημόσια διαθέσιμους πόρους και υποβλήθηκαν σε επεξεργασία, φιλτράρισμα και απο διπλοτυπία για τη διασφάλιση της ποιότητας των δεδομένων. Για την υποστήριξη των δίγλωσσων δυνατοτήτων, χρησιμοποιήθηκαν επιπλέον 10.5 δισεκατομμύρια Αγγλικά tokens και ένα παράλληλο Ελληνο-Αγγλικό σύνολο δεδομένων 600 εκατομμυρίων tokens. Για τη δημιουργία της έκδοσης Meltemi-7B-Instruct-v1, η οποία είναι αυτή που αξιοποιήσαμε και εμείς, που είναι σχεδιασμένη για συνομιλιακές εφαρμογές, χρησιμοποιήθηκαν περίπου 100.000 Ελληνικές οδηγίες. Αυτές περιλάμβαναν μηχανικά μεταφρασμένες εκδόσεις υπαρχόντων συνόλων δεδομένων συνομιλίας (όπως Open-Platypus, Evol-Instruct, Capybara) καθώς και ένα χειροκίνητα δημιουργημένο Ελληνικό σύνολο δεδομένων με παραδείγματα πολλαπλών γύρων για την καθοδήγηση του μοντέλου προς ασφαλείς αποκρίσεις.

Οι βασικές δυνατότητες του Meltemi περιλαμβάνουν την κατανόηση και παραγωγή κειμένου τόσο στα Ελληνικά όσο και στα Αγγλικά, με την έκδοση Instruct να εστιάζει σε συνομιλιακές χρήσεις. Σε μια μελέτη για την ταξινόμηση ειδησεογραφικών άρθρων στα Ελληνικά ανά είδος, αναφέρεται ότι ενώ γενικά LLMs όπως το Meltemi δείχνουν κάποιες δυνατότητες στην ταξινόμηση μηδενικού δείγματος (zero-shot classification), τα εξειδικευμένα μοντέλα που έχουν υποστεί τελειοποίηση (όπως το Greek BERT στην εν λόγω μελέτη) μπορούν να τα ξεπεράσουν.9 Αυτό υποδηλώνει ότι η ισχύς του Meltemi μπορεί να είναι πιο γενική ή να απαιτεί περαιτέρω τελειοποίηση για συγκεκριμένες εργασίες. Το Meltemi διατίθεται με άδεια ανοιχτού κώδικα Apache 2.0 και είναι προσβάσιμο μέσω του Hugging Face (ilsp/Meltemi-7B-v1, ilsp/Meltemi-7B-Instruct-v1) [6]. Μια πιθανή πρόκληση είναι ότι, παρόλο που η προεκπαίδευση του Mistral-7B επεκτάθηκε με Ελληνικά δεδομένα, η αρχική του βάση ήταν κυρίως Αγγλική, και η αποτελεσματικότητά του θα εξαρτηθεί από την ποιότητα και την ευρύτητα του προστιθέμενου Ελληνικού σώματος κειμένων.

Η ανάπτυξη μοντέλων όπως το Meltemi, που βασίζονται σε υπάρχοντα ισχυρά μοντέλα ανοικτού κώδικα (όπως το Mistral-7B), αποτελεί μια στρατηγική επιλογή. Η δημιουργία ενός κορυφαίου LLM από το μηδέν είναι μια διαδικασία εξαιρετικά απαιτητική σε πόρους (δεδομένα, υπολογιστική ισχύ, τεχνογνωσία). Η αξιοποίηση υπαρχόντων μοντέλων θεμελίωσης (foundation models) παρέχει ένα σημαντικό προβάδισμα, καθώς αυτά τα μοντέλα διαθέτουν ήδη ισχυρές γενικές δυνατότητες κατανόησης και παραγωγής γλώσσας. Το κύριο έργο για τους ερευνητές επικεντρώνεται τότε στη συλλογή και επιμέλεια υψηλής ποιότητας Ελληνικών δεδομένων, στην πιθανή επέκταση του λεξιλογίου και στη συνεχή προεκπαίδευση ή τελειοποίηση για να ενσωματωθεί στο μοντέλο η Ελληνική γλωσσική επάρκεια και το πολιτισμικό πλαίσιο.



Σχήμα 2.1: Χαρακτηριστική φωτογραφία Meltemi

2.1.3 KriKri

Το Llama-Krikri-8B είναι ένα ακόμη σημαντικό μοντέλο που αναπτύχθηκε από το Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ). Βασίζεται στο Llama-3.1-8B, ένα μοντέλο 8 δισεκατομμυρίων παραμέτρων, και διατίθεται σε δύο εκδόσεις: τη βασική (Llama-Krikri-8B-Base) και μια έκδοση οδηγιών (Llama-Krikri-8B-Instruct). Ένα αξιοσημείωτο χαρακτηριστικό είναι το μήκος πλαισίου (context length) των 128k tokens, που αντιστοιχεί περίπου σε 80.000 Ελληνικές λέξεις [9].

Η εκπαίδευση του Krikri περιλάμβανε την επέκταση του λεξιλογίου του Llama-3.1 tokenizer με Ελληνικά tokens. Ακολούθησε συνεχής προεκπαίδευση (continual pretraining) σε ένα μεγάλο σώμα υψηλής ποιότητας και τοπικά συναφών Ελληνικών κειμένων, που αντιστοιχούσε σε 56.7 δισεκατομμύρια μονο γλωσσικά Ελληνικά tokens από δημόσια διαθέσιμους πόρους. Για τη διατήρηση των δίγλωσσων δυνατοτήτων (Ελληνικά-Αγγλικά) και την άμβλυνση του φαινομένου της καταστροφικής λήθης (catastrophic forgetting), χρησιμοποιήθηκαν επιπλέον υποσώματα με μονο γλωσσικά Αγγλικά κείμενα (21 δισεκατομμύρια tokens) και παράλληλα Ελληνο-Αγγλικά δεδομένα (5.5 δισεκατομμύρια tokens) [9]. Οι βασικές του δυνατότητες εστιάζουν στην ενισχυμένη επάρκεια για εργασίες στην Ελληνική γλώσσα, διατηρώντας παράλληλα τις δίγλωσσες ικανότητες. Η έκδοση instruct είναι σχεδιασμένη για καθοδηγούμενες αλληλεπιδράσεις. Το μοντέλο μπορεί να χρησιμοποιηθεί με τη βιβλιοθήκη Transformers και μέσω ενός εξυπηρετητή συμβατού με OpenAI μέσω του vLLM. Το Llama-Krikri-8B αναφέρεται επίσης σε πειράματα ταξινόμησης κειμένου.

Όσον αφορά τις επιδόσεις, αναφέρθηκαν βελτιώσεις σε σχέση με το Llama-3.1-8B: +10.8% σε Ελληνικά σημεία αναφοράς και +0.8% σε Αγγλικά σημεία αναφοράς. Το Llama-Krikri-8B είναι ένα ανοιχτό Ελληνικό LLM, διαθέσιμο στο Hugging Face (ilsp/Llama-Krikri-8B-Base, ilsp/Llama-Krikri-8B-Instruct). Ως προσαρμογή, η μέγιστη απόδοσή του ενδέχεται να επηρεάζεται

από το βασικό μοντέλο Llama-3.1-8B, και η ποιότητα των "τοπικά συναφών Ελληνικών κειμένων" που χρησιμοποιήθηκαν για την εκπαίδευση είναι καθοριστική.

Η ρητή αναφορά στην εκπαίδευση του Llama-Krikri με "τοπικά συναφή Ελληνικά κείμενα" και η λήψη μέτρων για την "άμβλυνση της καταστροφικής λήθης" μέσω της συμπερίληψης Αγγλικών και παράλληλων δεδομένων είναι ιδιαίτερα σημαντική[10]. Για να είναι ένα LLM πραγματικά χρήσιμο για μια συγκεκριμένη γλωσσική κοινότητα, πρέπει να κατανοεί όχι μόνο τη γλώσσα αλλά και το πολιτισμικό της πλαίσιο, την επικαιρότητα και τις κοινές γνώσεις που είναι συγκεκριμένες για την περιοχή. Τα "τοπικά συναφή κείμενα" στοχεύουν στην παροχή αυτής της διάστασης. Επιπλέον, κατά τη συνεχή προεκπαίδευση ενός μοντέλου σε μια νέα γλώσσα (Ελληνικά), υπάρχει ο κίνδυνος να "ξεχάσει" τις δυνατότητές του στην αρχική γλώσσα (Αγγλικά στην περίπτωση του Llama). Η συμπερίληψη ενός μείγματος μονο γλωσσικών Αγγλικών και Ελληνο-Αγγλικών παράλληλων δεδομένων κατά τη συνεχή προεκπαίδευση του Llama-Krikri είναι μια σκόπιμη στρατηγική για τη διατήρηση της διγλωσσίας του και την πρόληψη της υποβάθμισης της απόδοσης στα Αγγλικά. Αυτό υποδηλώνει μια εξελιγμένη κατανόηση της δυναμικής εκπαίδευσης των LLM και συνεπάγεται σημαντική προσπάθεια επιμέλειας δεδομένων για τον εντοπισμό και την επεξεργασία "τοπικά συναφών" κειμένων.

2.1.4 Συγκριτική Αξιολόγηση και Αποτίμηση των LLMs

Η συγκριτική αξιολόγηση των τριών ελληνικών γλωσσικών μοντέλων μεγάλης κλίμακας (LLMs) – Meltemi, Krikri και GreekBert – αποκαλύπτει μια ταχεία εξέλιξη στον τομέα της Επεξεργασίας Φυσικής Γλώσσας (NLP) για την ελληνική γλώσσα. Τα μοντέλα αυτά, προερχόμενα από διαφορετικές αρχιτεκτονικές και με διαφορετικούς στόχους, παρουσιάζουν ποικίλες επιδόσεις και δυνατότητες, αντανακλώντας την πρόοδο από θεμελιώδη μοντέλα κατανόησης φυσικής γλώσσας (NLU) προς πιο προηγμένα παραγωγικά και διαδραστικά συστήματα. Η αξιολόγησή τους βασίζεται σε ένα συνδυασμό γενικών και εξειδικευμένων ελληνικών συγκριτικών δοκιμασιών (benchmarks), καθώς και σε αγγλικές δοκιμασίες για την εκτίμηση των δίγλωσσων ικανοτήτων τους

Σε γενικές γραμμές, το Krikri αναδεικνύεται ως το πιο προηγμένο μοντέλο, επιδεικνύοντας κορυφαία απόδοση στις περισσότερες ελληνικές συγκριτικές δοκιμασίες, ιδιαίτερα σε εργασίες παρακολούθησης οδηγιών και διαλόγου, ξεπερνώντας σημαντικά το Meltemi. Αυτή η υπεροχή αποδίδεται στο μεγαλύτερο και πιο ποικιλόμορφο σύνολο δεδομένων εκπαίδευσης, στην πιο εξελιγμένη διαδικασία ευθυγράμμισης μετά την εκπαίδευση και στο σημαντικά διευρυμένο παράθυρο πλαισίου του. Το Meltemi, όντας το πρώτο ανοιχτό ελληνικό παραγωγικό LLM, προσφέρει μια αξιόλογη απόδοση για τα Νέα Ελληνικά και παραμένει μια καλή επιλογή για εφαρμογές που δεν απαιτούν τις κορυφαίες δυνατότητες του Krikri ή προτιμούν την πιο επιτρεπτική άδεια Apache 2.0. Από την άλλη πλευρά, το GreekBert, βασισμένο στην αρχιτεκτονική BERT, παραμένει ισχυρό σε εργασίες NLU όπως η ταξινόμηση κειμένου και η αναγνώριση ονοματικών οντοτήτων για τα Νέα Ελληνικά, όντας παράλληλα υπολογιστικά αποδοτικό. Ωστόσο, υπολείπεται σημαντικά σε παραγωγικές δυνατότητες και στην υποστήριξη ιστορικών παραλλαγών της ελληνικής γλώσσας σε σύγκριση με τα Meltemi και Krikri [11].

Όνομα Μοντέλου	Προγραμματιστής /Οργανισμός	Βασική Αρχιτεκτονική	Παράμετροι	Βασικά Ελληνικά Δεδομένα (Μέγεθος/Τύπος)	Κύριες Δυνατότητες στα Ελληνικά
Meltemi	ΙΕΛ/ΕΚ Αθηνά	Mistral-7B	7 δισ.	28.5 δισ. Ελληνικά tokens (δημόσια διαθέσιμοι πόροι)	Κατανόηση & παραγωγή κειμένου, συνομιλίες (Instruct έκδοση)
Llama-Krikri-8B	ΙΕΛ/ΕΚ Αθηνά	Llama-3.1-8B	8 δισ.	56.7 δισ. Ελληνικά tokens (δημόσια διαθέσιμοι πόροι, τοπικά συναφή κείμενα)	Βελτιωμένη κατανόηση Ελληνικών, δίγλωσσο (Ελληνικά-Αγγλικά)
GreekBERT	Ομάδα NLP ΟΠΑ	BERT-base	110 εκατ.	29GB Ελληνικού κειμένου (Wikipedia, Europarl, OSCAR)	Γενικές εργασίες NLP (POS tagging, NER, NLI)

Πίνακας 2.1: Σύνοψη Βασικών Ελληνικών LLMs

Η επιλογή του καταλληλότερου μοντέλου εξαρτάται σε μεγάλο βαθμό από τις ειδικές ανάγκες της εκάστοτε εφαρμογής, τους διαθέσιμους πόρους και τις απαιτήσεις αδειοδότησης. Για απαιτητικές εργασίες που περιλαμβάνουν μεγάλα έγγραφα, ιστορικά ελληνικά ή εξελιγμένους διαδραστικούς βοηθούς, το Krikri αποτελεί την προτιμώμενη λύση. Για γενικές εφαρμογές παραγωγής κειμένου στα Νέα Ελληνικά, το Meltemi προσφέρει μια ισορροπημένη επιλογή. Το GreekBert είναι ιδανικό για εξειδικευμένες εργασίες NLU όπου η αποδοτικότητα είναι κρίσιμη. Η εξέλιξη αυτή υπογραμμίζει την αυξανόμενη ωριμότητα της ελληνικής κοινότητας NLP και προδιαγράφει ένα μέλλον με ακόμα πιο εξειδικευμένα και ικανά μοντέλα για την ελληνική γλώσσα.

Μοντέλο	Medical MCQA EL (15-shot)	Belebele EL (5-shot)	HellaSwag EL (10-shot)	ARC-C EL (25-shot)	MMLU EL (5-shot)	XNLI Greek (Acc)	GUDT PoS (Acc)	NER (F1)
Meltemi-7B-v1.5-Base	48.1%	68.6%	65.7%	47.1%	42.4%	-	-	-
Llama-Krikri-8B-Base	53.8%	82.7%	64.6%	49.4%	52.0%	-	-	-
GreekBert	-	-	-	-	-	78.6%	98.1%	85.7%

Πίνακας 2.2: Απόδοση σε Βασικές Ελληνικές Συγκριτικές Δοκιμασίες NLU (Βασικά Μοντέλα & GreekBert)

2.2 Τεχνικές NLP για Ελληνικά

Η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP) είναι ένας κλάδος της Τεχνητής Νοημοσύνης που ασχολείται με την αλληλεπίδραση μεταξύ υπολογιστών και ανθρώπινης γλώσσας. Για την αποτελεσματική χρήση των LLMs και την προετοιμασία των ελληνικών δημοσίων εγγράφων για το σύστημα ανάκτησης, απαιτείται η εφαρμογή μιας σειράς τεχνικών NLP.

Η προεπεξεργασία (preprocessing) του κειμένου είναι ένα θεμελιώδες στάδιο. Περιλαμβάνει:

- **Λεξιλογική Ανάλυση/Τμηματοποίηση (Tokenization)**: Η διαδικασία διαχωρισμού του κειμένου σε μικρότερες μονάδες, όπως λέξεις ή υπο-λέξεις (tokens). Αυτό είναι απαραίτητο για την περαιτέρω επεξεργασία από τα γλωσσικά μοντέλα.
- **Αφαίρεση Άχρηστων Λέξεων (Stop-word removal)**: Η αφαίρεση κοινών λέξεων (π.χ., "και", "ο", "η", "το", "αλλά") που δεν προσφέρουν σημαντική σημασιολογική πληροφορία. Μια εκτενής λίστα ελληνικών stopwords χρησιμοποιείται στον κώδικα.
- **Μορφολογική Ομαλοποίηση (Stemming/Lemmatization)**: Η διαδικασία μετατροπής των λέξεων στη βασική ή ριζική τους μορφή. Ο κώδικας περιλαμβάνει μια συνάρτηση simple_greek_stemmer, που αποτελεί μια απλοϊκή προσέγγιση μορφολογικής ομαλοποίησης για την ελληνική γλώσσα, αφαιρώντας γνωστές καταλήξεις.

Η ελληνική γλώσσα, με την πλούσια μορφολογία της (πολυτυπία, κλίσεις ουσιαστικών, επιθέτων, ρημάτων) και την σχετικά ελεύθερη σύνταξη, παρουσιάζει ιδιαίτερες προκλήσεις για τις τεχνικές NLP. Η διαχείριση των διακριτικών (τόνοι, πνεύματα – αν και τα πνεύματα δεν χρησιμοποιούνται στη

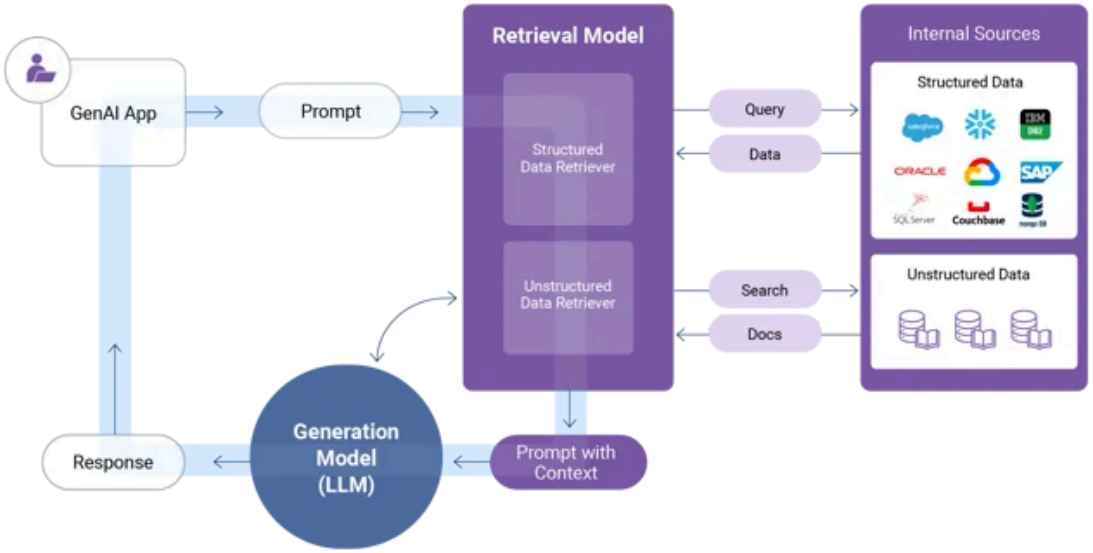
σύγχρονη ελληνική γραφή δημοσίων εγγράφων) και η αντιμετώπιση της πολυσημίας απαιτούν εξειδικευμένες προσεγγίσεις ή μοντέλα που έχουν εκπαιδευτεί επαρκώς σε ελληνικό κείμενο. Η περιορισμένη διαθεσιμότητα μεγάλων, σχολιασμένων συνόλων δεδομένων για την ελληνική γλώσσα, σε σύγκριση με την αγγλική, αποτελεί επίσης έναν παράγοντα που επηρεάζει την ανάπτυξη και την απόδοση των εργαλείων NLP.

2.3 Vector Embeddings & μέτρα ομοιότητας και RAG

Για την αποτελεσματική σημασιολογική αναζήτηση, το κείμενο πρέπει να μετατραπεί σε μια μορφή που οι υπολογιστές μπορούν να επεξεργαστούν αριθμητικά, διατηρώντας παράλληλα τη σημασιολογική του ουσία.

- **Διανυσματικές Αναπαραστάσεις (Vector Embeddings)**: Τα vector embeddings είναι πυκνές διανυσματικές αναπαραστάσεις λέξεων, φράσεων ή ολόκληρων εγγράφων σε έναν πολυδιάστατο χώρο. Σε αυτόν τον χώρο, κείμενα με παρόμοια σημασία τείνουν να βρίσκονται πιο κοντά το ένα στο άλλο. Τα σύγχρονα LLMs, όπως αυτά που χρησιμοποιούνται στην παρούσα εργασία, είναι ιδιαίτερα αποτελεσματικά στην παραγωγή τέτοιων embeddings. Για παράδειγμα, η τεχνική mean pooling, που αναφέρεται στον κώδικα μας για το Meltemi, λαμβάνει τις εξόδους των κρυφών επιπέδων του μοντέλου για κάθε token στην είσοδο και υπολογίζει τον μέσο όρο τους για να δημιουργήσει ένα μοναδικό διάνυσμα που αναπαριστά ολόκληρο το απόσπασμα κειμένου [12].
- **Μέτρα Ομοιότητας (Similarity Measures)**: Αφού τα κείμενα (τόσο τα αποθηκευμένα έγγραφα όσο και τα ερωτήματα των χρηστών) μετατραπούν σε embeddings, χρειάζεται ένας τρόπος για να μετρηθεί η μεταξύ τους ομοιότητα. Το πιο διαδεδομένο μέτρο για αυτόν τον σκοπό είναι η Ομοιότητα Συνημιτόνου (Cosine Similarity). Αυτή υπολογίζει το συνημίτονο της γωνίας μεταξύ δύο διανυσμάτων, με τιμές που κυμαίνονται από -1 (εντελώς αντίθετα) έως 1 (εντελώς όμοια), ενώ το 0 υποδηλώνει ορθογωνιότητα (έλλειψη ομοιότητας). Μια τιμή κοντά στο 1 υποδηλώνει υψηλή σημασιολογική ομοιότητα [13].
- **Retrieval-Augmented Generation (RAG)**: Η αρχιτεκτονική RAG συνδυάζει τις δυνατότητες των προεκπαιδευμένων LLMs με έναν μηχανισμό ανάκτησης πληροφοριών από μια εξωτερική βάση γνώσεων. Στο πλαίσιο αυτής της εργασίας, η διαδικασία έχει ως εξής:
    1. Τα προ επεξεργασμένα τμήματα των δημοσίων εγγράφων μετατρέπονται σε embeddings και αποθηκεύονται σε μια διανυσματική βάση δεδομένων(π.χ., ChromaDB, όπως χρησιμοποιείται στον κώδικα).
    2. Όταν ο χρήστης υποβάλλει ένα ερώτημα, το ερώτημα μετατρέπεται επίσης σε embedding.
    3. Το σύστημα αναζητά στη διανυσματική βάση δεδομένων τα αποθηκευμένα embeddings των εγγράφων που είναι πιο "όμοια" (με βάση την ομοιότητα συνημιτόνου) με το embedding του ερωτήματος.
    4. Τα πιο σχετικά τμήματα κειμένου (chunks) που ανακτώνται, χρησιμοποιούνται για να "αυξήσουν" το αρχικό ερώτημα που δίνεται στο LLM. Δηλαδή, το LLM λαμβάνει τόσο το ερώτημα του χρήστη όσο και τα σχετικά αποσπάσματα ως πλαίσιο (context).
    5. Βάσει αυτού του επαυξημένου πλαισίου, το LLM (στην περίπτωση αυτή, κυρίως για την κατανόηση και την καθοδήγηση της ανάκτησης, καθώς η εργασία εστιάζει στην επιστροφή εγγράφων) βοηθά στην παρουσίαση των πιο σχετικών εγγράφων [14]. Η

προσέγγιση RAG είναι ιδιαίτερα χρήσιμη καθώς επιτρέπει στα LLMs να αξιοποιούν εξειδικευμένες ή επικαιροποιημένες πληροφορίες που δεν περιλαμβάνονταν στα αρχικά τους δεδομένα εκπαίδευσης, μειώνοντας τον κίνδυνο παραγωγής λανθασμένων ή μη σχετικών απαντήσεων (hallucinations) και αυξάνοντας την ακρίβεια και τη συνάφεια [15].



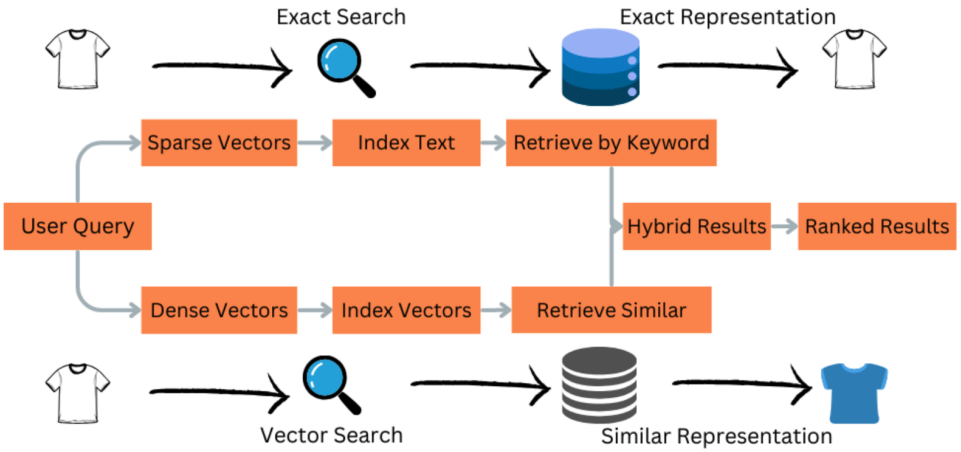
Σχήμα 2.2: Retrieval Augmented Generation (RAG) σε μορφή

2.4 Υβριδικά πληροφοριακά συστήματα αναζήτησης

Τα υβριδικά συστήματα αναζήτησης (Hybrid Search Systems) συνδυάζουν πολλαπλές τεχνικές ανάκτησης πληροφοριών με στόχο την επίτευξη καλύτερης απόδοσης από ό,τι θα μπορούσε να επιτευχθεί με τη χρήση μιας μεμονωμένης μεθόδου. Στο πλαίσιο της παρούσας εργασίας, και όπως χρησιμοποιήθηκε στον κώδικα εφαρμογής, υιοθετείται μια υβριδική προσέγγιση που συνδυάζει:

- **Σημασιολογική Αναζήτηση (Semantic Search)**: Βασίζεται στα vector embeddings που παράγονται από τα LLMs (GreekBERT, Meltemi, KriKri) και στην ομοιότητα συνημιτόνου για τον εντοπισμό τμημάτων κειμένου που είναι σημασιολογικά συναφή με το ερώτημα του χρήστη, ακόμα και αν δεν περιέχουν ακριβώς τις ίδιες λέξεις.
- **Λεξική Αναζήτηση (Lexical Search)**: Βασίζεται σε παραδοσιακές τεχνικές αντιστοίχισης λέξεων-κλειδιών. Ο κώδικας χρησιμοποιεί HashingVectorizer για τη δημιουργία αναπαραστάσεων TF-IDF (Term Frequency-Inverse Document Frequency) ή παρόμοιων, τόσο σε επίπεδο χαρακτήρων (character n-grams) όσο και σε επίπεδο λέξεων (word n-grams). Αυτή η προσέγγιση είναι αποτελεσματική στον εντοπισμό εγγράφων που περιέχουν ακριβείς όρους ή φράσεις από το ερώτημα.

Η υβριδική φύση του συστήματος επιτυγχάνεται με τον συνδυασμό των βαθμολογιών (scores) από τις δύο αυτές μεθόδους. Ο κώδικας αναφέρει παραμέτρους ALPHA_BASE και ALPHA_LONGQ που χρησιμοποιούνται για τη στάθμιση της σημασίας κάθε τύπου αναζήτησης, δίνοντας διαφορετική βαρύτητα ανάλογα με το μήκος ή τη φύση του ερωτήματος. Επιπλέον, συχνά εφαρμόζεται ενίσχυση (boosting) για τα αποτελέσματα που προκύπτουν από ακριβείς αντιστοιχίες (exact matches).



Σχήμα 2.3: Μια προσέγγιση της Hybrid Search

2.5 Υφιστάμενα έργα διαφάνειας (ΔΙΑΥΓΕΙΑ, ΕΚΤ e-repository κ.λπ.)

Η διαφάνεια στη δημόσια διοίκηση και η ανοικτή πρόσβαση στην πληροφορία αποτελούν θεμελιώδεις αρχές της σύγχρονης διακυβέρνησης. Στην Ελλάδα, έχουν αναπτυχθεί σημαντικές πρωτοβουλίες και πλατφόρμες για την προώθηση αυτών των αρχών.

Πρόγραμμα "ΔΙΑΥΓΕΙΑ": Η "ΔΙΑΥΓΕΙΑ" (diavgeia.gov.gr) αποτελεί την κεντρική πρωτοβουλία για την ανάρτηση και δημοσιοποίηση όλων των πράξεων και αποφάσεων των κυβερνητικών και διοικητικών οργάνων στο διαδίκτυο. Ο σκοπός της είναι η ενίσχυση της διαφάνειας, της λογοδοσίας και της καταπολέμησης της διαφθοράς. Στη "ΔΙΑΎΓΕΙΑ" αναρτάται ένα τεράστιος όγκος εγγράφων, συμπεριλαμβανομένων νόμων, προεδρικών διαταγμάτων, υπουργικών αποφάσεων, προκηρύξεων (όπως αυτές που αποτελούν αντικείμενο της παρούσας εργασίας), συμβάσεων, δαπανών κ.ά. Η φύση αυτών των εγγράφων (κυρίως PDF) και ο μεγάλος τους αριθμός καθιστούν την πλατφόρμα μια πλούσια, αλλά και απαιτητική ως προς την αναζήτηση, πηγή δεδομένων.

Εθνικό Κέντρο Τεκμηρίωσης και Ηλεκτρονικού Περιεχομένου (ΕΚΤ): Το ΕΚΤ (ekt.gr) διαδραματίζει κεντρικό ρόλο στη συλλογή, οργάνωση, διάχυση και διατήρηση της επιστημονικής, ερευνητικής και πολιτιστικής παραγωγής της Ελλάδας. Το ηλεκτρονικό αποθετήριο του (e-repository) περιλαμβάνει δημοσιεύσεις, διδακτορικές διατριβές, ερευνητικά δεδομένα και άλλο ψηφιακό περιεχόμενο. Αν και ο τύπος των εγγράφων διαφέρει από αυτόν της "ΔΙΑΥΓΕΙΑΣ", το ΕΚΤ αποτελεί παράδειγμα μιας μεγάλης, δομημένης συλλογής ελληνικού περιεχομένου που απαιτεί αποτελεσματικά εργαλεία αναζήτησης.

Άλλες πιθανές πηγές δημοσίων δεδομένων μπορεί να περιλαμβάνουν τις ιστοσελίδες μεμονωμένων υπουργείων, δημοσίων οργανισμών, και ακαδημαϊκών ιδρυμάτων, τα οποία συχνά δημοσιεύουν κανονισμούς, εκθέσεις, και άλλα έγγραφα πολιτικής.

Η ύπαρξη αυτών των έργων διαφάνειας και των αποθετηρίων παρέχει μια πολύτιμη πρώτη ύλη για την ανάπτυξη συστημάτων ανάκτησης πληροφοριών όπως. Ταυτόχρονα, ο όγκος, η ποικιλομορφία και η συχνά μη πλήρως δομημένη φύση των δεδομένων αυτών υπογραμμίζουν την ανάγκη για ευφυείς τεχνολογίες αναζήτησης που να μπορούν να εξάγουν τις ζητούμενες πληροφορίες με ακρίβεια και ταχύτητα ως προς τον τελικό χρήστη και πολίτη..

2.6 Επίλογος

Στο παρόν κεφάλαιο πραγματοποιήθηκε μια επισκόπηση των βασικών θεωρητικών εννοιών και τεχνολογιών που αποτελούν τον πυρήνα της παρούσας διπλωματικής εργασίας. Αναλύθηκαν τα Μεγάλα Γλωσσικά Μοντέλα με έμφαση στα ελληνικά μοντέλα GreekBERT, Meltemi και KriKri, οι θεμελιώδεις τεχνικές Επεξεργασίας Φυσικής Γλώσσας που απαιτούνται για την προετοιμασία των ελληνικών κειμένων, καθώς και η σημασία των διανυσματικών αναπαραστάσεων και των μέτρων ομοιότητας. Ιδιαίτερη έμφαση δόθηκε στην αρχιτεκτονική RAG και στα υβριδικά συστήματα αναζήτησης, τα οποία συνδυάζουν λεξικές και σημασιολογικές μεθόδους για βελτιστοποιημένη ανάκτηση. Τέλος, εξετάστηκαν τα υφιστάμενα έργα διαφάνειας, όπως η "ΔΙΑΥΓΕΙΑ", που παρέχουν το πλαίσιο και τα δεδομένα για την ανάπτυξη του προτεινόμενου συστήματος. Οι γνώσεις που παρουσιάστηκαν θέτουν τις βάσεις για την κατανόηση του σχεδιασμού, της υλοποίησης και της αξιολόγησης του διαλογικού βοηθού που θα αναπτυχθεί στα επόμενα κεφάλαια.

Κεφάλαιο 3ο: Συλλογή Δεδομένων & Κατασκευή Dataset

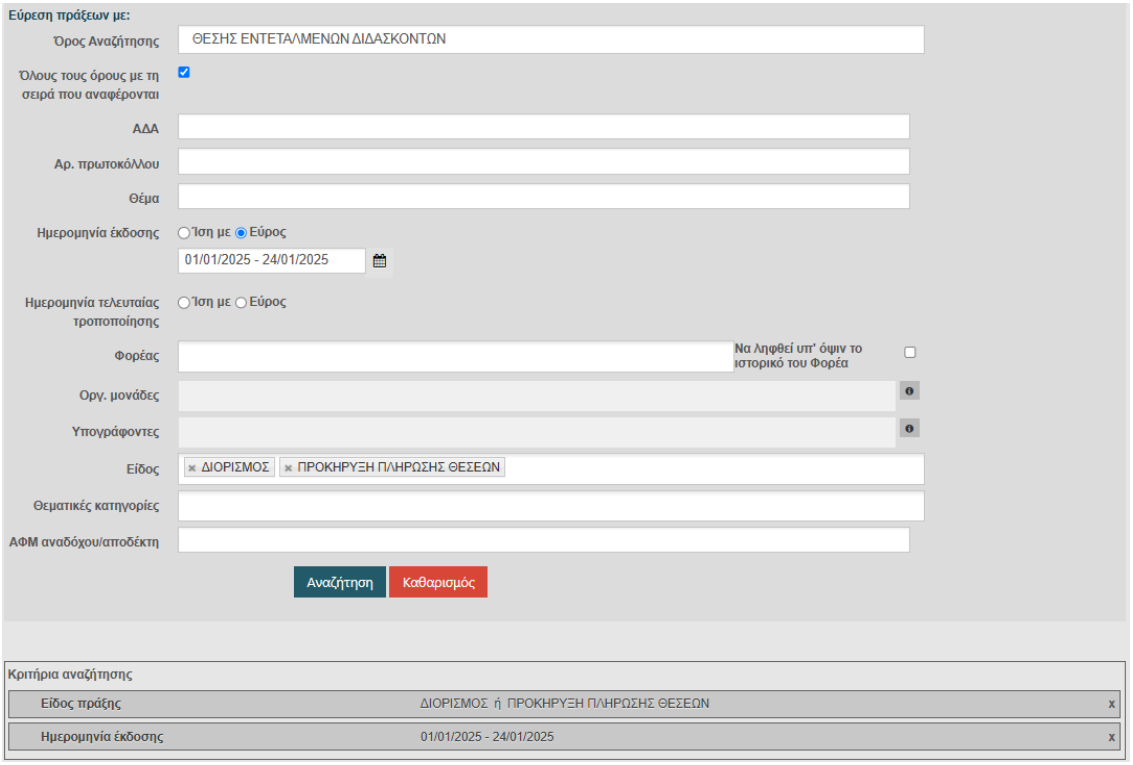
3.1 Πηγή ΔΙΑΥΓΕΙΑ

Η επιτυχής ανάπτυξη ενός συστήματος ανάκτησης πληροφοριών βασισμένου σε Μεγάλα Γλωσσικά Μοντέλα εξαρτάται άμεσα από την ποιότητα και τη συνάφεια του συνόλου δεδομένων (dataset) στο οποίο θα εκπαιδευτεί και θα αξιολογηθεί. Το παρόν κεφάλαιο περιγράφει λεπτομερώς τη διαδικασία που ακολουθήθηκε για τη συλλογή των πρωτογενών δεδομένων, την αυτοματοποιημένη λήψη των σχετικών αρχείων, την προεπεξεργασία τους μέσω εξειδικευμένου κώδικα, την αποθήκευσή τους, καθώς και την αρχική ανάλυση των χαρακτηριστικών τους.

Ως κύρια πηγή για τη συλλογή των εγγράφων επιλέχθηκε η διαδικτυακή πύλη "ΔΙΑΥΓΕΙΑ". Η πλατφόρμα αυτή αποτελεί κεντρικό σημείο ανάρτησης πράξεων των ελληνικών κυβερνητικών και διοικητικών οργάνων, προσφέροντας ένα ευρύ φάσμα δημοσίων εγγράφων. Αρχικά, διεξήχθη διερευνητική έρευνα στην πλατφόρμα για τον εντοπισμό διαφόρων τύπων εγγράφων και πιθανών περιπτώσεων χρήσης (use cases) που θα μπορούσαν να αξιοποιηθούν.

Κατά τη διερεύνηση αυτή, παρατηρήθηκε ότι πολλές φορές οι αρχικές αναζητήσεις επέστρεφαν σημαντικό αριθμό μη σχετικών αρχείων, καθιστώντας αναγκαία τη χρήση πιο εξειδικευμένων όρων και φίλτρων. Για τους σκοπούς της παρούσας διπλωματικής εργασίας, αποφασίστηκε η εστίαση στις προκηρύξεις για θέσεις εντεταλμένων διδασκόντων σε Πανεπιστήμια. Ο τελικός όρος αναζήτησης που χρησιμοποιήθηκε ήταν "ΘΕΣΗΣ ΕΝΤΕΤΑΛΜΕΝΩΝ ΔΙΔΑΣΚΟΝΤΩΝ". Για την περαιτέρω εξειδίκευση των αποτελεσμάτων, εφαρμόστηκαν τα φίλτρα "Διορισμός" και "Προκήρυξη Πλήρωσης Θέσεων". Μια πρώτη επισκόπηση των επιστρεφόμενων αρχείων έδειξε ότι η συντριπτική πλειοψηφία τους ήταν σε μορφή PDF και, σημαντικό για την επεξεργασία, το περιεχόμενό τους ήταν αναγνωρίσιμο ως κείμενο (searchable PDFs), εξαλείφοντας την ανάγκη για τεχνικές Οπτικής Αναγνώρισης Χαρακτήρων (OCR) σε αυτό το στάδιο.

Παρακάτω γίνεται σχετική απεικόνιση της αναζήτησης στην πλατφόρμα της Διαύγειας.



Σχήμα 3.1: Αναζήτηση στην Διαύγεια

3.1.1 Κριτήρια Επιλογής και Όγκος Δεδομένων

Για τον καθορισμό του τελικού συνόλου δεδομένων, τέθηκε ως βασικό χρονικό κριτήριο η συλλογή πρόσφατων και επομένως επίκαιρων αρχείων, ώστε να διασφαλιστεί η επικαιρότητα και η αξιοπιστία των πληροφοριών που θα χρησιμοποιηθούν. Πιο συγκεκριμένα, η περίοδος αναζήτησης περιορίστηκε αυστηρά στο χρονικό διάστημα από 1 Ιανουαρίου 2025 έως 24 Ιανουαρίου 2025, διασφαλίζοντας ότι το σύνολο δεδομένων αντιστοιχεί σε σαφώς προσδιορισμένη και περιορισμένη χρονική περίοδο, ενισχύοντας έτσι την αξιοπιστία των ευρημάτων.

Από την εφαρμογή των παραπάνω κριτηρίων, που περιλάμβαναν συγκεκριμένο όρο αναζήτησης, πολλαπλά φίλτρα ποιότητας και σχετικότητας, καθώς και τον περιορισμό στην προαναφερθείσα χρονική περίοδο, προέκυψε ένα σύνολο 114 αρχείων PDF. Το μέγεθος αυτό του δείγματος κρίθηκε ιδιαίτερα κατάλληλο για τις ανάγκες της διπλωματικής εργασίας. Ειδικότερα, ο αριθμός αυτός είναι αρκετά μεγάλος ώστε να προσφέρει ικανοποιητική αντιπροσωπευτικότητα του θέματος, ενισχύοντας τη δυνατότητα γενίκευσης και την εξαγωγή ισχυρών συμπερασμάτων από την εκπαίδευση και αξιολόγηση των γλωσσικών μοντέλων.

Παράλληλα, το πλήθος των αρχείων διατηρείται σε ένα εύρος που είναι διαχειρίσιμο από υπολογιστική άποψη, γεγονός που επιτρέπει αποτελεσματική επεξεργασία και αναλυτική αξιολόγηση χωρίς την ανάγκη υπερβολικών υπολογιστικών πόρων ή χρονοβόρων διαδικασιών. Η ισορροπία αυτή είναι ιδιαίτερα σημαντική, καθώς διασφαλίζει τόσο τη δυνατότητα για σε βάθος ανάλυση, όσο και την πρακτικότητα της υλοποίησης του έργου. Έτσι, το τελικό σύνολο των 114 αρχείων PDF αποτελεί μια

ιδανική βάση για την επιτυχημένη πραγματοποίηση της μελέτης, συνδυάζοντας βέλτιστα την ποιότητα και την ποσότητα των δεδομένων.

3.2 Αυτοματοποιημένη λήψη αρχείων

Η πλατφόρμα "ΔΙΑΥΓΕΙΑ" παρέχει τη δυνατότητα εξαγωγής των αποτελεσμάτων μιας αναζήτησης σε μορφή αρχείου XLS (Excel). Το αρχείο αυτό περιέχει δομημένες πληροφορίες για κάθε ανάρτηση, όπως:

- **ΑΔΑ (Αριθμός Διαδικτυακής Ανάρτησης)**: Ο μοναδικός κωδικός αναγνώρισης κάθε πράξης.
- **Ημ/νια Ανάρτησης**: Η ημερομηνία δημοσίευσης της πράξης στη ΔΙΑΥΓΕΙΑ.
- **Θέμα**: Ο τίτλος ή η περιγραφή της πράξης.
- **Φορέας ID**: Ο μοναδικός κωδικός του δημόσιου φορέα.
- **Φορέας**: Η ονομασία του δημόσιου φορέα που εξέδωσε την πράξη.
- **Τύπος Πράξης ID**: Ο κωδικός του τύπου της πράξης.
- **Τύπος Πράξης**: Η περιγραφή του τύπου της πράξης (π.χ., Προκήρυξη Πλήρωσης Θέσεων).
- **URL Εγγράφου**: Ο απευθείας σύνδεσμος (URL) προς το αρχείο PDF της πράξης.
- **Αρ. Πρωτοκόλλου**: Ο αριθμός πρωτοκόλλου του εγγράφου.
- **Ημ/νια Πράξης**: Η ημερομηνία έκδοσης της πράξης.

Από τις παραπάνω πληροφορίες, η στήλη "URL Εγγράφου" ήταν κρίσιμης σημασίας, καθώς περιείχε τους απευθείας συνδέσμους για τη λήψη των αρχείων PDF. Οι σύνδεσμοι αυτοί εξήχθησαν από το αρχείο XLS και αποθηκεύτηκαν σε ένα απλό αρχείο κειμένου (114PDFS.txt), κάθε σύνδεσμος σε μια νέα γραμμή.

Για την αυτοματοποιημένη και μαζική λήψη των 114 αρχείων PDF, χρησιμοποιήθηκε το εργαλείο γραμμής εντολών wget με το ακόλουθο script:

```
C:\Tools\wget.exe ^
  --user-agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36" ^
  --no-iri ^
  --local-encoding=UTF-8 ^
  -i "C:\Users\chris\114PDFS.txt" ^
  -P "C:\Users\chris\Desktop\LocalFile"
```

Σχήμα 3.2: Script Κώδικας

Αναλυτικότερα, το script αυτό:

- Καλεί το εκτελέσιμο αρχείο wget.exe (που βρίσκεται στον φάκελο C:\Tools\).
- Ορίζει ένα user-agent που προσομοιώνει έναν περιηγητή ιστού (Chrome σε Windows), ώστε να αποφευχθούν πιθανοί περιορισμοί από τον εξυπηρετητή της ΔΙΑΥΓΕΙΑΣ.
- Η παράμετρος --no-iri διασφαλίζει τη σωστή διαχείριση των Διεθνοποιημένων Αναγνωριστικών Πόρων (IRIs) που ενδέχεται να περιέχουν μη-ASCII χαρακτήρες στους συνδέσμους.
- Η παράμετρος --local-encoding=UTF-8 ορίζει την κωδικοποίηση που θα χρησιμοποιηθεί για τα τοπικά ονόματα αρχείων.
- Η παράμετρος -i C:\Users\chris\114PDFS.txt καθοδηγεί το wget να διαβάσει τους συνδέσμους λήψης από το αρχείο 114PDFS.txt.

- Τέλος, η παράμετρος -P C:\Users\chris\Desktop\LocalFile ορίζει τον κατάλογο LocalFile στην επιφάνεια εργασίας ως τον προορισμό όπου θα αποθηκευτούν τα ληφθέντα αρχεία PDF.

Με αυτόν τον τρόπο, επιτεύχθηκε η ταχεία και αποτελεσματική συλλογή όλων των απαραίτητων αρχείων τοπικά στον υπολογιστή για την περαιτέρω επεξεργασία τους.

3.3 Αποθήκευση στο GDrive

Μετά την τοπική λήψη των 114 αρχείων PDF και πριν την επεξεργασία τους με το Google Colab script, τα αρχεία μεταφορτώθηκαν σε έναν ειδικό φάκελο στο Google Drive. Η επιλογή αυτή έγινε για διάφορους πρακτικούς λόγους:

- **Προσβασιμότητα από το Google Colab**: Το Google Colaboratory ενσωματώνεται άψογα με το Google Drive, επιτρέποντας στα scripts που εκτελούνται στο Colab να έχουν εύκολη και γρήγορη πρόσβαση στα αρχεία που είναι αποθηκευμένα στο Drive. Αυτό απλοποιεί σημαντικά τη διαδικασία εισαγωγής δεδομένων στο περιβάλλον επεξεργασίας.
- **Δημιουργία Αντιγράφων Ασφαλείας (Backup)**: Η αποθήκευση των πρωτογενών δεδομένων στο Google Drive λειτουργεί και ως επιπλέον μέτρο δημιουργίας αντιγράφων ασφαλείας, προστατεύοντας από ενδεχόμενη απώλεια των τοπικών αρχείων.
- **Οργάνωση και Διαχείριση**: Το Google Drive παρέχει ένα δομημένο περιβάλλον για την οργάνωση των αρχείων και των φακέλων του έργου, διευκολύνοντας την παρακολούθηση και διαχείρισή τους.

Το τελικό επεξεργασμένο dataset σε μορφή JSON (dataset14.json) αποθηκεύτηκε επίσης στο Google Drive, στον ίδιο ή σε παρακείμενο φάκελο, για τους ίδιους λόγους προσβασιμότητας και διαχείρισης, έτοιμο να χρησιμοποιηθεί από τα scripts δοκιμών και αξιολόγησης των μοντέλων στην πλατφόρμα Hugging Face Spaces.

3.4 Google Colab

Για την υλοποίηση της απαιτητικής διαδικασίας επεξεργασίας των αρχείων PDF και την κατασκευή του τελικού δομημένου dataset, επιλέχθηκε η πλατφόρμα Google Colaboratory (Colab). Το Google Colab είναι ένα δωρεάν, cloud-based περιβάλλον εκτέλεσης Jupyter notebooks, το οποίο παρέχεται από την Google Research. Επιτρέπει τη συγγραφή και εκτέλεση κώδικα Python απευθείας στον περιηγητή, χωρίς να απαιτείται η εγκατάσταση ή παραμετροποίηση εξειδικευμένου λογισμικού στον τοπικό υπολογιστή του χρήστη.Η επιλογή του Google Colab για την παρούσα διπλωματική εργασία βασίστηκε σε μια σειρά από σημαντικά πλεονεκτήματα που προσφέρει:

1. **Προσβασιμότητα και Οικονομική Αποδοτικότητα**: Το Colab παρέχει δωρεάν πρόσβαση σε υπολογιστικούς πόρους, συμπεριλαμβανομένης της χρήσης CPU και, υπό προϋποθέσεις διαθεσιμότητας, GPU ή TPU. Αυτό καθιστά εφικτή την επεξεργασία δεδομένων και την εκτέλεση υπολογιστικά έντονων εργασιών χωρίς την ανάγκη για επένδυση σε ακριβό τοπικό εξοπλισμό.
2. **Προεγκατεστημένες Βιβλιοθήκες και Ευκολία Διαχείρισης Πακέτων**: Πολλές από τις ευρέως χρησιμοποιούμενες βιβλιοθήκες Python για την επιστήμη δεδομένων και τη μηχανική μάθηση (π.χ., NumPy, Pandas) είναι ήδη προεγκατεστημένες. Επιπλέον, η εγκατάσταση νέων βιβλιοθηκών, όπως η PyPDF2 που χρησιμοποιήθηκε εκτενώς στην παρούσα εργασία, γίνεται εύκολα μέσω της εντολής !pip install.

3. **Άμεση Ενσωμάτωση με το Google Drive**: Το Colab επιτρέπει την απρόσκοπτη προσάρτηση (mount) του προσωπικού Google Drive του χρήστη. Αυτή η λειτουργία αξιοποιήθηκε πλήρως για την ανάγνωση των αρχείων PDF που είχαν μεταφορτωθεί (όπως περιγράφεται στην ενότητα 3.3) καθώς και για την αποθήκευση του τελικού παραγόμενου αρχείου JSON του dataset.
4. **Διαδραστικό Περιβάλλον Ανάπτυξης**: Η δομή των Jupyter notebooks επιτρέπει την εκτέλεση κώδικα σε διακριτές κυψέλες (cells), διευκολύνοντας την τμηματική ανάπτυξη, τον έλεγχο και την αποσφαλμάτωση (debugging) του κώδικα. Αυτό ήταν ιδιαίτερα χρήσιμο κατά την ανάπτυξη των πολλαπλών συναρτήσεων προεπεξεργασίας.

Στο περιβάλλον του Google Colab αναπτύχθηκε και εκτελέστηκε ένα εξειδικευμένο script στην Python, με σκοπό τη μετατροπή ακατέργαστων δεδομένων από 114 αρχεία PDF σε ένα καθαρό και δομημένο dataset. Το script πραγματοποίησε εξαγωγή και προ επεξεργασία του κειμένου, αφαιρώντας στοιχεία όπως επικεφαλίδες, υποσέλιδα, υπερσυνδέσμους και τυποποιημένες ενότητες, ενώ παράλληλα προστάτευε και επανέφερε σημαντικές πληροφορίες, όπως ημερομηνίες. Η διαδικασία καθαρισμού περιελάμβανε επίσης ομογενοποίηση κειμένου, αφαίρεση μη πληροφοριακών λέξεων (stopwords) και εφαρμογή απλών μορφολογικών τεχνικών για την ελληνική γλώσσα [16] .

Η ευελιξία και η υπολογιστική δύναμη του Google Colab επέτρεψε την αποτελεσματική εφαρμογή της διαδικασίας αυτής σε όλα τα αρχεία PDF, συγκεντρώνοντας τα αποτελέσματα σε ένα ενιαίο JSON αρχείο, το οποίο αποθηκεύτηκε στο Google Drive. Η χρήση του Colab αποτέλεσε καθοριστικό παράγοντα για την ομαλή και γρήγορη υλοποίηση της προετοιμασίας των δεδομένων, καθιστώντας τα έτοιμα για περαιτέρω επεξεργασία και ανάλυση.

3.5 Έρευνα στα PDFs και Dataset

Πριν και κατά τη διάρκεια της αυτοματοποιημένης προεπεξεργασίας, πραγματοποιήθηκε μια ποιοτική έρευνα στα περιεχόμενα των 114 ληφθέντων αρχείων PDF. Αυτή η επισκόπηση ήταν σημαντική για την κατανόηση της φύσης των δεδομένων και τον εντοπισμό πιθανών προκλήσεων.

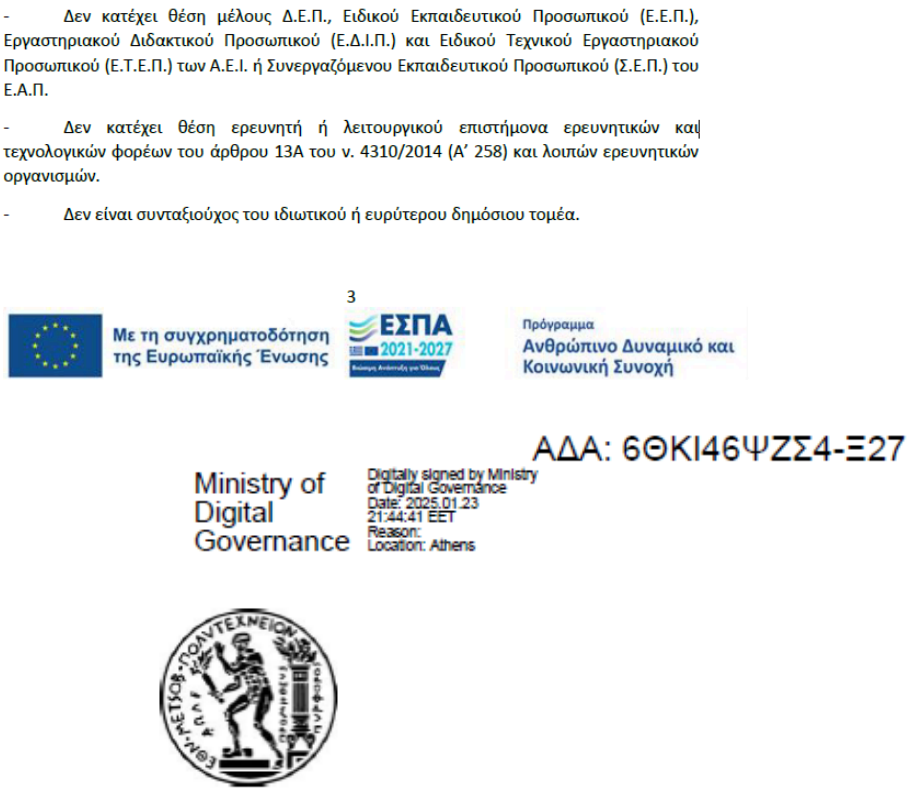
Βασικά ευρήματα αυτής της έρευνας περιλαμβάνουν:

- **Δυνατότητα Αναζήτησης (Searchability)**: Επιβεβαιώθηκε ότι όλα τα εξετασθέντα αρχεία PDF ήταν "searchable", δηλαδή το κείμενό τους ήταν ψηφιακά αναγνωρίσιμο και μπορούσε να εξαχθεί χωρίς την ανάγκη εφαρμογής τεχνικών Οπτικής Αναγνώρισης Χαρακτήρων (OCR). Αυτό απλοποίησε σημαντικά το στάδιο της εξαγωγής κειμένου.
- **Μορφολογικές Διαφοροποιήσεις**: Παρατηρήθηκε ανομοιογένεια στη δομή και τη μορφοποίηση των προκηρύξεων. Αν και ακολουθούν ένα γενικό πρότυπο, υπήρχαν διαφορές στη διάταξη, στη χρήση γραμματοσειρών και στην οργάνωση των ενοτήτων μεταξύ εγγράφων από διαφορετικά ακαδημαϊκά ιδρύματα ή ακόμα και από διαφορετικές χρονικές περιόδους. Αυτό ενίσχυσε την ανάγκη για ευέλικτες τεχνικές προεπεξεργασίας.
- **Παρουσία Νομικής Ορολογίας**: Όπως ήταν αναμενόμενο, οι προκηρύξεις περιείχαν εκτενή χρήση εξειδικευμένης νομικής και διοικητικής ορολογίας, κάτι που έπρεπε να ληφθεί υπόψη κατά την αξιολόγηση της κατανόησης των γλωσσικών μοντέλων.
- **Πίνακες και Δομημένα Δεδομένα**: Πολλά έγγραφα περιείχαν πίνακες με πληροφορίες όπως απαιτούμενα προσόντα, μόρια, ή στοιχεία επιτροπών. Η εξαγωγή και ορθή αναπαράσταση αυτών των δομημένων δεδομένων αποτελεί μια πρόκληση για την απλή εξαγωγή κειμένου και

υποδεικνύει πιθανές μελλοντικές βελτιώσεις με πιο εξειδικευμένα εργαλεία ανάλυσης πινάκων.

- **Λογότυπα και Γραφικά Στοιχεία**: Στα έγγραφα υπήρχαν συχνά λογότυπα της Ευρωπαϊκής Ένωσης, των οικείων Πανεπιστημίων, του ΕΣΠΑ (Εταιρικό Σύμφωνο για το Πλαίσιο Ανάπτυξης) και άλλων φορέων. Αν και αυτά τα γραφικά στοιχεία δεν περιέχουν άμεσα κειμενική πληροφορία σχετική με το περιεχόμενο της προκήρυξης, η παρουσία τους έπρεπε να αντιμετωπιστεί κατά την προεπεξεργασία ώστε να μην εισάγουν θόρυβο στο εξαγόμενο κείμενο.

Παρακάτω είναι σχετικά παραδείγματα από μια προκήρυξη που διακρίνονται τα λογότυπα του πανεπιστημιακού ιδρύματος, αλλά και του ΕΣΠΑ και του προγράμματος "Ανθρώπινο Δυναμικό και Κοινωνική Συνοχή".



Σχήμα 3.3: Θόρυβος στα PDFs

Η αρχική αυτή έρευνα στα PDFs βοήθησε στην καλύτερη κατανόηση των χαρακτηριστικών του dataset και στην προσαρμογή των βημάτων του Colab script για την αποτελεσματικότερη δυνατή εξαγωγή και καθαρισμό του κειμενικού περιεχομένου. Το τελικό dataset, αν και βασίζεται σε κείμενο, φέρει τα ίχνη της πολυπλοκότητας και της ποικιλομορφίας των πρωτότυπων δημοσίων εγγράφων.

3.6 Επίλογος

Η διαδικασία συλλογής δεδομένων και κατασκευής του dataset που περιγράφηκε στο παρόν κεφάλαιο αποτέλεσε ένα θεμελιώδες και κρίσιμο στάδιο για την επιτυχία της διπλωματικής εργασίας.

Ξεκινώντας από την έρευνα στην πλατφόρμα "ΔΙΑΥΓΕΙΑ" και την εφαρμογή συγκεκριμένων κριτηρίων για την επιλογή των προκηρύξεων, επιτεύχθηκε η δημιουργία ενός εστιασμένου σώματος 114 αρχείων PDF. Η αυτοματοποιημένη λήψη αυτών των αρχείων και η επακόλουθη λεπτομερής προεπεξεργασία τους μέσω του εξειδικευμένου Google Colab script, οδήγησαν στην παραγωγή ενός καθαρού και δομημένου dataset σε μορφή JSON. Η αρχική ποιοτική έρευνα των PDFs επιβεβαίωσε την αναγνωσιμότητα του κειμένου και ανέδειξε τις μορφολογικές ιδιαιτερότητες που έπρεπε να ληφθούν υπόψη. Η αποθήκευση των δεδομένων στο Google Drive διασφάλισε την προσβασιμότητα και την ευκολία διαχείρισης. Το παραγόμενο dataset αποτελεί πλέον τη βάση πάνω στην οποία θα αναπτυχθεί και θα αξιολογηθεί το σύστημα ανάκτησης πληροφοριών με χρήση Μεγάλων Γλωσσικών Μοντέλων, όπως θα αναλυθεί στα επόμενα κεφάλαια.

Κεφάλαιο 4ο: Προεπεξεργασία PDFs

4.1 Εισαγωγή

Η μετατροπή των αρχικών εγγράφων PDF, που συλλέχθηκαν από τη "ΔΙΑΥΓΕΙΑ", σε ένα καθαρό και δομημένο σύνολο δεδομένων (dataset) αποτελεί ένα κρίσιμο προπαρασκευαστικό στάδιο για την επιτυχή εφαρμογή τεχνικών Επεξεργασίας Φυσικής Γλώσσας και την τροφοδότηση των Μεγάλων Γλωσσικών Μοντέλων. Τα αρχεία PDF, από τη φύση τους, συχνά περιέχουν μη δομημένο κείμενο, στοιχεία μορφοποίησης, επαναλαμβανόμενες πληροφορίες όπως επικεφαλίδες και υποσέλιδα, υπερσυνδέσμους, καθώς και ειδικές ενότητες που δεν είναι σχετικές με το κύριο πληροφοριακό περιεχόμενο. Για τους λόγους αυτούς, αναπτύχθηκε μια αυτοματοποιημένη αλυσίδα (pipeline) προεπεξεργασίας σε περιβάλλον Google Colab, με στόχο την εξαγωγή του ουσιώδους κειμένου και την ομαλοποίησή του. Το παρόν κεφάλαιο αναλύει τα επιμέρους βήματα αυτής της διαδικασίας.

4.2 Εξαγωγή κειμένου (PyPDF2)

Το θεμελιώδες πρώτο βήμα στην επεξεργασία των αρχείων PDF είναι η εξαγωγή του ακατέργαστου κειμενικού τους περιεχομένου. Για τον σκοπό αυτό, αξιοποιήθηκε η βιβλιοθήκη Python PyPDF2. Συγκεκριμένα, η συνάρτηση extract_pages_from_pdf(file_path) που αναπτύχθηκε, ανοίγει το αρχείο PDF στην καθορισμένη διαδρομή (file_path) σε δυαδική λειτουργία ανάγνωσης ("rb"). Στη συνέχεια, χρησιμοποιεί την κλάση PyPDF2.PdfReader για να διαβάσει το περιεχόμενο του PDF. Η συνάρτηση επαναλαμβάνεται σε κάθε σελίδα του εγγράφου (reader.pages) και μέσω της μεθόδου page.extract_text(), εξάγει το κείμενο από αυτήν. Σε περίπτωση που μια σελίδα δεν περιέχει εξαγώγιμο κείμενο ή είναι κενή, η μέθοδος επιστρέφει None, το οποίο η συνάρτηση χειρίζεται επιστρέφοντας μια κενή συμβολοσειρά ("") για τη συγκεκριμένη σελίδα. Το αποτέλεσμα είναι μια λίστα από συμβολοσειρές, όπου κάθε συμβολοσειρά αντιστοιχεί στο κείμενο μιας σελίδας του αρχικού PDF.

Ενώ υπάρχουν και άλλες ισχυρές βιβλιοθήκες Python για την επεξεργασία PDF, όπως οι pdfminer.six για πιο λεπτομερή ανάλυση διάταξης, η ReportLab για τη δημιουργία PDF, ή λύσεις που ενσωματώνουν OCR όπως το Tesseract OCR για την εξαγωγή κειμένου από εικόνες, η PyPDF2 προτιμήθηκε για την αρχική αυτή φάση λόγω της απλότητας της στην εξαγωγή ακατέργαστου κειμένου και της έλλειψης εξωτερικών εξαρτήσεων για αυτή τη βασική λειτουργία [17]. Πιο εξειδικευμένες βιβλιοθήκες, παρότι προσφέρουν προηγμένες δυνατότητες, θα εισήγαγαν μεγαλύτερη πολυπλοκότητα ή απαιτήσεις που δεν ήταν απαραίτητες για τον θεμελιώδη στόχο της εξαγωγής του βασικού κειμενικού περιεχομένου, για τον οποίο η απόδοση και η ευκολία χρήσης της PyPDF2 κρίθηκαν επαρκείς.

4.3 Αφαίρεση headers/footers

Πολλά επίσημα έγγραφα, συμπεριλαμβανομένων των προκηρύξεων, περιέχουν επαναλαμβανόμενες επικεφαλίδες (headers) και υποσέλιδα (footers) σε κάθε σελίδα (π.χ., τίτλος εγγράφου, αριθμός σελίδας, στοιχεία φορέα, ΕΣΠΑ, Προγραμμα…). Αυτά τα στοιχεία, αν και χρήσιμα για την ανάγνωση

του εγγράφου, εισάγουν θόρυβο κατά την αυτοματοποιημένη επεξεργασία κειμένου. Για την αντιμετώπιση αυτού του ζητήματος, υλοποιήθηκε η συνάρτηση: remove_headers_footers(pages, min_pages=3, top_lines=5, bottom_lines=2, ratio=0.8).

Η λογική της συνάρτησης αυτής είναι να:

1. Ελέγχει αρχικά αν ο αριθμός των σελίδων (len(pages)) είναι μικρότερος από ένα κατώφλι (min_pages, εξ ορισμού 3). Αν ναι, δεν πραγματοποιείται αφαίρεση, καθώς δεν υπάρχει επαρκής αριθμός σελίδων για την ασφαλή ταυτοποίηση επαναλαμβανόμενων μοτίβων.
2. Συλλέγει τις πρώτες top_lines (εξ ορισμού 5) γραμμές κειμένου από κάθε σελίδα και τις τελευταίες bottom_lines (εξ ορισμού 2) γραμμές, αφού αφαιρεθούν τα αρχικά/τελικά κενά (strip()).
3. Χρησιμοποιώντας την collections.Counter, καταμετρά τη συχνότητα εμφάνισης κάθε γραμμής στις συλλογές των επικεφαλίδων και των υποσέλιδων αντίστοιχα.
4. Μια γραμμή θεωρείται επαναλαμβανόμενη επικεφαλίδα ή υποσέλιδο εάν εμφανίζεται σε τουλάχιστον ratio (εξ ορισμού 80%) του συνολικού αριθμού σελίδων.
5. Τέλος, δημιουργείται μια νέα λίστα σελίδων (cleaned), όπου από κάθε αρχική σελίδα αφαιρούνται οι γραμμές που ταυτοποιήθηκαν ως κοινές επικεφαλίδες ή υποσέλιδα.

Με την αφαίρεση των επαναλαμβανόμενων κεφαλίδων και υποσέλιδων από τα έγγραφα PDF, επιτύχαμε μια σημαντική "κάθαρση" του κειμένου, απομακρύνοντας στοιχεία που δεν ανήκουν στον κύριο κορμό του περιεχομένου. Συγκεκριμένα, αφαιρέθηκαν πληροφορίες όπως οι αριθμοί ΑΔΑ, οι φράσεις σχετικά με τη συγχρηματοδότηση από την Ευρωπαϊκή Ένωση και το ΕΣΠΑ, οι ονομασίες και διευθύνσεις των πανεπιστημίων και οι αριθμοί των σελίδων. Αυτή η διαδικασία έχει ως αποτέλεσμα ένα πιο εστιασμένο σύνολο κειμένου, απαλλαγμένο από περιττές επαναλήψεις και μεταδεδομένα, καθιστώντας το πιο κατάλληλο για περαιτέρω ανάλυση και επεξεργασία, όπως η εξαγωγή πληροφοριών ή η εκπαίδευση μοντέλων μηχανικής μάθησης, καθώς μειώνεται ο "θόρυβος" και το κείμενο επικεντρώνεται στην ουσιαστική πληροφορία του κάθε εγγράφου ξεχωριστά.

4.4 Καθαρισμός κειμένου

Μετά την εξαγωγή του βασικού κειμένου και την αφαίρεση των επικεφαλίδων/υποσέλιδων, ακολουθεί μια σειρά από βήματα καθαρισμού με στόχο την περαιτέρω ομαλοποίηση και προετοιμασία του κειμένου για τις επόμενες φάσεις επεξεργασίας. Αυτά τα βήματα υλοποιούνται κυρίως εντός της συνάρτησης clean_text(text, lowercase=False).

4.4.1 Αφαίρεση link & URL

Τα δημόσια έγγραφα συχνά περιέχουν υπερσυνδέσμους (links) προς ιστοσελίδες ή άλλα έγγραφα. Αυτοί οι σύνδεσμοι, αν και χρήσιμοι για τον άνθρωπο αναγνώστη, δεν προσφέρουν άμεση σημασιολογική αξία για την ανάλυση του περιεχομένου του ίδιου του εγγράφου και μπορούν να θεωρηθούν θόρυβος. Η συνάρτηση remove_links(text) χρησιμοποιεί μια κανονική έκφραση (regex) url_pattern = r'https?://\S+|www\.\S+|\S+\.(com|gr|org|net|edu)\S*' για να εντοπίσει και να αφαιρέσει τις περισσότερες κοινές μορφές URL από το κείμενο, αντικαθιστώντας τες με κενή συμβολοσειρά.

4.4.2 Προστασία ημερομηνιών

Η αφαίρεση σημείων στίξης, η οποία αποτελεί επόμενο βήμα καθαρισμού, μπορεί να αλλοιώσει τη δομή των ημερομηνιών (π.χ., αφαίρεση των "/" ή "." από την "24/01/2025"). Για να αποφευχθεί αυτό, εφαρμόζεται ένας μηχανισμός προστασίας και επαναφοράς ημερομηνιών.

Η συνάρτηση protect_dates(text) χρησιμοποιεί την κανονική έκφραση date_pattern = r'\b\d{1,2}[/\.-]\d{1,2}[/\.-]\d{4}\b' για να εντοπίσει ημερομηνίες στο κείμενο. Κάθε εντοπισμένη ημερομηνία αντικαθίσταται προσωρινά με ένα μοναδικό placeholder της μορφής __DATE{i}__, όπου i είναι ένας αύξων αριθμός. Οι αρχικές ημερομηνίες αποθηκεύονται σε μια λίστα. Μετά την ολοκλήρωση των υπολοίπων βημάτων καθαρισμού (όπως η αφαίρεση στίξης), η συνάρτηση restore_dates(text, dates) αναλαμβάνει να αντικαταστήσει τα placeholders με τις αρχικές, άθικτες ημερομηνίες [18].

Έτσι, αποτρέπονται σφάλματα στην ερμηνεία ή απώλεια πολύτιμων χρονικών πληροφοριών, που θα μπορούσαν να προκύψουν αν οι ημερομηνίες μεταβάλλονταν κατά την αφαίρεση σημείων στίξης. Με τον τρόπο αυτό, το τελικό κείμενο διατηρεί την ορθότητα και την ακεραιότητα των δεδομένων του.

4.4.3 Regex σήμανση & ομαλοποίηση στίξης

Ο καθαρισμός από τα σημεία στίξης και η ομαλοποίηση των κενών διαστημάτων είναι ουσιώδη βήματα για την απλοποίηση του κειμένου. Εντός της συνάρτησης clean_text:

1. Χρησιμοποιείται η κανονική έκφραση PUNCTUATION = r'[.,;!:/\——«»()’·]' για να οριστεί ένα σύνολο κοινών σημείων στίξης. Η εντολή re.sub(PUNCTUATION, '', text) αφαιρεί όλα αυτά τα σημεία από το κείμενο.
2. Ακολουθεί η ομαλοποίηση των κενών διαστημάτων. Η εντολή re.sub(r'\s+', ' ', text) αντικαθιστά πολλαπλά διαδοχικά κενά, αλλαγές γραμμής ή tabs με ένα μοναδικό κενό διάστημα.
3. Ομοίως, η εντολή re.sub(r'\n+', '\n', text) αντικαθιστά πολλαπλές διαδοχικές αλλαγές γραμμής με μία μόνο, διατηρώντας όμως τη βασική δομή των παραγράφων όπου αυτή υποδηλώνεται από αλλαγή γραμμής.

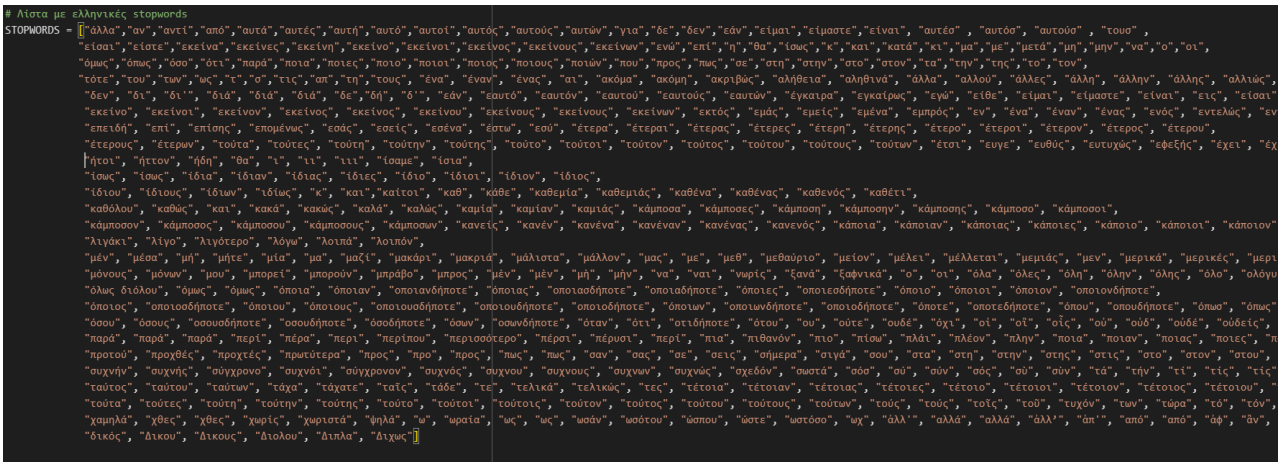
Προαιρετικά, η συνάρτηση clean_text δέχεται την παράμετρο lowercase=True (η οποία χρησιμοποιείται εξ ορισμού στην κύρια αλυσίδα επεξεργασίας) για τη μετατροπή όλου του κειμένου σε πεζούς χαρακτήρες μέσω της text.casefold(), συμβάλλοντας στην ομογενοποίηση του κειμένου.

4.5 Stop-words list

Οι άχρηστες λέξεις (stopwords) είναι λέξεις που εμφανίζονται με μεγάλη συχνότητα σε μια γλώσσα (π.χ., άρθρα, σύνδεσμοι, προθέσεις) αλλά συνήθως δεν προσφέρουν σημαντική σημασιολογική πληροφορία για την κατανόηση του περιεχομένου ενός κειμένου, ειδικά για εργασίες ανάκτησης πληροφοριών. Στον κώδικα, ορίζεται μια εκτενής λίστα ελληνικών stopwords (STOPWORDS).

Κατά την επεξεργασία του κειμένου εντός της συνάρτησης clean_text, μετά την αρχική τμηματοποίηση σε λέξεις (text.split()), κάθε λέξη ελέγχεται. Εάν μια λέξη (μετά τη μετατροπή της σε πεζά) περιλαμβάνεται στη λίστα STOPWORDS και έχει μήκος μεγαλύτερο από 3 χαρακτήρες (μια ευρετική για να αποφευχθεί η αφαίρεση πολύ μικρών, δυνητικά σημαντικών λέξεων που μπορεί να

έτυχε να μοιάζουν με stopword), τότε αυτή η λέξη αφαιρείται από το τελικό καθαρισμένο κείμενο. Αυτό το βήμα βοηθά στη μείωση του "θορύβου" και επιτρέπει στα γλωσσικά μοντέλα να εστιάσουν στις πιο σημαντικές, πληροφοριακά, λέξεις [19].



Σχήμα 4.1: StopWords που Χρησιμοποιήθηκαν

4.6 Greek stemmer

Η μορφολογική ομαλοποίηση (stemming) είναι η διαδικασία αναγωγής των λέξεων στον μορφολογικό τους θέμα ή ρίζα, αφαιρώντας τις καταλήξεις τους. Στόχος είναι η ομαδοποίηση διαφορετικών γραμματικών τύπων της ίδιας λέξης (π.χ., "προκήρυξη", "προκηρύξεις", "προκηρύξεων") κάτω από μια κοινή μορφή, βελτιώνοντας την ανάκληση κατά την αναζήτηση.

Στον κώδικα, υλοποιείται μια απλή συνάρτηση μορφολογικής ομαλοποίησης, η simple_greek_stemmer(word). Η συνάρτηση αυτή λειτουργεί βάσει κανόνων: περιέχει μια προκαθορισμένη λίστα από κοινές ελληνικές καταλήξεις (suffixes). Για κάθε λέξη εισόδου, ελέγχει αν τελειώνει με κάποια από αυτές τις καταλήξεις. Αν ναι, η κατάληξη αφαιρείται και επιστρέφεται το υπόλοιπο της λέξης. Αν καμία κατάληξη δεν ταιριάζει, η λέξη επιστρέφεται ως έχει.

Αυτός ο απλός stemmer εφαρμόζεται στις λέξεις που παραμένουν μετά την αφαίρεση των stopwords, εντός της συνάρτησης clean_text. Αν και δεν πρόκειται για έναν πλήρη μορφολογικό αναλυτή lemmatizer που θα λάμβανε υπόψη το γραμματικό γένος ή τον αριθμό, προσφέρει ένα ικανοποιητικό και αποτελεσματικό επίπεδο ομαλοποίησης για τους σκοπούς της παρούσας εργασίας [20].

Παρακάτω βρίσκεται ο κώδικας που χρησιμοποιήθηκε για το Greek Stemmer, καθώς κυρίως έχει σημασία να προσέξουμε τις διαφορετικές καταλήξεις της ελληνικής γλώσσας που μπορεί να περιγραφεί ως πλούσια και με μεγάλη ποικιλομορφία.

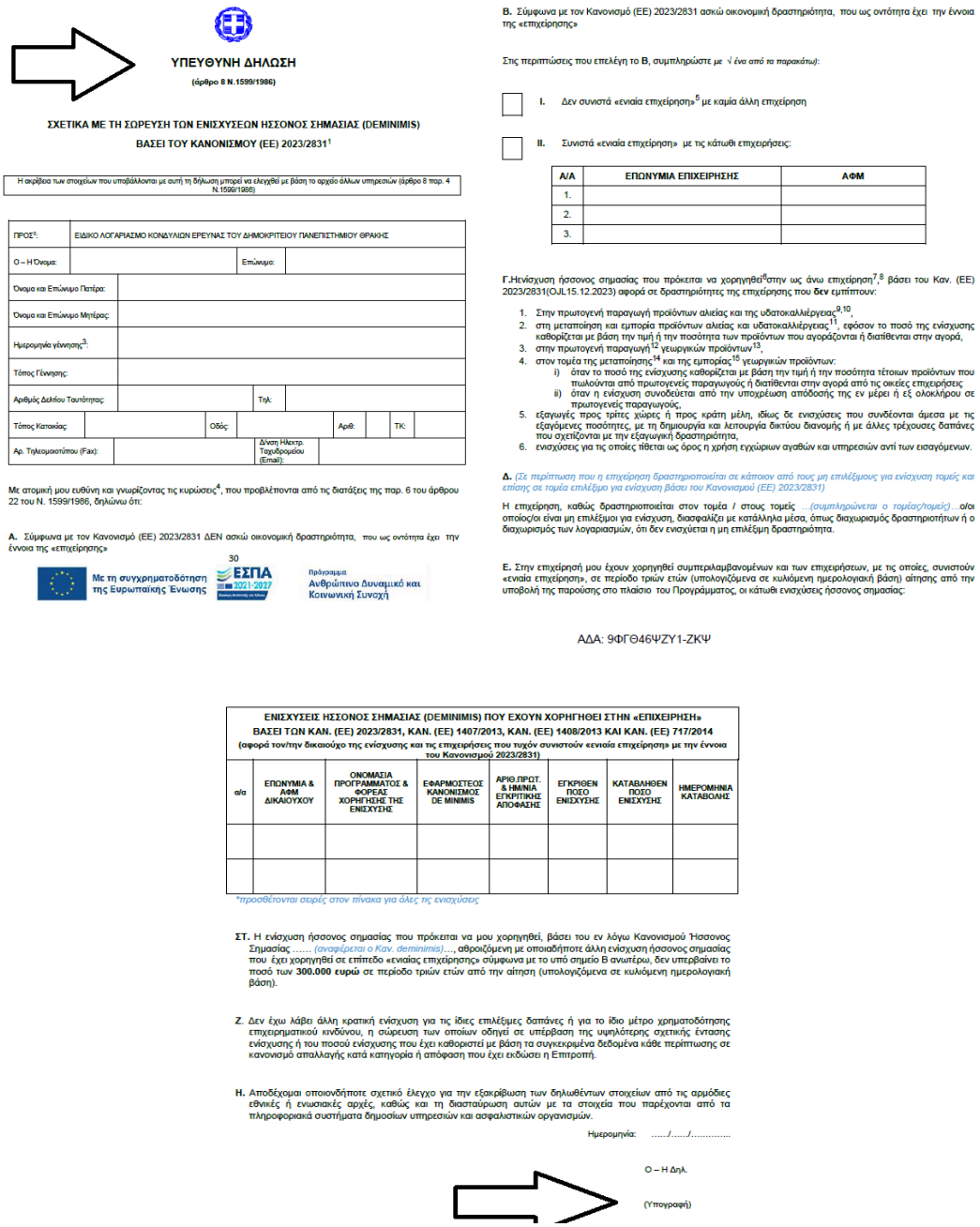
```
def simple_greek_stemmer(word):
    suffixes = ['ω', 'εις', 'ει', 'ουμε', 'ετε', 'ουν', 'α', 'ε', 'ο', 'ου', 'ων', 'ας', 'η', 'ης', 'ες', 'οι', 'ων', 'ου', 'αμε', 'ιεσ', 'ιέσ'
                'ατε', 'ανε', 'ος', 'αι', 'ους', 'ασ', 'άσ', 'ωσ', 'ώσ', 'ησ', 'ήσ' , 'οσ', 'όσ', 'εσ', 'έσ', 'εισ', 'είσ', 'ουσ', 'ούσ'
                'ώ', 'είς', 'εί', 'ούμε', 'ουμέ', 'έτε', 'ετέ', 'ούν', 'ά', 'έ', 'ό', 'ού', 'ών', 'άς', 'ή', 'ής', 'ές', 'οί',
                'ών', 'ού', 'άμε', 'άτε', 'άνε', 'ός', 'αί', 'ούς', 'ώς' ]
    for s in suffixes:
        if word.endswith(s):
            return word[:-len(s)]
    return word
```

Σχήμα 4.2: Greek stemmer και οι καταλήξεις

4.7 Pipeline process_document() – διάγραμμα ροής

Η συνάρτηση process_document(file_path, lowercase=False) ενσωματώνει και ενορχηστρώνει όλα τα προαναφερθέντα βήματα προεπεξεργασίας, λειτουργώντας ως η κεντρική αλυσίδα (pipeline) μετατροπής ενός αρχείου PDF σε καθαρό, επεξεργασμένο κείμενο. Η ροή των λειτουργιών εντός αυτής της συνάρτησης μπορεί να περιγραφεί ως εξής:

1. **Είσοδος**: Η διαδρομή προς ένα αρχείο PDF (file_path) και μια προαιρετική παράμετρος για μετατροπή σε πεζά (lowercase).
2. **Εξαγωγή Σελίδων**: Καλείται η extract_pages_from_pdf() για να εξαχθεί το κείμενο από κάθε σελίδα του PDF, επιστρέφοντας μια λίστα από συμβολοσειρές.
3. **Αφαίρεση Επικεφαλίδων/Υποσέλιδων**: Η λίστα των σελίδων περνά από τη συνάρτηση remove_headers_footers() για την αφαίρεση επαναλαμβανόμενων και μη χρήσιμων στοιχείων κατά την εργασία μας.
4. **Συνένωση Κειμένου**: Οι επεξεργασμένες σελίδες συνενώνονται σε μία ενιαία συμβολοσειρά, με χαρακτήρα αλλαγής γραμμής (\n) ως διαχωριστικό.
5. **Αφαίρεση "Υπεύθυνης Δήλωσης"**: Το ενιαίο κείμενο περνά από τη remove_declarations() για την αφαίρεση των συγκεκριμένων τυποποιημένων ενοτήτων. Ο κώδικας αυτός λειτουργεί σαν ένα "ψαλίδι" που εντοπίζει τμήματα κειμένου που ξεκινούν με "ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ" (ή οποιαδήποτε άλλη start_keyword οριστεί) και τελειώνουν με "(Υπογραφή)" (ή οποιαδήποτε άλλη end_keyword). Επαναλαμβάνει αυτή τη διαδικασία μέχρι να μην υπάρχουν άλλα τέτοια τμήματα στο κείμενο και επιστρέφει το κείμενο χωρίς αυτές τις "δηλώσεις". Παρατηρήθηκε πολλές φορές σε πολλά PDFs η ύπαρξη τέτοιων σχετικών δηλώσεων και με αυτό τον τρόπο εξαφανίστηκαν από το κείμενο δίνοντας ένα πιο καθαρό dataset έτοιμο για περαιτέρω επεξεργασία. Παρακάτω όπως φαίνεται με την σειρά από αριστερά στα δεξιά, αρχικός εντοπισμός είναι ο τίτλος "Υπεύθυνη Δήλωσης" και καταλήγει στο τέλος με την λέξη "(Υπογραφή)".



ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ

(άρθρο 8 Ν.1599/1986)

ΣΧΕΤΙΚΑ ΜΕ ΤΗ ΣΩΡΕΥΣΗ ΤΩΝ ΕΝΙΣΧΥΣΕΩΝ ΗΣΣΟΝΟΣ ΣΗΜΑΣΙΑΣ (DEMINIMIS)

ΒΑΣΕΙ ΤΟΥ ΚΑΝΟΝΙΣΜΟΥ (ΕΕ) 2023/2831[1]

Η ακρίβεια των στοιχείων που υποβάλλονται με αυτή τη δήλωση μπορεί να ελεγχθεί με βάση το αρχείο άλλων υπηρεσιών (άρθρο 8 παρ. 4 Ν.1599/1986)

ΠΡΟΣ[2]:	ΕΙΔΙΚΟ ΛΟΓΑΡΙΑΣΜΟ ΚΟΝΔΥΛΙΩΝ ΕΡΕΥΝΑΣ ΤΟΥ ΔΗΜΟΚΡΙΤΕΙΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΘΡΑΚΗΣ		
Ο – Η Όνομα:		Επώνυμο:	
Όνομα και Επώνυμο Πατέρα:			
Όνομα και Επώνυμο Μητέρας:			
Ημερομηνία γέννησης[3]:			
Τόπος Γέννησης:			
Αριθμός Δελτίου Ταυτότητας:		Τηλ:	
Τόπος Κατοικίας:	Οδός:	Αριθ:	ΤΚ:
Αρ. Τηλεομοιοτύπου (Fax):		Δ/νση Ηλεκτρ. Ταχυδρομείου (Email):	

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις[4], που προβλέπονται από τις διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

Α. Σύμφωνα με τον Κανονισμό (ΕΕ) 2023/2831 ΔΕΝ ασκώ οικονομική δραστηριότητα, που ως οντότητα έχει την έννοια της «επιχείρησης»

30

Με τη συγχρηματοδότηση της Ευρωπαϊκής Ένωσης | ΕΣΠΑ 2021-2027 | Πρόγραμμα: Ανθρώπινο Δυναμικό και Κοινωνική Συνοχή

Β. Σύμφωνα με τον Κανονισμό (ΕΕ) 2023/2831 ασκώ οικονομική δραστηριότητα, που ως οντότητα έχει την έννοια της «επιχείρησης»

Στις περιπτώσεις που επελέγη το Β, συμπληρώστε *με √ ένα από τα παρακάτω*:

☐ **Ι.** Δεν συνιστά «ενιαία επιχείρηση»[5] με καμία άλλη επιχείρηση

☐ **ΙΙ.** Συνιστά «ενιαία επιχείρηση» με τις κάτωθι επιχειρήσεις:

Α/Α	ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗΣΗΣ	ΑΦΜ
1.		
2.		
3.		

Γ.Ενίσχυση ήσσονος σημασίας που πρόκειται να χορηγηθεί[6]στην ως άνω επιχείρηση[7,8] βάσει του Καν. (ΕΕ) 2023/2831(OJL15.12.2023) αφορά σε δραστηριότητες της επιχείρησης που **δεν** εμπίπτουν:

1. Στην πρωτογενή παραγωγή προϊόντων αλιείας και της υδατοκαλλιέργειας[9,10],
2. στη μεταποίηση και εμπορία προϊόντων αλιείας και υδατοκαλλιέργειας[11], εφόσον το ποσό της ενίσχυσης καθορίζεται με βάση την τιμή ή την ποσότητα των προϊόντων που αγοράζονται ή διατίθενται στην αγορά,
3. στην πρωτογενή παραγωγή[12] γεωργικών προϊόντων[13],
4. στον τομέα της μεταποίησης[14] και της εμπορίας[15] γεωργικών προϊόντων:
   i) όταν το ποσό της ενίσχυσης καθορίζεται με βάση την τιμή ή την ποσότητα τέτοιων προϊόντων που πωλούνται από πρωτογενείς παραγωγούς ή διατίθενται στην αγορά από τις οικείες επιχειρήσεις
   ii) όταν η ενίσχυση συνοδεύεται από την υποχρέωση απόδοσής της εν μέρει ή εξ ολοκλήρου σε πρωτογενείς παραγωγούς,
5. εξαγωγές προς τρίτες χώρες ή προς κράτη μέλη, ιδίως δε ενισχύσεις που συνδέονται άμεσα με τις εξαγόμενες ποσότητες, με τη δημιουργία και λειτουργία δικτύου διανομής ή με άλλες τρέχουσες δαπάνες που σχετίζονται με την εξαγωγική δραστηριότητα,
6. ενισχύσεις για τις οποίες τίθεται ως όρος η χρήση εγχώριων αγαθών και υπηρεσιών αντί των εισαγόμενων.

Δ. *(Σε περίπτωση που η επιχείρηση δραστηριοποιείται σε κάποιον από τους μη επιλέξιμους για ενίσχυση τομείς και επίσης σε τομέα επιλέξιμο για ενίσχυση βάσει του Κανονισμού (ΕΕ) 2023/2831)*

Η επιχείρηση, καθώς δραστηριοποιείται στον τομέα / στους τομείς *...(συμπληρώνεται ο τομέας/τομείς)...* α/αι οποίος/οι είναι μη επιλέξιμοι για ενίσχυση, διασφαλίζει με κατάλληλα μέσα, όπως διαχωρισμός δραστηριοτήτων ή ο διαχωρισμός των λογαριασμών, ότι δεν ενισχύεται η μη επιλέξιμη δραστηριότητα.

Ε. Στην επιχείρησή μου έχουν χορηγηθεί συμπεριλαμβανομένων και των επιχειρήσεων, με τις οποίες, συνιστούν «ενιαία επιχείρηση», σε περίοδο τριών ετών (υπολογιζόμενα σε κυλιόμενη ημερολογιακή βάση) αίτησης από την υποβολή της παρούσης στο πλαίσιο του Προγράμματος, οι κάτωθι ενισχύσεις ήσσονος σημασίας:

ΑΔΑ: 9ΦΓΘ46ΨΖΥ1-ΖΚΨ

ΕΝΙΣΧΥΣΕΙΣ ΗΣΣΟΝΟΣ ΣΗΜΑΣΙΑΣ (DEMINIMIS) ΠΟΥ ΕΧΟΥΝ ΧΟΡΗΓΗΘΕΙ ΣΤΗΝ «ΕΠΙΧΕΙΡΗΣΗ» ΒΑΣΕΙ ΤΩΝ ΚΑΝ. (ΕΕ) 2023/2831, ΚΑΝ. (ΕΕ) 1407/2013, ΚΑΝ. (ΕΕ) 1408/2013 ΚΑΙ ΚΑΝ. (ΕΕ) 717/2014
(αφορά τον/την δικαιούχο της ενίσχυσης και τις επιχειρήσεις που τυχόν συνιστούν «ενιαία επιχείρηση» με την έννοια του Κανονισμού 2023/2831)

α/α	ΕΠΩΝΥΜΙΑ & ΑΦΜ ΔΙΚΑΙΟΥΧΟΥ	ΟΝΟΜΑΣΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ & ΦΟΡΕΑΣ ΧΟΡΗΓΗΣΗΣ ΤΗΣ ΕΝΙΣΧΥΣΗΣ	ΕΦΑΡΜΟΣΤΕΟΣ ΚΑΝΟΝΙΣΜΟΣ DE MINIMIS	ΑΡΙΘ.ΠΡΩΤ. & ΗΜ/ΝΙΑ ΕΓΚΡΙΤΙΚΗΣ ΑΠΟΦΑΣΗΣ	ΕΓΚΡΙΘΕΝ ΠΟΣΟ ΕΝΙΣΧΥΣΗΣ	ΚΑΤΑΒΛΗΘΕΝ ΠΟΣΟ ΕΝΙΣΧΥΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ ΚΑΤΑΒΟΛΗΣ

**προσθέτονται σειρές στον πίνακα για όλες τις ενισχύσεις*

ΣΤ. Η ενίσχυση ήσσονος σημασίας που πρόκειται να μου χορηγηθεί, βάσει του εν λόγω Κανονισμού Ήσσονος Σημασίας *..... (αναφέρεται ο Καν. deminimis)...*, αθροιζόμενη με οποιαδήποτε άλλη ενίσχυση ήσσονος σημασίας που έχει χορηγηθεί σε επίπεδο «ενιαίας επιχείρησης» σύμφωνα με το υπό σημείο Β ανωτέρω, δεν υπερβαίνει το ποσό των **300.000 ευρώ** σε περίοδο τριών ετών από την αίτηση (υπολογιζόμενα σε κυλιόμενη ημερολογιακή βάση).

Ζ. Δεν έχω λάβει άλλη κρατική ενίσχυση για τις ίδιες επιλέξιμες δαπάνες ή για το ίδιο μέτρο χρηματοδότησης επιχειρηματικού κινδύνου, η σώρευση των οποίων οδηγεί σε υπέρβαση της υψηλότερης σχετικής έντασης ενίσχυσης ή του ποσού ενίσχυσης που έχει καθοριστεί με βάση τα συγκεκριμένα δεδομένα κάθε περίπτωσης σε κανονισμό απαλλαγής κατά κατηγορία ή απόφαση που έχει εκδώσει η Επιτροπή.

Η. Αποδέχομαι οποιονδήποτε σχετικό έλεγχο για την εξακρίβωση των δηλωθέντων στοιχείων από τις αρμόδιες εθνικές ή ενωσιακές αρχές, καθώς και τη διασταύρωση αυτών με τα στοιχεία που παρέχονται από τα πληροφοριακά συστήματα δημοσίων υπηρεσιών και ασφαλιστικών οργανισμών.

Ημερομηνία: ...../...../.....

Ο – Η Δηλ.

(Υπογραφή)

Σχήμα 4.3: Αφαίρεση "Υπεύθυνης Δήλωσης"

6. **Τελικός Καθαρισμός και Ομαλοποίηση**: Το κείμενο τροφοδοτείται στη συνάρτηση clean_text() (με την καθορισμένη επιλογή για lowercase), η οποία εκτελεί με τη σειρά:
   - Ομογενοποίηση πεζών/κεφαλαίων (αν lowercase=True).
   - Προστασία ημερομηνιών (protect_dates).
   - Αφαίρεση υπερσυνδέσμων (remove_links).
   - Αφαίρεση καθορισμένων σημείων στίξης (PUNCTUATION regex).
   - Ομαλοποίηση πολλαπλών κενών και αλλαγών γραμμής.
   - Tokenization (μέσω text.split()).
   - Αφαίρεση stopwords (για λέξεις > 3 χαρακτήρων που δεν είναι placeholders ημερομηνιών).

- Εφαρμογή του simple_greek_stemmer στις υπόλοιπες λέξεις.
- Επαναφορά των προστατευμένων ημερομηνιών (restore_dates).
- Αφαίρεση αρχικών/τελικών κενών από το τελικό αποτέλεσμα.

7. **Έξοδος:** Η συνάρτηση επιστρέφει την τελική, καθαρισμένη και ομαλοποιημένη συμβολοσειρά που αντιπροσωπεύει το περιεχόμενο του αρχικού PDF.

Αυτή τα βήματα διασφαλίζουν ότι από κάθε αρχείο PDF εξάγεται ένα κείμενο όσο το δυνατόν πιο καθαρό από θόρυβο και στοιχεία μορφοποίησης, έτοιμο για την επόμενη φάση της τμηματοποίησης (chunking) και της δημιουργίας διανυσματικών αναπαραστάσεων.

4.8 Ποιοτικός έλεγχος & παραδείγματα καθαρισμού

Μετά την ολοκλήρωση της υλοποίησης και εφαρμογής της αλυσίδας προεπεξεργασίας μέσω της συνάρτησης process_document(), σε όλα τα συγκεντρωθέντα αρχεία PDF, κρίθηκε σκόπιμη η διενέργεια ποιοτικού ελέγχου στα παραγόμενα αποτελέσματα. Σκοπός του ελέγχου ήταν να διασφαλιστεί ότι οι διάφορες λειτουργίες καθαρισμού εκτελέστηκαν ορθά, απομακρύνοντας ανεπιθύμητο περιεχόμενο χωρίς να επηρεάζουν ουσιώδεις πληροφορίες που περιέχονται στον πυρήνα των προκηρύξεων.

Η διαδικασία περιλάμβανε την επιλογή δείγματος από τα επεξεργασμένα έγγραφα του παραχθέντος JSON dataset και την αντιπαραβολή τους με τα αντίστοιχα αρχικά PDF αρχεία. Πραγματοποιήθηκε αναλυτική εξέταση συγκεκριμένων περιπτώσεων προκειμένου να διαπιστωθεί:

- Η επιτυχής αφαίρεση των URLs και των κοινών επικεφαλίδων/υποσέλιδων.
- Η σωστή διατήρηση των ημερομηνιών παρά την αφαίρεση της στίξης.
- Η αποτελεσματικότητα της αφαίρεσης των stopwords και η λογική συνέπεια του stemming.
- Η γενική αναγνωσιμότητα και συνοχή του καθαρισμένου κειμένου.

- **Παράδειγμα 1: Αφαίρεση URL και Στίξης**
    1. **Κείμενο Πριν (6ΥΥΚ46ΨΖΣ4-Γ2Ι.pdf):** Οι υποψήφιοι/ες ενημερώνονται με δική τους επιμέλεια για τα αποτελέσματα μέσω της ιστοσελίδας της Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών , καθώς και της ιστοσελίδας του ΕΛΚΕ του Εθνικού Μετσόβιου Πολυτεχνείου (https://www.elke.ntua.gr/), καθώς και στο πρόγραμμα ΔΙΑΥΓΕΙΑ.
    2. **Κείμενο Μετά (dataset.json):** υποψήφιο ενημερώνοντ δικ επιμέλει αποτελέσματ μέσ ιστοσελίδ σχολ εφαρμοσμέν μαθηματικ φυσικ επιστημ καθ ιστοσελίδ ελκ εθνικ μετσόβι πολυτεχνεί καθ πρόγραμμ διαυγει
- **Παράδειγμα 2: Αφαίρεση Stopwords και Stemming**
    1. **Κείμενο Πριν (ΨΗ0746ΨΖ2Ν-ΡΨ5.pdf):** Ειδικότερα, τα προσωπικά δεδομένα επεξεργάζονται από το Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών για τις ανάγκες του έργου με βάση τη διαδικασία που περιγράφεται στην πρόσκληση και

αποστέλλονται στον φορέα χρηματοδότησης και συγκεκριμένα στην αρμόδια μονάδα της Ειδικής Υπηρεσίας Διαχείρισης. Η επεξεργασία των προσωπικών δεδομένων περιορίζεται στον απολύτως αναγκαίο βαθμό και αποτρέπεται κάθε περαιτέρω ή ανακριβής επεξεργασία κατά τρόπο ασύμβατο με τους προκαθορισμένους σκοπούς.

2. **Κείμενο Μετά (dataset.json):** ειδικότερ προσωπικ δεδομέν επεξεργάζοντ εθνικ καποδιστριακ πανεπιστήμι αθην ανάγκ έργ βάσ διαδικασί περιγράφετ πρόσκλησ αποστέλλοντ φορέ χρηματοδότησ συγκεκριμέν αρμόδι μονάδ ειδικ υπηρεσί διαχείρισ επεξεργασί προσωπικ δεδομέν περιορίζετ απολύτ αναγκαί βαθμ αποτρέπετ περαιτέρ ανακριβ επεξεργασί τρόπ ασύμβατ προκαθορισμέν σκοπούσ

Ο ποιοτικός έλεγχος επιβεβαίωσε σε μεγάλο βαθμό την ορθότητα της λογικής προεπεξεργασίας, αν και πάντα υπάρχει το ενδεχόμενο κάποιες ειδικές περιπτώσεις ή ασυνήθιστες μορφοποιήσεις στα PDF να μην αντιμετωπίζονται τέλεια από τις κατάλληλες μεθόδους που χρησιμοποιήθηκαν (π.χ., remove_headers_footers). Ωστόσο, για την πλειοψηφία των εγγράφων, το παραγόμενο κείμενο ήταν σημαντικά καθαρότερο και πιο κατάλληλο για τις ανάγκες των γλωσσικών μοντέλων.

Με αυτόν τον τρόπο, παρουσιάζεται το dataset, που με μια μικρή ματιά φαίνεται χαοτικό, αλλά το γλωσσικό μοντέλο, είναι σε θέση να εντοπίσει μοτίβα και κρυμμένες σχέσεις, εξάγοντας πολύτιμες πληροφορίες.

Εικόνα 4.4: Τελικό Dataset

4.9 Επίλογος

Η προεπεξεργασία των εγγράφων PDF αποτέλεσε μια απαραίτητη και πολύπλοκη φάση κατά την προετοιμασία του συνόλου δεδομένων για την παρούσα διπλωματική εργασία. Αξιοποιώντας μια σειρά από αυτοματοποιημένες διαδικασίες σε περιβάλλον Google Colab, μετατρέψαμε αποτελεσματικά τις 114 προκηρύξεις από τη πλατφόρμα «ΔΙΑΥΓΕΙΑ», από την αρχική, συχνά δυσνόητη μορφή τους, σε ένα καθαρό και ομοιογενές κειμενικό σύνολο. Η μεθοδολογία περιλάμβανε αρχικά την εξαγωγή κειμένου μέσω της βιβλιοθήκης PyPDF2, ακολουθούμενη από ευρετική απομάκρυνση επαναλαμβανόμενων τμημάτων, όπως οι επικεφαλίδες και τα υποσέλιδα, καθώς και από τον προσεκτικό καθαρισμό των κειμένων από υπερσυνδέσμους και περιττά σημεία στίξης, προστατεύοντας παράλληλα τις ημερομηνίες. Επιπλέον, εφαρμόστηκαν βασικές τεχνικές

επεξεργασίας φυσικής γλώσσας, όπως η απομάκρυνση των stopwords και η απλή μορφολογική κανονικοποίηση στην ελληνική γλώσσα. Η υλοποίηση όλων αυτών των βημάτων ενσωματώθηκε στην κεντρική συνάρτηση process_document(), εξασφαλίζοντας έτσι τη συνέπεια σε όλο το σύνολο των εγγράφων. Τέλος, ο ποιοτικός έλεγχος που πραγματοποιήθηκε επιβεβαίωσε την αποτελεσματικότητα της διαδικασίας, δημιουργώντας ένα σύνολο δεδομένων κατάλληλο για την λειτουργία Μεγάλων Γλωσσικών Μοντέλων και την ανάπτυξη του συστήματος ανάκτησης πληροφοριών που θα εξεταστεί στα επόμενα κεφάλαια.

Κεφάλαιο 5ο: Δημιουργία Embeddings & Διανυσματική Βάση

5.1 Εισαγωγή

Μετά την ολοκλήρωση της προεπεξεργασίας των αρχείων PDF, όπως περιγράφηκε στο Κεφάλαιο 4, και τη δημιουργία του καθαρού, δομημένου dataset κειμένων , το επόμενο κρίσιμο στάδιο είναι η μετατροπή αυτών των κειμένων σε μια μορφή που επιτρέπει την αποτελεσματική σημασιολογική αναζήτηση. Αυτό επιτυγχάνεται μέσω της δημιουργίας διανυσματικών αναπαραστάσεων (embeddings) για κάθε απόσπασμα κειμένου και της αποθήκευσής τους σε μια εξειδικευμένη διανυσματική βάση δεδομένων. Το παρόν κεφάλαιο αναλύει τη διαδικασία επιλογής των γλωσσικών μοντέλων για την παραγωγή των embeddings, την τεχνική της τμηματοποίησης του κειμένου (chunking), τις στρατηγικές εξαγωγής των διανυσμάτων ανάλογα με την αρχιτεκτονική του κάθε μοντέλου (π.χ., [CLS] token, mean pooling, last token embedding), τη δομή και τη χρήση της βάσης δεδομένων ChromaDB για την αποθήκευση και ευρετηρίαση, καθώς και την καταγραφή των παραγόμενων διανυσμάτων μαζί με τα σχετικά μεταδεδομένα. Τέλος, εξετάζεται η απόδοση της συνολικής διαδικασίας ευρετηρίασης όσον αφορά το μέγεθος των παραγόμενων αρχείων, τον απαιτούμενο χρόνο εκτέλεσης και τις ανάγκες σε υπολογιστικούς πόρους (RAM/VRAM)

5.2 Επιλογή μοντέλων

Αρχικά ο πίνακας συνοψίζει με απλό τρόπο τις τρεις βασικές αρχιτεκτονικές που χρησιμοποιούνται στα μοντέλα τύπου Transformer, ανάλογα με τον ρόλο που καλούνται να παίξουν. Ο τύπος encoder-only επικεντρώνεται αποκλειστικά στην κατανόηση του κειμένου – διαβάζει, αναλύει και ερμηνεύει, χωρίς να παράγει νέο περιεχόμενο. Είναι ιδανικός για εφαρμογές όπως η αναγνώριση συναισθήματος ή η κατηγοριοποίηση κειμένων. Από την άλλη, ο decoder-only τύπος κάνει το ακριβώς αντίθετο: ξεκινά από μια είσοδο (όπως μια ερώτηση ή ένα prompt) και συνεχίζει γράφοντας το επόμενο κείμενο λέξη-λέξη, πράγμα που τον καθιστά κατάλληλο για chatbots, μεταφράσεις ή γεννήτριες περιεχομένου. Τέλος, η encoder-decoder αρχιτεκτονική είναι κάτι σαν τον συνδυασμό και των δύο, αφού συνδυάζει την κατανόηση με την παραγωγή και χρησιμοποιείται όταν χρειάζεται πλήρης «κατανόηση» του αρχικού κειμένου πριν παραχθεί το αντίστοιχο αποτέλεσμα – όπως γίνεται, για παράδειγμα, στη μηχανική μετάφραση ή στις αυτόματες συνοψίσεις.

Τύπος	Χρήση
Encoder-only	Κατανόηση κειμένου (π.χ. ανάλυση συναισθήματος, κατηγοριοποίηση)
Decoder-only	Παραγωγή κειμένου (π.χ. απαντήσεις, μετάφραση, συνοψίσεις)
Encoder-Decoder	Συνδυασμός κατανόησης και παραγωγής (π.χ. μετάφραση, συνοψίσεις)

Πίνακας 5.1: Encoder και Decoder

Για την παραγωγή των διανυσματικών αναπαραστάσεων (embeddings) των τμημάτων κειμένου, η παρούσα εργασία αξιοποίησε τρία διακριτά Μεγάλα Γλωσσικά Μοντέλα (LLMs), τα οποία έχουν αναπτυχθεί και προσαρμοστεί για την ελληνική γλώσσα. Η επιλογή τους έγινε με στόχο τη διερεύνηση και σύγκριση της απόδοσης μοντέλων διαφορετικών αρχιτεκτονικών και μεγεθών:

1. **GreekBERT**: Το μοντέλο nlpaueb/bert-base-greek-uncased-v1, που είναι μοντέλο βασισμένο στην αρχιτεκτονική BERT (encoder-only) και προεκπαιδευμένο σε ελληνικά κείμενα όπως έχει αναφερθεί .
2. **Meltemi**: Το μοντέλο ilsp/Meltemi-7B-Instruct-v1.5, πολύ μεγάλο μοντέλο μεγέθους 7 δισεκατομμυρίων παραμέτρων, βασισμένο σε αρχιτεκτονική decoder-only και βελτιστοποιημένο για την κατανόηση οδηγιών (instruct-tuned).
3. **KriKri**: Το μοντέλο ilsp/Llama-Krikri-8B-Base, το άλλο μεγάλο μοντέλο μεγέθους 8 δισεκατομμυρίων παραμέτρων, βασισμένο στην αρχιτεκτονική Llama (decoder-only).

Η φόρτωση των μοντέλων και των αντίστοιχων tokenizers τους πραγματοποιήθηκε μέσω της βιβλιοθήκης AutoTokenizer και AutoModel της transformers από το Hugging Face [21]. Λόγω του μεγέθους των Meltemi και KriKri, εφαρμόστηκαν ειδικές ρυθμίσεις κατά τη φόρτωση για τη διαχείριση των υπολογιστικών πόρων:

- Η παράμετρος trust_remote_code=True χρησιμοποιήθηκε για τα μοντέλα Meltemi και KriKri, όπου αυτό απαιτούνταν. Αυτή επιτρέπει στη βιβλιοθήκη Hugging Face Transformers να κατεβάσει και να εκτελέσει κώδικα Python. Αυτό είναι απαραίτητο επειδή ειδικά layers και βοηθητικές συναρτήσεις δεν αποτελούν μέρος της τυπικής βιβλιοθήκης transformers.
- Για το Meltemi, ορίστηκε torch_dtype=torch.float16 για τη μείωση των απαιτήσεων σε VRAM.
- Για το KriKri, χρησιμοποιήθηκε η παράμετρος device_map="auto" για την αυτόματη κατανομή του μοντέλου στις διαθέσιμες υπολογιστικές μονάδες (π.χ., TPU, GPU ή CPU), μια τεχνική απαραίτητη για μοντέλα αυτού του μεγέθους. Το GreekBERT, όντας μικρότερο, φορτώθηκε με πιο τυπικές ρυθμίσεις (.to(DEVICE).eval()).

5.3 Chunking κειμένου

Για την παραγωγή ενός μοναδικού διανυσματικού αναπαράστασης (embedding) για κάθε chunk κειμένου, χρησιμοποιήθηκαν διαφορετικές στρατηγικές ανάλογα με την αρχιτεκτονική του κάθε LLM. Όλες οι στρατηγικές κατέληγαν σε L2-κανονικοποίηση των τελικών διανυσμάτων. Ως L2-κανονικοποίηση των τελικών διανυσμάτων αναφέρεται η διαδικασία κατά την οποία κάθε διάνυσμα-embedding που παράγεται για ένα chunk κειμένου τροποποιείται έτσι ώστε το Ευκλείδειο μήκος του (ή η L2 νόρμα του) να ισούται με 1.

Με απλά λόγια, κάθε διάνυσμα μπορεί να νοηθεί ως ένα βέλος στον πολυδιάστατο χώρο. Η L2-κανονικοποίηση «μαζεύει» κάθε τέτοιο βέλος, έτσι ώστε όλα να αποκτούν το ίδιο μήκος (μήκος 1), διατηρώντας όμως την κατεύθυνσή τους.

5.3.1 Embedding [CLS] Token για Encoder Μοντέλα

Για το GreekBERT , το οποίο είναι ένα μοντέλο βασισμένο στην αρχιτεκτονική encoder του Transformer, χρησιμοποιήθηκε η συνάρτηση _cls_embed_setup. Σε αυτή την προσέγγιση, το embedding του ειδικού token [CLS], το οποίο προστίθεται στην αρχή κάθε ακολουθίας εισόδου και η αναπαράστασή του στο τελευταίο κρυφό επίπεδο του μοντέλου (last_hidden_state[:, 0, :]), θεωρείται ότι περιέχει μια συνολική σημασιολογική πληροφορία για ολόκληρη την ακολουθία. Αυτό το διάνυσμα [CLS] εξάγεται και, μετά από L2-κανονικοποίηση, χρησιμοποιείται ως το embedding του chunk.

5.3.2 Mean Pooling για Decoder-Instruct Μοντέλα

Για το μοντέλο Meltemi , το οποίο είναι ένα decoder-based μοντέλο, εφαρμόστηκε η τεχνική της μέσης συνελικτικής προβολής (mean pooling), όπως υλοποιήθηκε στη συνάρτηση _mean_pooling_embed_setup [22]. Η διαδικασία περιλαμβάνει τον υπολογισμό του μέσου όρου όλων των διανυσμάτων των tokens από το τελευταίο κρυφό επίπεδο (last_hidden_state) του μοντέλου, λαμβάνοντας υπόψη τη μάσκα προσοχής (attention_mask) ώστε να αγνοηθούν τα padding tokens. Το μέσο διάνυσμα που προκύπτει υποβάλλεται σε L2-κανονικοποίηση.

Πιο απλά, το μοντέλο επεξεργάζεται μια πρόταση, δημιουργώντας για κάθε λέξη μια κωδικοποιημένη μορφή (μια σειρά αριθμών). Για να σχηματίσει μια συνολική αντίληψη για όλη την πρόταση, το μοντέλο υπολογίζει τον μέσο όρο αυτών των κωδικοποιημένων μορφών. Η "μάσκα προσοχής" βοηθά το μοντέλο να εστιάσει μόνο στις σημαντικές λέξεις, παραβλέποντας όσες χρησιμεύουν απλώς ως "γέμισμα". Στο τέλος, αυτή η μέση τιμή προσαρμόζεται για να έχει ένα τυπικό μέγεθος, κάνοντας τις συγκρίσεις πιο εύκολες.

5.3.3 Embedding Τελευταίου Token ή Mean Pooling για Decoder-Base Μοντέλα

Το KriKri , το οποίο βασίζεται στην αρχιτεκτονική Llama, η συνάρτηση _extract_embeddings_setup σχεδιάστηκε για να εξάγει το embedding του τελευταίου πραγματικού token της ακολουθίας από το τελευταίο κρυφό επίπεδο (model_output.hidden_states[-1]). Η ταυτοποίηση του τελευταίου πραγματικού token γίνεται με βάση τη μάσκα προσοχής και την κατεύθυνση του padding που χρησιμοποιεί ο tokenizer (tokenizer_setup.padding_side). Επιπρόσθετα, ο κώδικας μας περιέχει σχολιασμένη τη δυνατότητα εφαρμογής mean pooling, παρόμοια με αυτή του Meltemi, ως μια πιο

σοβαρή επιλογή εάν ο ακριβής εντοπισμός του τελευταίου token αποδεικνυόταν προβληματικός. Το τελικό διάνυσμα, είτε από το τελευταίο token είτε από mean pooling, υφίσταται L2-κανονικοποίηση.

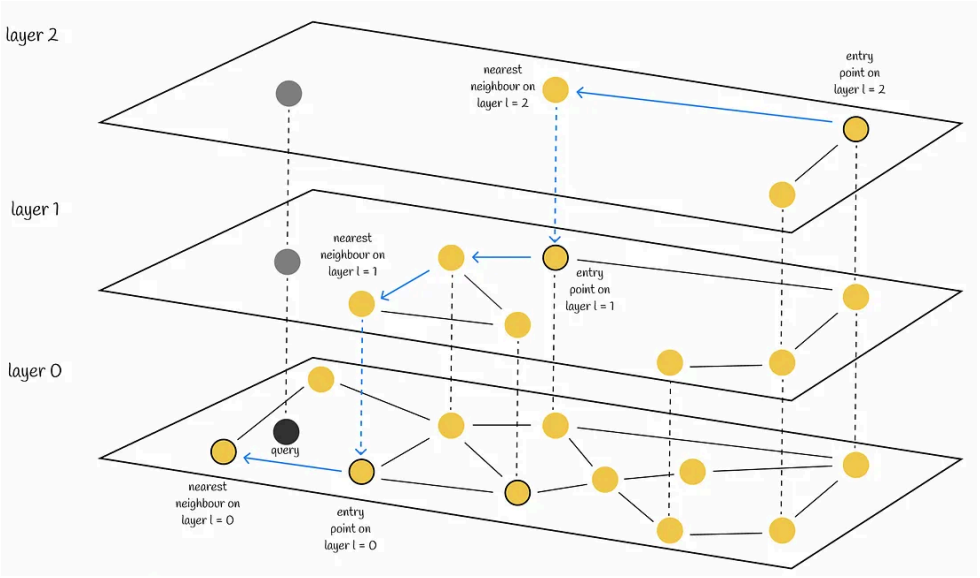
Σε όλες τις περιπτώσεις, η παραγωγή των embeddings γινόταν σε παρτίδες με μέγεθος BATCH_EMB που κυμαινόταν από 4 για το KriKri, 8 για το Meltemi, έως 32 για το GreekBERT, αντανακλώντας τις διαφορετικές υπολογιστικές τους απαιτήσεις για την αποδοτικότερη χρήση της μνήμης GPU.

5.4 ChromaDB Persistent Client – Δομή collection

Για την αποθήκευση, ευρετηρίαση και αποτελεσματική ανάκτηση των παραγόμενων διανυσματικών αναπαραστάσεων, επιλέχθηκε η ανοιχτού κώδικα διανυσματική βάση δεδομένων ChromaDB. Η χρήση του chromadb.PersistentClient επέτρεψε τη δημιουργία μιας μόνιμης βάσης δεδομένων στο σύστημα αρχείων, σε διαδρομές μοναδικές για κάθε μοντέλο.

Εντός της βάσης, δημιουργήθηκε μια ξεχωριστή συλλογή (collection) για κάθε μοντέλο, με μοναδικό όνομα (COL_NAME, π.χ., collection_chatbotvol107). Κατά τη δημιουργία της συλλογής, μέσω της μεθόδου get_or_create_collection, ορίστηκε ως μεταδεδομένο η χρήση του αλγορίθμου ευρετηρίασης HNSW (Hierarchical Navigable Small World)[23] και ως μετρική απόστασης/ομοιότητας η "cosine" (metadata={"hnsw:space":"cosine"}). Αυτή η επιλογή είναι κατάλληλη για τα L2-κανονικοποιημένα embeddings που παρήχθησαν, καθώς η ομοιότητα συνημιτόνου μετρά τη γωνία μεταξύ των διανυσμάτων, αγνοώντας το μέγεθός τους (το οποίο είναι ήδη ομοιόμορφο λόγω κανονικοποίησης).

Η σχετική εικόνα δείχνει την αρχιτεκτονική ενός γραφήματος Hierarchical Navigable Small World (HNSW). Απεικονίζει μια πολυεπίπεδη δομή όπου η αναζήτηση για τον πλησιέστερο γείτονα ενός σημείου "query" (μαύρος κύκλος) ξεκινά από το ανώτερο, πιο αραιό επίπεδο (layer 2). Στη συνέχεια, η αναζήτηση κατεβαίνει διαδοχικά στα κατώτερα, πυκνότερα επίπεδα (layer 1 και layer 0), χρησιμοποιώντας τον πλησιέστερο γείτονα του προηγούμενου επιπέδου ως σημείο εισόδου για το επόμενο. Αυτή η ιεραρχική προσέγγιση επιταχύνει σημαντικά την εύρεση των πλησιέστερων γειτόνων σε μεγάλους όγκους δεδομένων.



Σχήμα 5.1: Hierarchical Navigable Small World [24]

5.5 Αποθήκευση μεταδεδομένων & embedding vectors

Η εισαγωγή των δεδομένων στην κατάλληλη συλλογή της ChromaDB για κάθε μοντέλο έγινε με τη χρήση της μεθόδου add(). Για κάθε τμήμα κειμένου (chunk) που προέκυψε από την επεξεργασία των PDF, αποθηκεύτηκαν τα ακόλουθα στοιχεία:

- **ids**: Ένα μοναδικό αναγνωριστικό για κάθε chunk, συνήθως με τη μορφή {doc_id}_c{chunk_idx} (π.χ., 1_c1, 1_c2) όπου doc_id είναι το αναγνωριστικό του αρχικού εγγράφου από το dataset και chunk_idx ο αύξων αριθμός του chunk εντός του συγκεκριμένου εγγράφου.
- **embeddings**: Τα διανυσματικά αναπαραστάσης του chunk, όπως αυτά παρήχθησαν από τη στρατηγική του εκάστοτε μοντέλου (CLS για GreekBERT, mean pooling για Meltemi, last token/mean pooling για KriKri) και κανονικοποιήθηκαν (L2 normalization). Τα NumPy arrays των embeddings μετατράπηκαν σε Python lists για συμβατότητα με την ChromaDB.
- **documents**: Το ίδιο το κείμενο του chunk, σύμφωνα με τον κώδικα, τα κείμενα που αποθηκεύονταν ως documents ήταν τα pre_chunks_setup, δηλαδή τα τμήματα που είχαν υποστεί την αρχική προεπεξεργασία (stemming, stopwords κλπ) και χρησιμοποιούνταν επίσης για την παραγωγή των lexical matrices.
- **metadatas**: Ένα λεξικό Python που περιείχε πρόσθετες πληροφορίες (μεταδεδομένα) για κάθε chunk. Αυτά περιλάμβαναν το id του αρχικού εγγράφου, τον τίτλο του (title - συνήθως το όνομα του αρχείου PDF), το URL του αρχικού εγγράφου στο Google Drive (url), τον αύξοντα αριθμό του chunk εντός του εγγράφου (chunk_num), και τον συνολικό αριθμό των chunks για το συγκεκριμένο έγγραφο (total_chunks).

5.6 Απόδοση ευρετηρίασης

Η διαδικασία ευρετηρίασης, από τη φόρτωση των μοντέλων έως την τελική αποθήκευση των embeddings και μεταδεδομένων στην ChromaDB, έχει συγκεκριμένες απαιτήσεις όσον αφορά το μέγεθος αποθήκευσης, τον χρόνο εκτέλεσης και τη χρήση μνήμης.

- **Μέγεθος (Size)**: Το τελικό μέγεθος της διανυσματικής βάσης δεδομένων (DB_DIR_APP) εξαρτάται από τον αριθμό των chunks, τη διάσταση των embeddings που παράγει κάθε μοντέλο και τον τρόπο αποθήκευσης της ChromaDB. Στην δοκιμές που πραγματοποιήθηκαν ήμασταν στο εύρος απο 2000-3000.
- **Χρόνος (Time):** Η συνολική διάρκεια της διαδικασίας setup_database_and_assets επηρεάζεται σημαντικά από το επιλεγμένο LLM. Η φόρτωση μοντέλων μεγέθους 7-8 δισεκατομμυρίων παραμέτρων (Meltemi, KriKri) ήταν σημαντικά πιο χρονοβόρα από τη φόρτωση του GreekBERT. Η παραγωγή των embeddings αποτελεί το πιο υπολογιστικά έντονο και χρονοβόρο μέρος. Η ταχύτητα εξαρτάται από το μέγεθος του μοντέλου, τον αριθμό των chunks, το μέγεθος του BATCH_EMB και την ισχύ της μονάδας (CPU/GPU). Έγινε χρήση tqdm για την εμφάνιση γραμμών προόδου καθώς και την αποτελεσματικότερη αποκωδικοποίηση των προβλημάτων.
- **Απαιτήσεις Μνήμης (RAM/VRAM):** Η χρήση VRAM είναι κρίσιμη κατά τη φόρτωση και εκτέλεση των LLMs, ειδικά για τα Meltemi και KriKri. Η επιλογή torch_dtype=torch.float16

(για το Meltemi) και η χρήση device_map="auto" (KriKri) αποσκοπούν στη βελτιστοποίηση της χρήσης VRAM. Το μέγεθος BATCH_EMB (32 για GreekBERT, 8 για Meltemi, 4 για KriKri) προσαρμόστηκε ώστε να αντιστοιχεί στις απαιτήσεις της διαφορετικής μνήμης κάθε μοντέλου και της διαθέσιμης VRAM, αποφεύγοντας σφάλματα "out-of-memory".

Είναι σημαντική η κατανόηση για την εκτίμηση της πρακτικότητας και της ενδεχόμενης επέκτασης της λύσης, ειδικά εάν πρόκειται να εφαρμοστεί σε μεγαλύτερα σύνολα δεδομένων ή σε περιβάλλοντα με διαφορετικούς υπολογιστικούς πόρους.

5.7 Επίλογος

Σε αυτό το κεφάλαιο, εξετάσαμε πώς επιλέχθηκαν τρία ελληνικά γλωσσικά μοντέλα για να μετατρέψουν τα προ επεξεργασμένα κείμενα των προκηρύξεων σε ποιοτικά embeddings (διανυσματικές αναπαραστάσεις). Εξηγήσαμε ότι χρησιμοποιήσαμε τεχνικές όπως το mean pooling, ή την επιλογή συγκεκριμένων tokens ([CLS] ή τελευταίο token), ανάλογα με το πώς λειτουργεί κάθε μοντέλο. Στη συνέχεια, αναφέραμε πώς εγκαταστήσαμε και ρυθμίσαμε τη βάση δεδομένων ChromaDB, δίνοντας έμφαση στον τρόπο αποθήκευσης των διανυσμάτων και των μεταδεδομένων που τα συνοδεύουν. Τέλος, συζητήσαμε παράγοντες που επηρεάζουν την ταχύτητα και την απόδοση, όπως το μέγεθος των δεδομένων, τον χρόνο που απαιτείται για την επεξεργασία τους και την υπολογιστική ισχύ που χρειάζεται. Με την ολοκλήρωση αυτών των διαδικασιών, δημιουργήσαμε μια ολοκληρωμένη βάση δεδομένων, η οποία μπορεί πλέον να αξιοποιηθεί από ένα σύστημα που παρέχει ακριβείς και χρήσιμες απαντήσεις σε ερωτήσεις χρηστών.

Κεφάλαιο 6ο: Υβριδικός Αλγόριθμος Αναζήτησης

6.1 Εισαγωγή

Αφού τα προ επεξεργασμένα τμήματα κειμένου (chunks) από τις προκηρύξεις έχουν μετατραπεί σε διανυσματικές αναπαραστάσεις (embeddings) και έχουν αποθηκευτεί μαζί με τα σχετικά μεταδεδομένα και τις λεξικές τους αναπαραστάσεις όπως περιγράφηκε στα Κεφάλαια 4 και 5, το επόμενο κρίσιμο βήμα είναι η υλοποίηση ενός αποδοτικού μηχανισμού ανάκτησης πληροφορίας. Αυτός ο μηχανισμός πρέπει να είναι ικανός να κατανοεί τα ερωτήματα του χρήστη, διατυπωμένα σε φυσική ελληνική γλώσσα, και να επιστρέφει τα πιο σχετικά αποσπάσματα εγγράφων από την ευρετηριασμένη βάση γνώσεων.

Για την επίτευξη αυτού του στόχου, στην παρούσα εργασία αναπτύχθηκε ένας υβριδικός αλγόριθμος αναζήτησης (hybrid search). Η υβριδική προσέγγιση επιλέχθηκε διότι συνδυάζει τα πλεονεκτήματα δύο διαφορετικών τύπων αναζήτησης: της λεξικής αναζήτησης, που βασίζεται στην αντιστοίχιση λέξεων-κλειδιών και είναι αποτελεσματική στον εντοπισμό ακριβών όρων, και της σημασιολογικής αναζήτησης, που βασίζεται στα embeddings και είναι ικανή να αντιληφθεί το εννοιολογικό περιεχόμενο και τη συνάφεια, ακόμη και όταν δεν υπάρχει ακριβής λεξική ταύτιση. Το παρόν κεφάλαιο περιγράφει τα δύο αυτές προσεγγίσεις αναζήτησης, τον τρόπο στάθμισης των αποτελεσμάτων τους, τη μέθοδο ενίσχυσης των ακριβών ταυτίσεων και παρουσιάζει τον αλγόριθμο υλοποίησης της συνάρτησης hybrid_search_gradio που αποτελεί τον πυρήνα του συστήματος ανάκτησης.

6.2 Lexical μονοπάτι: HashingVectorizer char gram + word gram

Το λεξικό μονοπάτι του υβριδικού συστήματος αναζήτησης έχει ως κύριο στόχο τον εντοπισμό τμημάτων κειμένου που περιέχουν όρους ή φράσεις παρόμοιες ή ταυτόσημες με αυτές του ερωτήματος του χρήστη, βασιζόμενο στην επιφανειακή αντιστοίχιση των λέξεων. Για την υλοποίηση αυτού του μονοπατιού, αξιοποιήθηκαν οι λεξικές αναπαραστάσεις που δημιουργήθηκαν κατά το στάδιο της ευρετηρίασης , με χρήση της κλάσης HashingVectorizer από τη βιβλιοθήκη scikit-learn.

Ειδικά, για κάθε chunk στο dataset, δημιουργήθηκαν δύο τύποι λεξικών αναπαραστάσεων:

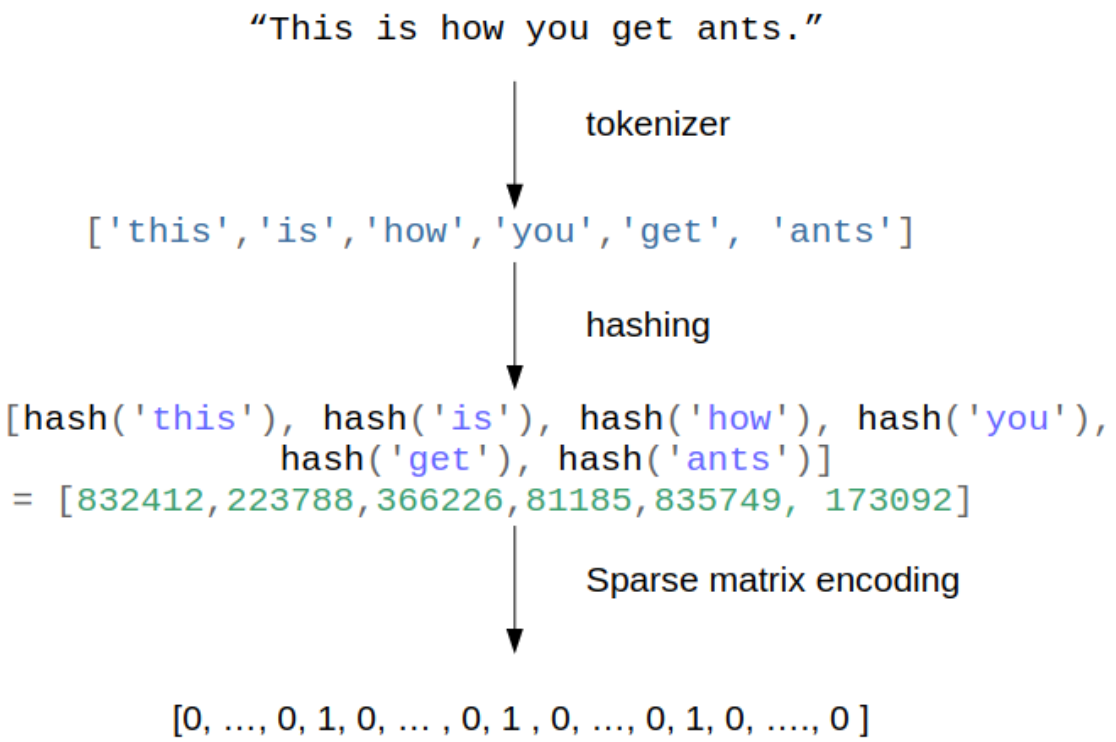
1. Character n-grams: Χρησιμοποιήθηκε ένας HashingVectorizer (char_vec) ρυθμισμένος να αναλύει το κείμενο σε επίπεδο ακολουθιών χαρακτήρων (n-grams), με μήκος από 2 έως 5 χαρακτήρες (ngram_range=(2,5)). Αυτή η προσέγγιση βοηθά στον εντοπισμό μερικών αντιστοιχίσεων και είναι ανθεκτική σε μικρά ορθογραφικά λάθη ή παραλλαγές λέξεων [25].
2. Word n-grams: Ένας δεύτερος HashingVectorizer (word_vec) χρησιμοποιήθηκε για την ανάλυση του κειμένου σε επίπεδο λέξεων και ακολουθιών δύο λέξεων (bigrams) (ngram_range=(1,2)) [26].

Οι παραγόμενες αραιοί πίνακες (sparse matrices) για όλα τα chunks (X_char και X_word) αποθηκεύτηκαν και κανονικοποιήθηκαν (L2 normalization μέσω sk_normalize) κατά τη φάση της προετοιμασίας.

Όταν ένας χρήστης υποβάλλει ένα ερώτημα, αυτό αρχικά υφίσταται την ίδια διαδικασία προεπεξεργασίας (q_pre = preprocess(query)) με τα κείμενα του dataset. Στη συνέχεια, το προ επεξεργασμένο ερώτημα μετατρέπεται σε διανυσματικές αναπαραστάσεις από τους ίδιους char_vec και word_vec. Οι βαθμολογίες ομοιότητας για κάθε chunk υπολογίζονται μέσω του εσωτερικού γινομένου (dot product, που υλοποιείται με τον τελεστή @ για αραιούς πίνακες) μεταξύ του κανονικοποιημένου διανύσματος του ερωτήματος και των κανονικοποιημένων διανυσμάτων όλων των chunks στις προϋπολογισμένες μήτρες X_char.T και X_word.T.

Οι τελικές λεξικές βαθμολογίες ομοιότητας (lex_sims) για κάθε chunk προκύπτουν από έναν σταθμισμένο συνδυασμό των επιμέρους βαθμολογιών από τα character n-grams (c_score) και τα word n-grams (w_score), σύμφωνα με τον τύπο 0.85 * c_score + 0.15 * w_score, όπως φαίνεται στον κώδικα που εφαρμόστηκε και για τα τρία μοντέλα. Αυτή η στάθμιση δίνει μεγαλύτερη έμφαση στις αντιστοιχίσεις σε επίπεδο χαρακτήρων, οι οποίες μπορεί να είναι πιο ευέλικτες.

Η εικόνα δείχνει τη ροή επεξεργασίας μιας φράσης από το στάδιο του tokenization ως το τελικό αραιό διάνυσμα που παράγει ο HashingVectorizer. Κάθε token μετατρέπεται μέσω συνάρτησης κατακερματισμού σε συγκεκριμένη στήλη του διανύσματος, αποτυπώνοντας έτσι τα n-grams σε μία συμπαγή, αραιή αναπαράσταση.



Σχήμα 6.1: Ροή Επεξεργασίας μιας Φράσης

6.3 Semantic μονοπάτι: cosine similarity σε embedding χώρο

Το σημασιολογικό μονοπάτι της αναζήτησης έχει ως σκοπό στην κατανόηση του βαθύτερου νοήματος του ερωτήματος του χρήστη και στην ανάκτηση τμημάτων κειμένου που είναι εννοιολογικά συναφή, ακόμη και αν δεν χρησιμοποιούν ακριβώς τις ίδιες λέξεις. Αυτό επιτυγχάνεται μέσω της χρήσης των διανυσματικών αναπαραστάσεων (embeddings) που παράγονται από τα επιλεγμένα Μεγάλα Γλωσσικά Μοντέλα μας, όπως περιγράφηκε στο Κεφάλαιο 5.3.

Η διαδικασία περιλαμβάνει:

- Το ερώτημα του χρήστη, μετά από την αρχική προεπεξεργασία (q_pre = preprocess(query)), μετατρέπεται σε ένα embedding vector. Αυτό γίνεται καλώντας την αντίστοιχη συνάρτηση εξαγωγής embedding για το ενεργό μοντέλο cls_embed([q_pre], tok, model) για το GreekBERT, mean_pooling_embed_app([q_pre], tok_app, model_app, DEVICE για το Meltemi, και extract_embeddings_app([q_pre], tok, model) για το KriKri.
- Το παραγόμενο L2-κανονικοποιημένο embedding του ερωτήματος (q_emb_list) χρησιμοποιείται για την υποβολή αιτήματος (query) στην αντίστοιχη συλλογή της ChromaDB, μέσω της μεθόδου col.query().
- Η ChromaDB, χρησιμοποιώντας τον ευρετηριασμένο χώρο των embeddings των chunks και την μετρική ομοιότητας συνημιτόνου (cosine similarity), επιστρέφει τα n_results (π.χ, min(k * 30, len(ids))) πιο "κοντινά" ή σημασιολογικά παρόμοια chunks. Η μέθοδος query της ChromaDB επιστρέφει, μεταξύ άλλων, τις αποστάσεις distances των ανακτηθέντων chunks από το embedding του ερωτήματος.
- Οι σημασιολογικές βαθμολογίες ομοιότητας (sem_sims) υπολογίζονται ως 1 - distance, καθώς η ομοιότητα συνημιτόνου είναι 1 για ταυτόσημα διανύσματα και η ChromaDB επιστρέφει αποστάσεις όπου μικρότερη απόσταση σημαίνει μεγαλύτερη ομοιότητα.

6.4 Στάθμιση αποτελεσμάτων

Για να παραχθεί μια ενιαία κατάταξη των τμημάτων κειμένου (chunks) που λαμβάνει υπόψη τόσο τη λεξική όσο και τη σημασιολογική συνάφειά τους με το ερώτημα του χρήστη, οι βαθμολογίες από τα δύο μονοπάτια αναζήτησης συνδυάζονται. Ο αλγόριθμος υλοποιεί μια γραμμική στάθμιση των δύο βαθμολογιών.

Για κάθε chunk chunk_id_key που έχει ανακτηθεί είτε από το λεξικό είτε από το σημασιολογικό μονοπάτι, ο τελικός υβριδικός βαθμός (s) υπολογίζεται ως εξής:

s = alpha * sem_sims.get(chunk_id_key, 0.0) + (1 - alpha) * lex_sims.get(chunk_id_key, 0.0)

Όπου:

- **sem_sims.get(chunk_id_key, 0.0)** είναι η σημασιολογική βαθμολογία ομοιότητας του chunk ή 0 αν δεν βρέθηκε από το σημασιολογικό μονοπάτι.
- **lex_sims.get(chunk_id_key, 0.0)** είναι η λεξική βαθμολογία ομοιότητας του chunk ή 0 αν δεν βρέθηκε από το λεξικό μονοπάτι.

- **alpha** είναι ένας συντελεστής στάθμισης που καθορίζει τη σχετική συμβολή κάθε μονοπατιού στην τελική βαθμολογία.

Ο κώδικας προβλέπει τη χρήση δύο διαφορετικών τιμών για τον συντελεστή alpha, ανάλογα με το μήκος του προεπεξεργασμένου ερωτήματος (q_pre):

- **ALPHA_BASE = 0.2**: Χρησιμοποιείται για σχετικά σύντομα ερωτήματα έως 30 λέξεις. Σε αυτή την περίπτωση, η λεξική ομοιότητα έχει μεγαλύτερη βαρύτητα με 80% στην τελική βαθμολογία.
- **ALPHA_LONGQ = 0.35**: Χρησιμοποιείται για μεγαλύτερα ερωτήματα, πάνω από 30 λέξεις. Εδώ, η σημασιολογική ομοιότητα λαμβάνει ελαφρώς μεγαλύτερη βαρύτητα με 35% σε σύγκριση με την προηγούμενη περίπτωση, αν και η λεξική εξακολουθεί να κυριαρχεί με 65%.

Η λογική πίσω από αυτή τη διαφοροποίηση είναι ότι για σύντομα, συγκεκριμένα ερωτήματα, η ακριβής αντιστοίχιση λέξεων ,που ευνοείται από τη λεξική αναζήτηση, τείνει να είναι πιο σημαντική. Αντίθετα, για πιο μακροσκελή και περιγραφικά ερωτήματα, η κατανόηση του συνολικού νοήματος, που ευνοείται από τη σημασιολογική αναζήτηση μπορεί να παίζει πιο καθοριστικό ρόλο. Η βελτιστοποίηση αυτών των τιμών alpha ήταν αντικείμενο πειραματισμού για την επίτευξη της καλύτερης δυνατής απόδοσης, με επαναλαμβανόμενες δοκιμές.

6.5 Boost ακριβούς ταύτισης (exact_ids_set)

Πέραν του σταθμισμένου συνδυασμού των λεξικών και σημασιολογικών βαθμολογιών, ο αλγόριθμος hybrid_search_gradio εφαρμόζει μια επιπλέον τεχνική για την ενίσχυση boosting των τμημάτων κειμένου που περιέχουν μια ακριβή ταύτιση (exact match) με το προ επεξεργασμένο ερώτημα του χρήστη.

Αυτό επιτυγχάνεται μέσω της δημιουργίας ενός συνόλου (exact_ids_set) το οποίο περιέχει τα αναγνωριστικά (ids) όλων των προ επεξεργασμένων chunks (pre_chunks) στα οποία η συμβολοσειρά του προεπεξεργασμένου ερωτήματος (q_pre) εντοπίζεται ως υποσυμβολοσειρά. Δηλαδή, exact_ids_set = {ids[i] for i, t in enumerate(pre_chunks) if q_pre in t}.

Κατά τον υπολογισμό της τελικής υβριδικής βαθμολογίας, εάν το αναγνωριστικό ενός chunk (chunk_id_key) περιλαμβάνεται στο exact_ids_set, τότε η υβριδική του βαθμολογία s αντικαθίσταται από μια μέγιστη τιμή, συγκεκριμένα 1.0 στον κώδικα μας. Αυτό εξασφαλίζει ότι τα τμήματα κειμένου που περιέχουν την ακριβή φράση του ερωτήματος του χρήστη θα λάβουν την υψηλότερη δυνατή βαθμολογία και, κατά συνέπεια, θα εμφανιστούν στην κορυφή των αποτελεσμάτων αναζήτησης, μετά την τελική ταξινόμηση.

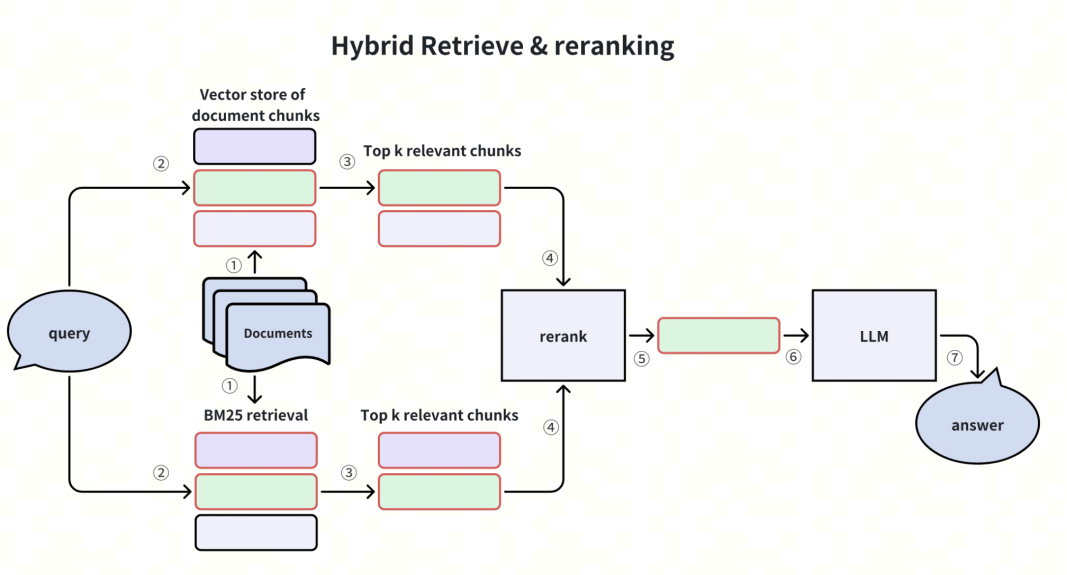
Αυτή η τεχνική είναι ιδιαίτερα χρήσιμη καθώς οι χρήστες συχνά αναζητούν συγκεκριμένες φράσεις ή ορολογίες, και η εμφάνιση των εγγράφων που τις περιέχουν ακριβώς βελτιώνει σημαντικά την αντιληπτή συνάφεια και χρησιμότητα του συστήματος αναζήτησης.

6.6 Αλγόριθμος υλοποίησης (hybrid_search_gradio)

Η υλοποίηση του αλγορίθμου hybrid_search_gradio αποτελεί τον πυρήνα της υβριδικής αναζήτησης, συνδυάζοντας λεξική και σημασιολογική αναζήτηση, με στόχο την ανάκτηση των πιο σχετικών κειμένων ως απάντηση στο ερώτημα του χρήστη [27]. Τα στάδια της λειτουργίας του αλγορίθμου είναι τα εξής:

- **Εισαγωγή Ερωτήματος και Αρχικοί Έλεγχοι**: Η συνάρτηση δέχεται ως είσοδο το κείμενο του ερωτήματος του χρήστη (query) και τον επιθυμητό αριθμό αποτελεσμάτων (k).
- **Προεπεξεργασία Ερωτήματος:** Το ακατέργαστο ερώτημα του χρήστη υποβάλλεται στην ίδια διαδικασία προεπεξεργασίας που εφαρμόστηκε και στα κείμενα του dataset κατά τη φάση της ευρετηρίασης. Αυτό περιλαμβάνει τη μετατροπή σε πεζά, την αφαίρεση τόνων και διακριτικών (strip_acc), την απομάκρυνση σημείων στίξης και ειδικών χαρακτήρων μέσω κανονικών εκφράσεων, την ομαλοποίηση των κενών διαστημάτων και την αφαίρεση των προκαθορισμένων ελληνικών stopwords. Το αποτέλεσμα είναι το προ επεξεργασμένο ερώτημα (q_pre).
- **Καθορισμός Συντελεστή Στάθμισης:** Ένας δυναμικός συντελεστής στάθμισης alpha καθορίζεται ανάλογα με το αριθμό λέξεων του q_pre. Για ερωτήματα που υπερβαίνουν τις 30 λέξεις, χρησιμοποιείται η τιμή ALPHA_LONGQ, ενώ για συντομότερα ερωτήματα χρησιμοποιείται η ALPHA_BASE. Αυτός ο συντελεστής θα χρησιμοποιηθεί αργότερα για τον συνδυασμό των βαθμολογιών από τα δύο μονοπάτια αναζήτησης.
- **Εκτέλεση Σημασιολογικής Αναζήτησης:** Το q_pre μετατρέπεται επίσης σε διανυσματικές αναπαραστάσεις μέσω των προ-φορτωμένων HashingVectorizers. Υπολογίζονται οι βαθμολογίες ομοιότητας αυτών των διανυσμάτων με τα αντίστοιχα διανύσματα όλων των chunks που είναι αποθηκευμένα στις προ-υπολογισμένες αραιούς πίνακες (X_char και X_word). Οι επιμέρους βαθμολογίες συνδυάζονται σταθμισμένα με 0.85 * char_score + 0.15 * word_score για να παραχθεί η τελική λεξική βαθμολογία (lex_sims) για κάθε chunk
- **Εντοπισμός Ακριβών Ταυτίσεων:** Δημιουργείται ένα σύνολο (exact_ids_set) που περιέχει τα αναγνωριστικά των chunks στα οποία το q_pre εντοπίζεται ως ακριβής υποσυμβολοσειρά του προεπεξεργασμένου τους κειμένου.
- **Υπολογισμός Υβριδικής Βαθμολογίας και Ταξινόμηση:** Για κάθε chunk που έχει ανακτηθεί είτε από το σημασιολογικό είτε από το λεξικό μονοπάτι, ή ανήκει στο exact_ids_set, υπολογίζεται μια συνολική υβριδική βαθμολογία. Αυτή προκύπτει από τον σταθμισμένο μέσο όρο της σημασιολογικής και της λεξικής του βαθμολογίας, χρησιμοποιώντας τον συντελεστή alpha που ειπώθηκε νωρίτερα. Εάν ένα chunk ανήκει στο exact_ids_set, η βαθμολογία του ενισχύεται και τίθεται ίση με 1, εξασφαλίζοντας την προτεραιοποίηση του. Όλα τα υποψήφια chunks ταξινομούνται κατά φθίνουσα σειρά βάσει αυτής της τελικής υβριδικής βαθμολογίας.
- **Μορφοποίηση και Επιστροφή Κορυφαίων Αποτελεσμάτων:** Τα καλύτερα αποτελέσματα επιλέγονται, αποφεύγοντας διπλότυπα από τα ίδια αρχικά έγγραφα. Εμφανίζονται σε μορφή Markdown, με τίτλο, απόσπασμα και σύνδεσμο προς το πλήρες έγγραφο PDF.

Στο διάγραμμα βλέπουμε το υβριδικό pipeline: το ερώτημα τροφοδοτείται παράλληλα σε λεξική αναζήτηση (BM25) και σε σημασιολογική αναζήτηση (embeddings σε vector-store)· οι δύο λίστες συγχωνεύονται και περνούν από στάδιο rerank, ώστε να επιλεγούν τα πιο σχετικά chunks.



Σχήμα 6.2: Υβριδικό pipeline

Μέσω αυτής της διαδικασίας, ο αλγόριθμος hybrid_search_gradio συνδυάζει τις δυνάμεις διαφορετικών τεχνικών αναζήτησης για να παρέχει στον χρήστη τα πιο ολοκληρωμένα και σχετικά αποτελέσματα στο ερώτημά του.

6.7 Επίλογος

Το παρόν κεφάλαιο παρουσίασε λεπτομερώς τον υβριδικό μηχανισμό αναζήτησης και ανάκτησης πληροφορίας που αναπτύχθηκε για τον διαλογικό βοηθό. Αναλύθηκαν τα δύο κύρια μονοπάτια που συνθέτουν την υβριδική προσέγγιση: το λεξικό μονοπάτι, που αξιοποιεί τεχνικές HashingVectorizer για την αντιστοίχιση n-grams χαρακτήρων και λέξεων, και το σημασιολογικό μονοπάτι, που βασίζεται στα embeddings που παράγονται από τα επιλεγμένα ελληνικά LLMs και την αναζήτηση ομοιότητας συνημιτόνου εντός της βάσης ChromaDB. Περιγράφηκε η μέθοδος στάθμισης των αποτελεσμάτων από τα δύο αυτά μονοπάτια, με τη χρήση διαφορετικών συντελεστών alpha ανάλογα με το μήκος του ερωτήματος, καθώς και η τεχνική ενίσχυσης (boosting) των αποτελεσμάτων που περιέχουν ακριβή ταύτιση με τη φράση αναζήτησης. Ο συνδυασμός αυτών των τεχνικών αποσκοπεί στη δημιουργία ενός ισχυρού συστήματος ανάκτησης, ικανού να παρέχει ακριβή και σημασιολογικά συναφή αποτελέσματα στα ερωτήματα των χρηστών. Η αξιολόγηση της απόδοσης αυτού του υβριδικού μηχανισμού θα αποτελέσει αντικείμενο των επόμενων κεφαλαίων.

Κεφάλαιο 7ο: Σχεδίαση & Υλοποίηση Εφαρμογής

7.1 Εισαγωγή

Μετά την ολοκλήρωση της δημιουργίας των διανυσματικών αναπαραστάσεων (embeddings) και την οργάνωση της διανυσματικής βάσης δεδομένων, όπως αναλύθηκε στο προηγούμενο κεφάλαιο, το επόμενο βήμα είναι η ανάπτυξη μιας λειτουργικής εφαρμογής που θα επιτρέπει στους τελικούς χρήστες να αλληλεπιδρούν με το σύστημα ανάκτησης πληροφορίας. Το παρόν κεφάλαιο επικεντρώνεται στη σχεδίαση και υλοποίηση αυτής της εφαρμογής. Περιγράφεται η συνολική αρχιτεκτονική του συστήματος, η επιλογή της πλατφόρμας φιλοξενίας Hugging Face Spaces και η χρήση του Persistent Storage, η διαχείριση των γλωσσικών μοντέλων, η αξιοποίηση του Google Cloud Storage για την παροχή των πρωτότυπων εγγράφων PDF, η ανάπτυξη της διεπαφής χρήστη (frontend) με τη βιβλιοθήκη Gradio, καθώς και οι απαιτήσεις του περιβάλλοντος εκτέλεσης. Στόχος είναι η δημιουργία μιας προσβάσιμης και εύχρηστης διεπαφής για την αξιοποίηση του υβριδικού μηχανισμού αναζήτησης που αναπτύχθηκε στα κεφάλαια πιο πάνω

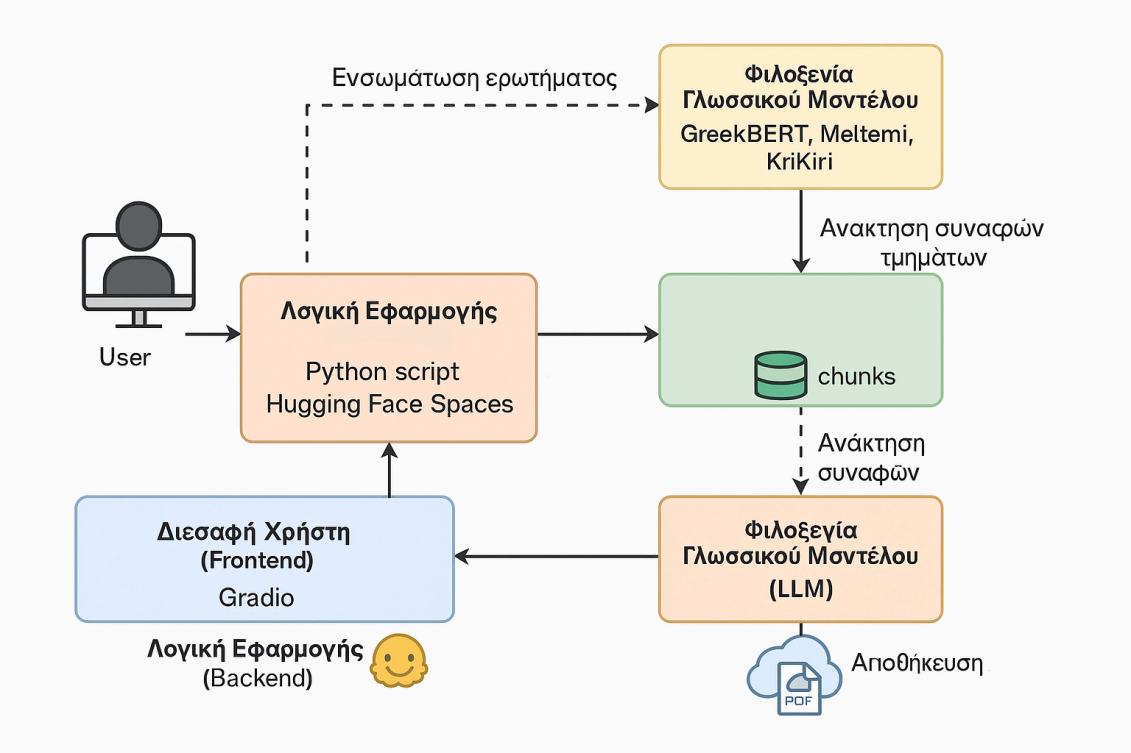
7.2 Αρχιτεκτονική συστήματος

Η αρχιτεκτονική της εφαρμογής του διαλογικού βοηθού σχεδιάστηκε ώστε να είναι πολυτμηματική και να αξιοποιεί τις δυνατότητες των σύγχρονων εργαλείων και πλατφορμών cloud. Για κάθε ένα από τα τρία εξεταζόμενα Μεγάλα Γλωσσικά Μοντέλα, υλοποιήθηκε μια ξεχωριστή, αυτόνομη παρουσία της εφαρμογής, επιτρέποντας την ανεξάρτητη λειτουργία και αξιολόγησή τους. Η γενική αρχιτεκτονική για κάθε παρουσία της εφαρμογής περιλαμβάνει τα εξής βασικά σημεία:

1. **Διεπαφή Χρήστη (Frontend)**: Υλοποιήθηκε με τη χρήση της βιβλιοθήκης Gradio [28]. Παρέχει ένα απλό web interface όπου ο χρήστης μπορεί να εισάγει τα ερωτήματά του σε φυσική ελληνική γλώσσα και να λαμβάνει τα αποτελέσματα της αναζήτησης.
2. **Λογική Εφαρμογής (Backend)**: Το αντίστοιχο LLM και ο tokenizer του φορτώνονται στη μνήμη του Hugging Face Space κατά την εκκίνηση της εφαρμογής και χρησιμοποιούνται για την παραγωγή των embeddings του ερωτήματος.
3. **Φιλοξενία Γλωσσικού Μοντέλου (Model Hosting):** Το αντίστοιχο LLM και ο tokenizer του φορτώνονται στη μνήμη του Hugging Face Space κατά την εκκίνηση της εφαρμογής και χρησιμοποιούνται για την παραγωγή των embeddings του ερωτήματος.
4. **Διανυσματική Βάση Δεδομένων (Vector Database):** Η ChromaDB χρησιμοποιείται για την αποθήκευση και ανάκτηση των embeddings των τμημάτων κειμένου (chunks). Η βάση δεδομένων φιλοξενείται στον Persistent Storage του Hugging Face Space, διασφαλίζοντας τη μόνιμη αποθήκευση των δεδομένων.
5. **Αποθήκευση Λεξικών Δεδομένων (Lexical Search Assets):** Οι προ-υπολογισμένοι HashingVectorizers και οι TF-IDF για τη λεξική αναζήτηση αποθηκεύονται επίσης στον Persistent Storage του Hugging Face Space.
6. **Αποθήκευση Πρωτότυπων Εγγράφων PDF (PDF Storage):** Τα αρχικά αρχεία PDF των προκηρύξεων φιλοξενούνται σε ένα Google Cloud Storage (GCS) bucket για εύκολη και δημόσια προσβάσιμη παροχή τους μέσω συνδέσμων.

Η αλληλεπίδραση των στοιχείων μπορεί να περιγραφεί ως εξής : Ο χρήστης υποβάλλει ένα ερώτημα μέσω της διεπαφής Gradio. Το backend script στο Hugging Face Space λαμβάνει το ερώτημα, το επεξεργάζεται και εκτελεί τον υβριδικό αλγόριθμο αναζήτησης, αξιοποιώντας το τοπικά φορτωμένο LLM για το σημασιολογικό μονοπάτι και τα αποθηκευμένα assets/ChromaDB για το λεξικό και σημασιολογικό μονοπάτι αντίστοιχα. Τα αποτελέσματα, που περιλαμβάνουν αποσπάσματα κειμένου και συνδέσμους προς τα πρωτότυπα PDF στο GCS, μορφοποιούνται σε Markdown και επιστρέφονται στη διεπαφή Gradio για εμφάνιση στον χρήστη.

Το διάγραμμα οπτικοποιεί τη ροή: ο χρήστης (Gradio) → Python backend → αναζήτηση σε ChromaDB & TF-IDF → ανάκτηση σχετικών chunks PDF → LLM (GreekBERT / Meltemi / KriKri) → απάντηση πίσω στο frontend.



Σχήμα 7.1: Αρχιτεκτονική συστήματος

7.3 Hugging Face Space με Persistent Storage

Η πλατφόρμα Hugging Face Spaces επιλέχθηκε για την ανάπτυξη (deployment) και φιλοξενία των τριών παρουσιών της εφαρμογής του chatbot. Η επιλογή αυτή προσφέρει σημαντικά πλεονεκτήματα:

- **Ευκολία Ανάπτυξης για Εφαρμογές Gradio**: Το Hugging Face Spaces είναι βελτιστοποιημένο για τη φιλοξενία εφαρμογών που βασίζονται σε δημοφιλείς βιβλιοθήκες δημιουργίας web interfaces για μηχανική μάθηση, όπως το Gradio.
- **Ενσωμάτωση με το Οικοσύστημα Hugging Face**: Παρέχει άμεση πρόσβαση σε μοντέλα και tokenizers που φιλοξενούνται στο Hugging Face Hub.
- **Δωρεάν Βαθμίδα :** Προσφέρει μια δωρεάν βαθμίδα με επαρκείς υπολογιστικούς πόρους CPU και περιορισμένη πρόσβαση σε GPU για την ανάπτυξη και επίδειξη πρωτότυπων εφαρμογών. Στην συγκεκριμένη εργασία χρησιμοποιήθηκαν GPU όπως οι Nvidia A10G large και Nvidia 1xL4 για την ταχύτερη απόδοση των LLM.
- **Διαχείριση Εκδόσεων (Version Control):** Κάθε Space είναι ουσιαστικά ένα Git repository, επιτρέποντας την εύκολη διαχείριση εκδόσεων του κώδικα. Αξιοποιήθηκε στο έπακρο καθώς πραγματοποιήθηκαν ποικίλες δοκιμές με διαφορετικές παραμέτρους στα διάφορα μοντέλα.

Επίσης, ένα κρίσιμο χαρακτηριστικό που αξιοποιήθηκε είναι το Persistent Storage των Hugging Face Spaces. Στον κώδικα, η μεταβλητή PERSISTENT_STORAGE_ROOT = Path("/data") αναφέρεται σε αυτή τη μόνιμη περιοχή αποθήκευσης. Εντός αυτού του χώρου αποθηκεύτηκαν:

1. Οι βάσεις δεδομένων ChromaDB για κάθε μοντέλο. Η ChromaDB αποθηκεύει εδώ τα ευρετήρια των embeddings και τα σχετικά δεδομένα.
2. Τα "assets" που απαιτούνται για τη λεξική αναζήτηση και τη λειτουργία της εφαρμογής. Αυτά περιλαμβάνουν τους σειριοποιημένους HashingVectorizers (char_vectorizer.joblib, word_vectorizer.joblib), τους πινακες TF-IDF (X_char_sparse.npz, X_word_sparse.npz), καθώς και τις λίστες με τα προ επεξεργασμένα (pre_chunks.pkl) και ακατέργαστα (raw_chunks.pkl) τμήματα κειμένου, τα αναγνωριστικά τους (ids.pkl) και τα μεταδεδομένα τους (metas.pkl).

Η χρήση του Persistent Storage είναι θεμελιώδους σημασίας, καθώς επιτρέπει στα αποτελέσματα της χρονοβόρας διαδικασίας setup_database_and_assets() ,δηλαδή την ευρετηρίαση των δεδομένων και τη δημιουργία των lexical assets να διατηρούνται μεταξύ των επανεκκινήσεων της εφαρμογής. Χωρίς αυτό, η ευρετηρίαση θα έπρεπε να εκτελείται κάθε φορά που ξεκινούσε το Space, καθιστώντας την εφαρμογή μη πρακτική, ειδικά με μεγάλα μοντέλα όπως το Meltemi και το KriKri.

7.4 Διαχείριση μοντέλου

Η διαχείριση των τριών διαφορετικών Μεγάλων Γλωσσικών Μοντέλων εντός του περιβάλλοντος των Hugging Face Spaces πραγματοποιήθηκε μέσω της φόρτωσής τους απευθείας από το Hugging Face Hub κατά την αρχικοποίηση κάθε αντίστοιχης εφαρμογής. Όπως αναφέρθηκε νωρίτερα, χρησιμοποιήθηκαν οι κλάσεις AutoTokenizer και AutoModel της βιβλιοθήκης transformers.

Για κάθε μοντέλο, ορίστηκε το αντίστοιχο MODEL_NAME, δηλαδή "nlpaueb/bert-base-greek-uncased-v1","ilsp/Meltemi-7B-Instruct-v1.5"και"ilsp/Llama-Krikri-8B-Base ". Οι συγκεκριμένες ρυθμίσεις φόρτωσης που εφαρμόστηκαν για κάθε μοντέλο είχαν ως στόχο την ισορροπία μεταξύ απόδοσης και διαχείρισης των περιορισμένων υπολογιστικών πόρων ενός Space:

- **GreekBERT**: Φορτώθηκε με τυπικές ρυθμίσεις και μεταφέρθηκε στην κατάλληλη υπολογιστική μονάδα (DEVICE).

- **Meltemi**: Χρησιμοποιήθηκε torch_dtype=torch.float16 για μείωση της χρήσης VRAM και trust_remote_code=True.
- **KriKri (Llama)**: Αξιοποιήθηκε το device_map="auto" για αυτόματη κατανομή του μοντέλου και επίσης trust_remote_code=True.

Κάθε μία από τις τρεις παρουσίες της εφαρμογής (μία για κάθε LLM) ήταν αυτόνομη, φορτώνοντας το δικό της μοντέλο, tokenizer, και τα αντίστοιχα ευρετήρια από τον Persistent Storage. Αυτή η προσέγγιση επέτρεψε την καθαρή σύγκριση της απόδοσης κάθε μοντέλου χωρίς αλληλεπιδράσεις.

7.5 Google Cloud Storage bucket

Για την παροχή πρόσβασης στα πρωτότυπα αρχεία PDF των προκηρύξεων, αποφασίστηκε η χρήση μιας εξωτερικής υπηρεσίας αποθήκευσης αντικειμένων (object storage) αντί της απευθείας φιλοξενίας τους εντός του Hugging Face Space. Επιλέχθηκε το Google Cloud Storage (GCS) για τον σκοπό αυτό.

Δημιουργήθηκε ένα GCS bucket με το όνομα chatbotthesisihu (όπως ορίζεται από τη μεταβλητή GCS_BUCKET_NAME). Τα 114 αρχεία PDF που συλλέχθηκαν από τη "ΔΙΑΥΓΕΙΑ" μεταφορτώθηκαν σε αυτό το bucket. Ορίστηκε επίσης ένα βασικό πρόθεμα για τους δημόσιους συνδέσμους πρόσβασης (GCS_PUBLIC_URL_PREFIX = f"https://storage.googleapis.com/{GCS_BUCKET_NAME}/").

Η χρήση του GCS για αυτόν τον σκοπό προσφέρει αρκετά πλεονεκτήματα και επιλέχθηκε με γνώμονα κάποιους από αυτούς :

- **Αποφόρτιση του Hugging Face Space**: Αποφεύγεται η αποθήκευση μεγάλων αρχείων PDF στον περιορισμένο αποθηκευτικό χώρο του Space.
- **Υψηλή Διαθεσιμότητα και Ταχύτητα**: Το GCS παρέχει αξιόπιστη και γρήγορη πρόσβαση στα αρχεία από οπουδήποτε.
- **Εύκολη Διαχείριση Δημόσιων Συνδέσμων**: Η δημιουργία δημόσιων συνδέσμων είναι απλή, διευκολύνοντας την ενσωμάτωση στην εφαρμογή.

Παρακάτω βρίσκεται ένα μέρος με το bucket των PDF στο Google Cloud Storage:

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryptio
60Ξ046M9ΞH-IΓΩ.pdf	906.2 KB	application/pdf	May 14, 2025, 3:37:59 PM	Standard	May 14, 2025, 3:37:59 PM	Public to internet Copy URL	–	Google-m
60ΨΦ469B7Γ-PΞΦ.pdf	3.1 MB	application/pdf	May 14, 2025, 3:38:10 PM	Standard	May 14, 2025, 3:38:10 PM	Public to internet Copy URL	–	Google-m
617N46ΨZ2N-E28.pdf	151.1 KB	application/pdf	May 14, 2025, 3:38:21 PM	Standard	May 14, 2025, 3:38:21 PM	Public to internet Copy URL	–	Google-m
61AH46M9ΞH-4X1.pdf	801.3 KB	application/pdf	May 14, 2025, 3:38:03 PM	Standard	May 14, 2025, 3:38:03 PM	Public to internet Copy URL	–	Google-m
627Σ46M9ΞH-ΠΞΛ.pdf	533.6 KB	application/pdf	May 14, 2025, 3:38:24 PM	Standard	May 14, 2025, 3:38:24 PM	Public to internet Copy URL	–	Google-m
62Ω946M9ΞH-38P.pdf	575.7 KB	application/pdf	May 14, 2025, 3:38:04 PM	Standard	May 14, 2025, 3:38:04 PM	Public to internet Copy URL	–	Google-m
63A646ΨΣ4-ΦΥΔ.pdf	683 KB	application/pdf	May 14, 2025, 3:38:07 PM	Standard	May 14, 2025, 3:38:07 PM	Public to internet Copy URL	–	Google-m
63Σ7469B7H-0X6.pdf	400 KB	application/pdf	May 14, 2025, 3:38:07 PM	Standard	May 14, 2025, 3:38:07 PM	Public to internet Copy URL	–	Google-m
63Φ0469B7Ξ-6NO.pdf	845.8 KB	application/pdf	May 14, 2025, 3:38:10 PM	Standard	May 14, 2025, 3:38:10 PM	Public to internet Copy URL	–	Google-m
65NM46M9ΞH-Ξ3A.pdf	420.6 KB	application/pdf	May 14, 2025, 3:38:10 PM	Standard	May 14, 2025, 3:38:10 PM	Public to internet Copy URL	–	Google-m
69AK46Ψ8XB-Σ10.pdf	1.3 MB	application/pdf	May 14, 2025, 3:38:16 PM	Standard	May 14, 2025, 3:38:16 PM	Public to internet Copy URL	–	Google-m
6AT0469B7H-963.pdf	1 MB	application/pdf	May 14, 2025, 3:36:52 PM	Standard	May 14, 2025, 3:36:52 PM	Public to internet Copy URL	–	Google-m
6ΓΞΠ469B7Ξ-ΦH3.pdf	784.3 KB	application/pdf	May 14, 2025, 3:36:50 PM	Standard	May 14, 2025, 3:36:50 PM	Public to internet Copy URL	–	Google-m
6ZOP469B6N-03I.pdf	333.4 KB	application/pdf	May 14, 2025, 3:36:49 PM	Standard	May 14, 2025, 3:36:49 PM	Public to internet Copy URL	–	Google-m
6HAΞ46Ψ8Z6-EΞΣ.pdf	1.7 MB	application/pdf	May 14, 2025, 3:36:57 PM	Standard	May 14, 2025, 3:36:57 PM	Public to internet Copy URL	–	Google-m
6HBK469B7Δ-4KH.pdf	1 MB	application/pdf	May 14, 2025, 3:36:56 PM	Standard	May 14, 2025, 3:36:56 PM	Public to internet Copy URL	–	Google-m
6HZΩ469B7Δ-ZM8.pdf	678.7 KB	application/pdf	May 14, 2025, 3:36:56 PM	Standard	May 14, 2025, 3:36:56 PM	Public to internet Copy URL	–	Google-m
6HIY469B7H-ΣΣΦ.pdf	350.1 KB	application/pdf	May 14, 2025, 3:36:58 PM	Standard	May 14, 2025, 3:36:58 PM	Public to internet Copy URL	–	Google-m
6Θ8Γ46ΨZΣ4-90Z.pdf	1.5 MB	application/pdf	May 14, 2025, 3:37:03 PM	Standard	May 14, 2025, 3:37:03 PM	Public to internet Copy URL	–	Google-m
60KI46ΨZΣ4-Ξ27.pdf	656.1 KB	application/pdf	May 14, 2025, 3:37:00 PM	Standard	May 14, 2025, 3:37:00 PM	Public to internet Copy URL	–	Google-m
60Π=469B7T-YΦP.pdf	273.7 KB	application/pdf	May 14, 2025, 3:37:00 PM	Standard	May 14, 2025, 3:37:00 PM	Public to internet Copy URL	–	Google-m
6ΛHA46ΨZΣ4-Λ0K.pdf	707.5 KB	application/pdf	May 14, 2025, 3:37:03 PM	Standard	May 14, 2025, 3:37:03 PM	Public to internet Copy URL	–	Google-m
6=6T469B4M-2N6.pdf	825.4 KB	application/pdf	May 14, 2025, 3:37:04 PM	Standard	May 14, 2025, 3:37:04 PM	Public to internet Copy URL	–	Google-m
6=HA469B7Δ-ZEM.pdf	858.3 KB	application/pdf	May 14, 2025, 3:37:07 PM	Standard	May 14, 2025, 3:37:07 PM	Public to internet Copy URL	–	Google-m
6P6B46Ψ8XB-0ΣΓ.pdf	1.7 MB	application/pdf	May 14, 2025, 3:37:11 PM	Standard	May 14, 2025, 3:37:11 PM	Public to internet Copy URL	–	Google-m
6TΓ6469B7A-YNΓ.pdf	290.3 KB	application/pdf	May 14, 2025, 3:37:06 PM	Standard	May 14, 2025, 3:37:06 PM	Public to internet Copy URL	–	Google-m
6TPΓ469B7H-ΣOO.pdf	435.6 KB	application/pdf	May 14, 2025, 3:37:09 PM	Standard	May 14, 2025, 3:37:09 PM	Public to internet Copy URL	–	Google-m

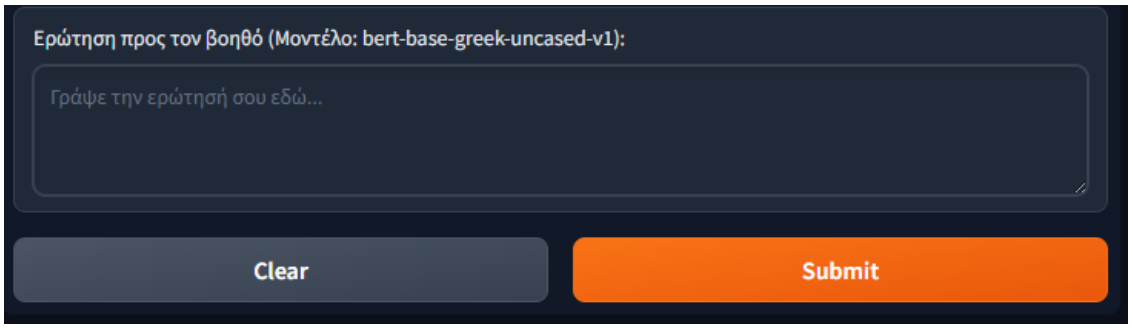
Σχήμα 7.2: Google Cloud Storage και τα PDFs

7.6 Gradio frontend

Η διεπαφή χρήστη (frontend) της εφαρμογής αναπτύχθηκε εξ ολοκλήρου με τη χρήση της βιβλιοθήκης Gradio. Το Gradio επιτρέπει τη γρήγορη δημιουργία απλών web interfaces για μοντέλα μηχανικής μάθησης και εφαρμογές Python, χωρίς την ανάγκη για εκτενή γνώση τεχνολογιών web development.

Η δομή της διεπαφής, όπως ορίζεται στην κλήση gr.Interface(...) στον κώδικα, περιλαμβάνει τα εξής κύρια στοιχεία:

- **Είσοδος Χρήστη (Input)**: Ένα πεδίο κειμένου πολλαπλών γραμμών (gr.Textbox) με placeholder "Γράψε την ερώτησή σου εδώ..." και κατάλληλη ετικέτα που αναφέρει το ενεργό μοντέλο (π.χ., "Ερώτηση προς τον βοηθό (Μοντέλο: Meltemi-7B-Instruct-v1.5):"). Εδώ ο χρήστης εισάγει το ερώτημά του σε φυσική ελληνική γλώσσα.

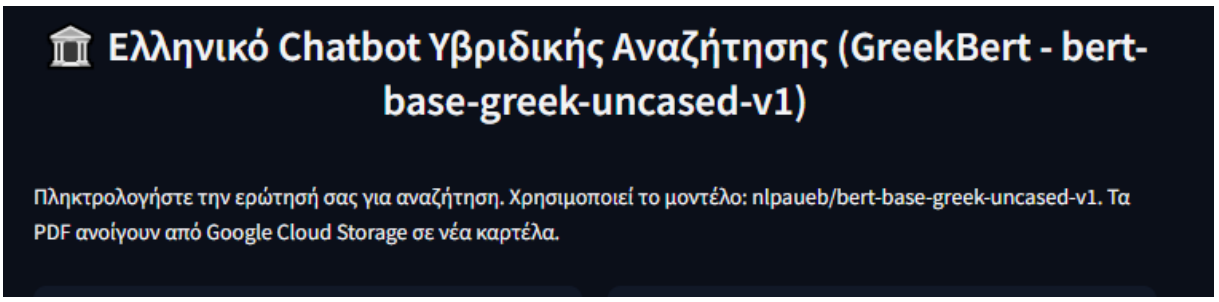


Εικόνα 7.3: Είσοδος Χρήστη Gradio

- **Έξοδος Αποτελεσμάτων (Output)**: Ένα πεδίο τύπου Markdown (gr.Markdown) με ετικέτα "Απαντήσεις από τα έγγραφα:". Το Markdown επιλέχθηκε για την ευελιξία του στην παρουσίαση μορφοποιημένου κειμένου, συμπεριλαμβανομένων τίτλων, λιστών και, κυρίως,

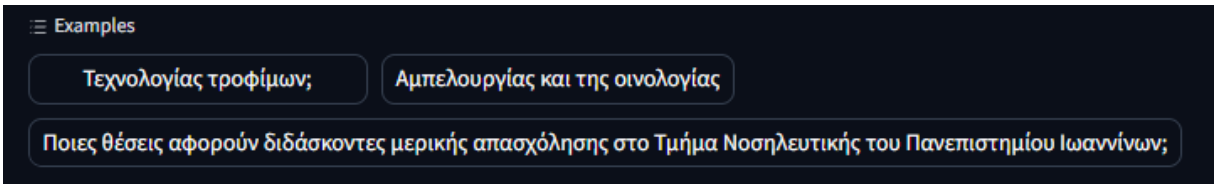
υπερσυνδέσμων. Η ρύθμιση sanitize_html=False είναι σημαντική καθώς επιτρέπει την απόδοση των HTML tags για τους συνδέσμους προς τα αρχεία PDF.

- **Τίτλος και Περιγραφή Εφαρμογής**: Ένας περιγραφικός τίτλος όπως "🏛️ Ελληνικό Chatbot Υβριδικής Αναζήτησης (Meltemi - ...)" και μια σύντομη περιγραφή που ενημερώνει τον χρήστη για τη λειτουργία της εφαρμογής, το χρησιμοποιούμενο μοντέλο και τον τρόπο πρόσβασης στα PDF.



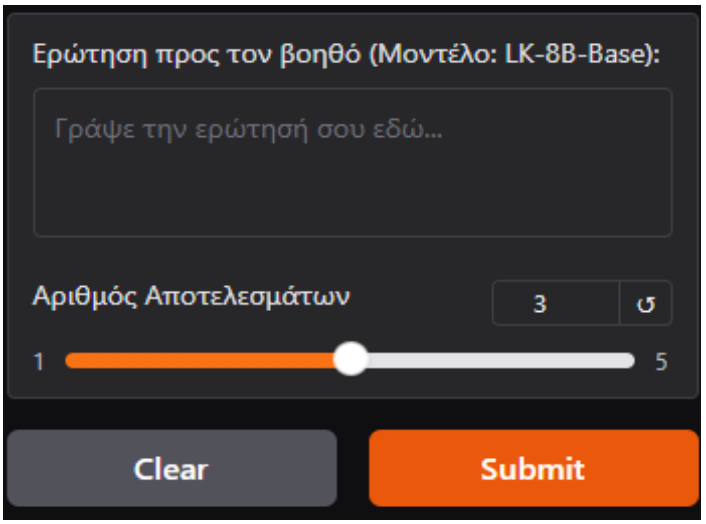
Σχήμα 7.4: Τίτλος και Περιγραφή Εφαρμογής

- **Παραδείγματα Ερωτήσεων (Examples)**: Μια λίστα από προκαθορισμένα παραδείγματα ερωτήσεων,παρέχεται για να βοηθήσει τον χρήστη να κατανοήσει το είδος των ερωτημάτων που μπορεί να υποβάλει και να δοκιμάσει γρήγορα την εφαρμογή.



Σχήμα 7.5: Παραδείγματα Ερωτήσεων

- **Επιλογή Αριθμού Αποτελεσμάτων:** Ελεύθερη επιλογή του χρήστη ως προς πόσα αποτελέσματα έχει να δει. Δίνεται η επιλογή από 1 έως 5 αποτελέσματα με ένα ειδικό slider button.



Σχήμα 7.6: Επιλογή Αριθμού Αποτελεσμάτων

7.7 Περιβάλλον εκτέλεσης & απαιτήσεις VRAM

Η εφαρμογή εκτελείται στο περιβάλλον των Hugging Face Spaces, το οποίο παρέχει έναν προρυθμισμένο χώρο εκτέλεσης για εφαρμογές Python. Οι βασικές εξαρτήσεις της εφαρμογής (Python βιβλιοθήκες όπως transformers, torch, gradio, chromadb, scikit-learn, PyPDF2, joblib, pickle, tqdm, unicodedata) δηλώνονται σε ένα αρχείο requirements.txt στο αποθετήριο του Space, ώστε το περιβάλλον να εγκαθιστά αυτόματα τις απαραίτητες εκδόσεις.

Η επιλογή της υπολογιστικής μονάδας (DEVICE) γίνεται δυναμικά κατά την εκκίνηση ("cuda" if torch.cuda.is_available() else "cpu"). Για την αποτελεσματική λειτουργία, ειδικά με τα μεγαλύτερα μοντέλα Meltemi και KriKri , η διαθεσιμότητα GPU στο Hugging Face Space ήταν κρίσιμης σημασίας.

- **GreekBERT (base model)**: Είναι το ελαφρύτερο από τα τρία και μπορεί συνήθως να λειτουργήσει σε GPU με μέτρια VRAM (8-16GB) με λογικό BATCH_EMB (32 κατά το setup).
- **Meltemi και KriKri**: Απαιτούν σημαντικά περισσότερη VRAM. Η χρήση torch.float16 [29] για το Meltemi μειώνει τις απαιτήσεις κατά το μισό. Η στρατηγική device_map="auto" για το KriKri επιτρέπει την κατανομή του μοντέλου σε πολλαπλές GPU ή μεταξύ CPU και GPU, καθιστώντας δυνατή τη φόρτωσή του ακόμα και αν δεν χωράει ολόκληρο σε μία μόνο GPU. Τα μικρότερα BATCH_EMB δηλαδή 8 για Meltemi και 4 για KriKri που χρησιμοποιήθηκαν κατά τη φάση της δημιουργίας των embeddings αντανακλούν αυτές τις αυξημένες απαιτήσεις. Κατά τη φάση της αναζήτησης (inference) για ένα μεμονωμένο ερώτημα, οι απαιτήσεις VRAM είναι χαμηλότερες από αυτές της παραγωγής embeddings, αλλά η φόρτωση του ίδιου του μοντέλου παραμένει η κύρια πρόκληση με την διάρκεια του να είναι κοντά στα 15 λεπτά..

7.8 Επίλογος

Η σχεδίαση και υλοποίηση της εφαρμογής του chatbot, όπως περιγράφηκε στο παρόν κεφάλαιο, εστίασε στη δημιουργία ενός λειτουργικού, προσβάσιμου και επεκτάσιμου συστήματος. Η επιλογή της πλατφόρμας Hugging Face Spaces, σε συνδυασμό με τη βιβλιοθήκη Gradio για τη διεπαφή χρήστη, επέτρεψε την ταχεία ανάπτυξη τριών ξεχωριστών παρουσιών της εφαρμογής, καθεμία εκ των οποίων αξιοποιεί ένα διαφορετικό ελληνικό Μεγάλο Γλωσσικό Μοντέλο (GreekBERT, Meltemi, KriKri). Η χρήση του Persistent Storage διασφαλίζει τη μόνιμη αποθήκευση των κρίσιμων ευρετηριασμένων δεδομένων, καθιστώντας την εφαρμογή πρακτική για επαναλαμβανόμενη χρήση. Η ενσωμάτωση του Google Cloud Storage για τη φιλοξενία των πρωτότυπων εγγράφων PDF παρέχει μια αποδοτική λύση για την παροχή πρόσβασης στις πηγές. Η διαχείριση των μοντέλων και οι εκτιμήσεις για τις απαιτήσεις σε υπολογιστικούς πόρους, ειδικά VRAM, αναδεικνύουν τις πρακτικές προκλήσεις της εργασίας με μεγάλα γλωσσικά μοντέλα. Συνολικά, το κεφάλαιο αυτό κάλυψε τις βασικές πτυχές της μετατροπής των ερευνητικών ευρημάτων και των αλγορίθμων σε μια απτή εφαρμογή, έτοιμη για αξιολόγηση και χρήση.

Κεφάλαιο 8ο: Πειραματική Αξιολόγηση & Benchmarks

8.1 Εισαγωγη

Μετά την υλοποίηση του συστήματος υβριδικής αναζήτησης και την ανάπτυξη των τριών παρουσιών της εφαρμογής (μία για κάθε επιλεγμένο Μεγάλο Γλωσσικό Μοντέλο: GreekBERT, Meltemi και KriKri), το παρόν κεφάλαιο επικεντρώνεται στην πειραματική αξιολόγηση της απόδοσής τους. Στόχος της αξιολόγησης είναι η ποσοτική και ποιοτική σύγκριση των τριών μοντέλων στην εργασία της ανάκτησης σχετικών προκηρύξεων εντεταλμένων διδασκόντων, βάσει ερωτημάτων διατυπωμένων σε φυσική ελληνική γλώσσα. Για τον σκοπό αυτό, ορίστηκε ένα σύνολο ερωτήσεων, επιλέχθηκαν κατάλληλες μετρικές αξιολόγησης. Τα αποτελέσματα αναλύονται διεξοδικά, συμπεριλαμβανομένης μιας ποιοτικής ανάλυσης. Επιπλέον, συζητείται η σημασία της ανάλυσης ευαισθησίας για βασικές παραμέτρους του συστήματος, όπως ο συντελεστής στάθμισης α της υβριδικής αναζήτησης. Τέλος, γίνεται μια συνολική συζήτηση των αποτελεσμάτων, αναδεικνύοντας τα πλεονεκτήματα και τα μειονεκτήματα κάθε μοντέλου στο συγκεκριμένο πλαίσιο εφαρμογής.

8.2 Σύνολο ερωτήσεων

Για την αξιολόγηση της απόδοσης των τριών υλοποιημένων συστημάτων, δημιουργήθηκε ένα σύνολο 30 αντιπροσωπευτικών ερωτήσεων. Οι ερωτήσεις αυτές σχεδιάστηκαν ώστε να καλύπτουν ένα ευρύ φάσμα πληροφοριακών αναγκών που μπορεί να έχει ένας χρήστης ο οποίος αναζητά προκηρύξεις εντεταλμένων διδασκόντων. Περιλαμβάνουν:

- Ερωτήσεις με συγκεκριμένες λέξεις-κλειδιά ("Τεχνολογίας τροφίμων;", "Αμπελουργίας και της οινολογίας").
- Ερωτήσεις που αφορούν συγκεκριμένα τμήματα και πανεπιστήμια ("Ποιες θέσεις αφορούν διδάσκοντες μερικής απασχόλησης στο Τμήμα Νοσηλευτικής του Πανεπιστημίου Ιωαννίνων;").
- Θεματικές αναζητήσεις ("Εχουμε κατι με μουσικη?", "Σχετικα με το μαθημα της γεωλογιας").
- Ερωτήσεις με γεωγραφικούς περιορισμούς ("Κατι που ειναι στην Κρητη").
- Ερωτήσεις που βασίζονται σε ημερομηνίες ("Εχουμε τιποτα που αναρτηθηκε στις 13/01/2025?").
- Ερωτήσεις σχετικά με απαιτήσεις και προϋποθέσεις ("Ποιες προκηρύξεις απαιτούν διδακτορικό τίτλο και τουλάχιστον 5 έτη διδακτικής εμπειρίας;", "Είναι απαραίτητη η κατάθεση της διδακτορικής διατριβής στο Εθνικό Αρχείο Διδακτορικών Διατριβών;").
- Ερωτήσεις που αναφέρονται σε νομικές διατάξεις ή κωδικούς ("Θελω την Διαταξη 49572022", "Περιλαμβάνει η προκήρυξη το άρθρο 173 του Ν. 49572022;", "Αναφέρεται ο Κωδικός MIS της πράξης;").
- Ερωτήσεις σχετικά με τη χρηματοδότηση ("Χρηματοδοτείται από το ΕΚΤ;", "Ποιες προκηρύξεις χρηματοδοτούνται από το Πρόγραμμα “Ανθρώπινο Δυναμικό και Κοινωνική Συνοχή 2021-2027”;").

- Ερωτήσεις για λεπτομέρειες της θέσης ("Ποια είναι η διάρκεια της σύμβασης;", "Περιλαμβάνονται επαναληπτικές εξετάσεις Σεπτεμβρίου στις υποχρεώσεις;", "Η θέση χαρακτηρίζεται ως μερικής απασχόλησης;").

Για κάθε μία από αυτές τις 30 ερωτήσεις, και για κάθε ένα από τα τρία συστήματα, καταγράφηκαν τα κορυφαία 5 επιστρεφόμενα τμήματα εγγράφων (chunks). Στη συνέχεια, αυτά τα αποτελέσματα υποβλήθηκαν αξιολόγηση συνάφειας.

8.3 Μετρικές Αξιολόγησης

Η απόδοση των συστημάτων ανάκτησης πληροφορίας αξιολογήθηκε με βάση την ικανότητά τους να επιστρέφουν σχετικά έγγραφα και να τα κατατάσσουν σε υψηλές θέσεις. Για τον σκοπό αυτό, εφαρμόστηκε μια διαδικασία χειροκίνητης αξιολόγησης συνάφειας και χρησιμοποιήθηκαν οι ακόλουθες ποσοτικές μετρικές:

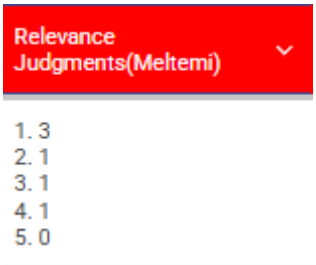
8.3.1 Χειροκίνητη Αξιολόγηση Συνάφειας

Για κάθε ερώτηση και για κάθε ένα από τα 5 κορυφαία αποτελέσματα που επέστρεψε κάθε μοντέλο (LLM), πραγματοποιήθηκε χειροκίνητη αξιολόγηση της συνάφειας του αποτελέσματος με την αρχική ερώτηση. Η συνάφεια βαθμολογήθηκε σε μια κλίμακα με τέσσερις βαθμούς:

- **0:** Μη σχετική, το αποτέλεσμα δεν έχει καμία σχέση με την ερώτηση.
- **1:** Οριακά σχετική, το αποτέλεσμα έχει κάποια έμμεση ή πολύ μικρή σχέση.
- **2:** Σχετική, το αποτέλεσμα είναι σαφώς σχετικό και απαντά εν μέρει ή παρέχει χρήσιμες πληροφορίες για την ερώτηση.
- **3:** Άκρως σχετική, το αποτέλεσμα απαντά άμεσα και πλήρως στην ερώτηση ή είναι εξαιρετικά συναφές.

Η αξιολόγηση πραγματοποιήθηκε με βάση τα παραπάνω προκαθορισμένα κριτήρια συνάφειας, εξασφαλίζοντας όσο το δυνατόν μεγαλύτερη αντικειμενικότητα και συγκρισιμότητα [30]. Η κρίση για κάθε απάντηση έγινε σύμφωνα με την κλίμακα συνάφειας, ενώ ιδιαίτερη προσοχή δόθηκε στη διατήρηση σταθερής προσέγγισης κατά την αντιπαραβολή των ερωτήσεων με τα αντίστοιχα αποτελέσματα σε όλα τα εξεταζόμενα αρχεία PDF, ώστε να διασφαλιστεί η συνέπεια και η αξιοπιστία της διαδικασίας αυτής.

Με βάση αυτό, για παράδειγμα μία ερώτηση και μια απάντηση από το Meltemi ήταν καταγεγραμμένο με αυτον τον τροπο, όπου είναι οι πέντε σχετικές απαντήσεις με την αξιολόγηση συνάφειας μας.



Σχήμα 8.1: Relevance Judgments (Meltemi) σε μία ερώτηση

8.3.2 Cumulative Gain (CG@k)

Το CG μετρά το άθροισμα των βαθμών συνάφειας των k πρώτων αποτελεσμάτων, χωρίς να λαμβάνει υπόψη τη θέση τους. Η φόρμουλα είναι:

$$CG@k = rel_1 + rel_2 + ... + rel$$

(8.1)

όπου rel_1 ο βαθμός συνάφειας του αποτελέσματος στη θέση 1. Στην παρούσα αξιολόγηση, υπολογίστηκε το CG@5, για κάθε ερώτηση ξεχωριστά καθώς και για κάθε LLM. Με αυτόν τον τρόπο υπολογίστηκε και το Mean Cumulative Gain (MCG@5), που ουσιαστικά είναι η διαίρεση του με τον συνολικό αριθμό των ερωτήσεων, αριθμό 30.

8.3.3 Discounted Cumulative Gain (DCG@k)

Το DCG είναι μια βελτίωση του CG που **λαμβάνει υπόψη τη θέση των αποτελεσμάτων**, δίνοντας μεγαλύτερη βαρύτητα στα αποτελέσματα που εμφανίζονται υψηλότερα στην κατάταξη. Ο υπολογισμός γίνεται με τον τύπο:

$$DCGk = \sum_{i=1}^{k} \frac{reli}{log(i+1)}$$

(8.2)

όπου Σ άθροισμα, reli βαθμολογία σχετικότητας, και log(i+1) λογάριθμος με βάση 2 της θέσης i συν ένα. Για το πρώτο αποτέλεσμα (i=1), ο παρονομαστής είναι log(1+1) = log(2)=1 και για το δεύτερο αποτέλεσμα (i=2), ο παρονομαστής είναι log(2+1) = log(3) ≈ 1.58. Με αυτόν τον τρόπο, Καθώς το i αυξάνεται, το log(i+1) επίσης αυξάνεται, κάνοντας τον παρονομαστή μεγαλύτερο. Ένα έγγραφο με υψηλή συνάφεια είναι πιο χρήσιμο αν εμφανίζεται νωρίς με αυτόν τον τρόπο.

8.3.4 Ideal Discounted Cumulative Gain (IDCG@k)

Το IDCG@k αντιπροσωπεύει τη μέγιστη δυνατή τιμή DCG@k για μια δεδομένη ερώτηση, η οποία θα επιτυγχανόταν εάν τα αποτελέσματα ήταν ταξινομημένα με τέλειο τρόπο δηλαδή, τα πιο σχετικά πρώτα.

Για παράδειγμα:

1. Παίρνουμε τις βαθμολογίες σχετικότητας που έδωσε το σύστημα για αυτή την ερώτηση: [2, 1, 3, 0, 0].
2. Ταξινομούμε αυτές τις βαθμολογίες σε φθίνουσα σειρά: [3, 2, 1, 0, 0].
3. Υπολογίζουμε το DCG@5 για αυτή την ιδανική σειρά:
    - Απάντηση στη θέση 1 (ιδανική): rel1=3. Discount = log2(1+1)=1. Συνεισφορά στο IDCG: 3/1=3
    - Απάντηση στη θέση 2 (ιδανική): rel2=2. Discount = log2(2+1)≈1.585. Συνεισφορά στο IDCG: 2/1.585≈1.2619

- Απάντηση στη θέση 3 (ιδανική): rel3=1. Discount = log2(3+1)=2. Συνεισφορά στο IDCG: 1/2=0.5
- Απάντηση στη θέση 4 (ιδανική): rel4=0. Discount = log2(4+1)≈2.322. Συνεισφορά στο IDCG: 0/2.322=0
- Απάντηση στη θέση 5 (ιδανική): rel5=0. Discount = log2(5+1)≈2.585. Συνεισφορά στο IDCG: 0/2.585=0

Άρα, το IDCG@5 για αυτή την ερώτηση και με βάση τις απαντήσεις που έδωσε το σύστημα είναι: IDCG@5=3+1.2619+0.5+0+0=4.7619

8.3.5 Normalized Discounted Cumulative Gain (nDCG@k)

Το nDCG κανονικοποιεί το DCG διαιρώντας το με το IDCG, με αποτέλεσμα μια μετρική που κυμαίνεται από 0 έως 1. Επιτρέπει τη σύγκριση της απόδοσης μεταξύ διαφορετικών ερωτήσεων. Ο τύπος είναι:

$$nDCGk = \frac{DCGk}{IDCGk} \tag{8.3}$$

Για παράδειγμα:

DCG@K = 4.5

IDCG@k = 4.8928 αρα, nDCGk= 0.9197 που προκύπτει από την μεταξύ τους διαίρεση.

8.3.6 Reciprocal Rank (RR) και Mean Reciprocal Rank (MRR)

Για τον υπολογισμό του MRR, οι αρχικές τετράβαθμες αξιολογήσεις συνάφειας μετατράπηκαν σε δυαδικές:

- Αποτελέσματα με βαθμό 2 (Σχετικά) και 3 (Άκρως σχετικά) θεωρήθηκαν 1.
- Αποτελέσματα με βαθμό 0 (Μη σχετικά) και 1 (Οριακά σχετικά) θεωρήθηκαν 0.

Ο Αντίστροφος Βαθμός (Reciprocal Rank - RR) για μια ερώτηση είναι ο αντίστροφος της θέσης του πρώτου σχετικού εγγράφου που επιστράφηκε. Εάν δεν επιστραφεί κανένα σχετικό έγγραφο στα top-k αποτελέσματα, το RR είναι 0.

Για παράδειγμα:

Εάν το σύστημα επέστρεψε [1,0,1,0,0], τοτε

$$RR = \frac{1}{r} \tag{8.4}$$

οπου r ουσιαστικά είναι η θέση που βλέπουμε για πρώτη φορά το 1, αρα για το παράδειγμα μας

RR=1, αντιθέτως εάν το σύστημα επέστρεφε [0,1,0,1,0] τοτε $RR = \frac{1}{2}$, άρα RR = 0.5

Ο Μέσος Αντίστροφος Βαθμός (Mean Reciprocal Rank - MRR) είναι ο μέσος όρος των τιμών RR για το σύνολο των ερωτήσεων:

$$MRR = \frac{1}{Q} \sum_{q=1}^{|Q|} \frac{1}{rankq} \tag{8.5}$$

όπου |Q| είναι ο αριθμός των ερωτήσεων (30) και rankq είναι η θέση του πρώτου σχετικού αποτελέσματος για την ερώτηση q [31][32].

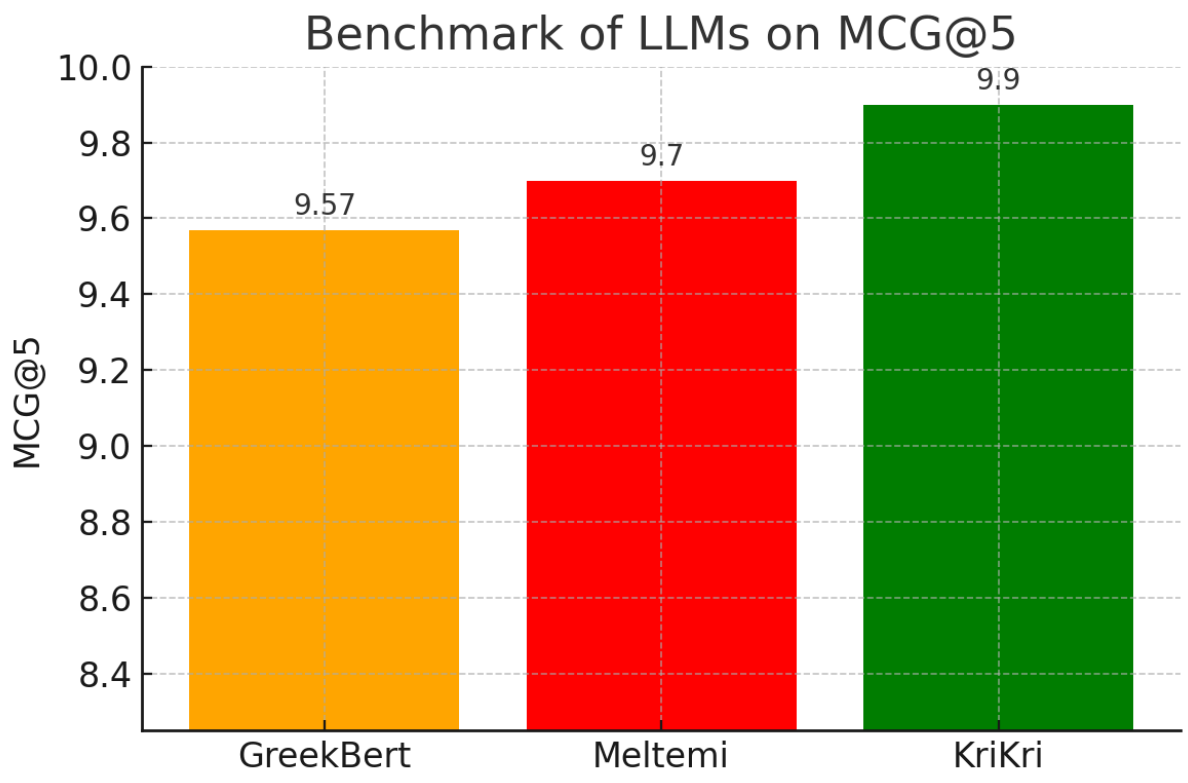
8.4 Αποτελέσματα GreekBERT vs Meltemi vs KriKri

Μετά τη διεξαγωγή των πειραμάτων με το σύνολο των 30 ερωτήσεων και την χειροκίνητη αξιολόγηση των κορυφαίων 5 αποτελεσμάτων για κάθε μοντέλο, υπολογίστηκαν οι μετρικές που περιγράφονται στην ενότητα 8.3.

8.4.1 Cumulative Gain (CG@5) και Mean Cumulative Gain (MCG@5)

Το CG@5 αντιπροσωπεύει το άθροισμα των βαθμών συνάφειας των 5 πρώτων αποτελεσμάτων για όλες τις 30 ερωτήσεις. Το MCG@5 είναι ο μέσος όρος αυτού του αθροίσματος ανά ερώτηση.

- **GreekBERT:** Συνολικό CG@5 = 287, MCG@5 = 287/30≈9.57
- **Meltemi:** Συνολικό CG@5 = 291, MCG@5 = 291/30=9.70
- **KriKri:** Συνολικό CG@5 = 297, MCG@5 = 297/30=9.90



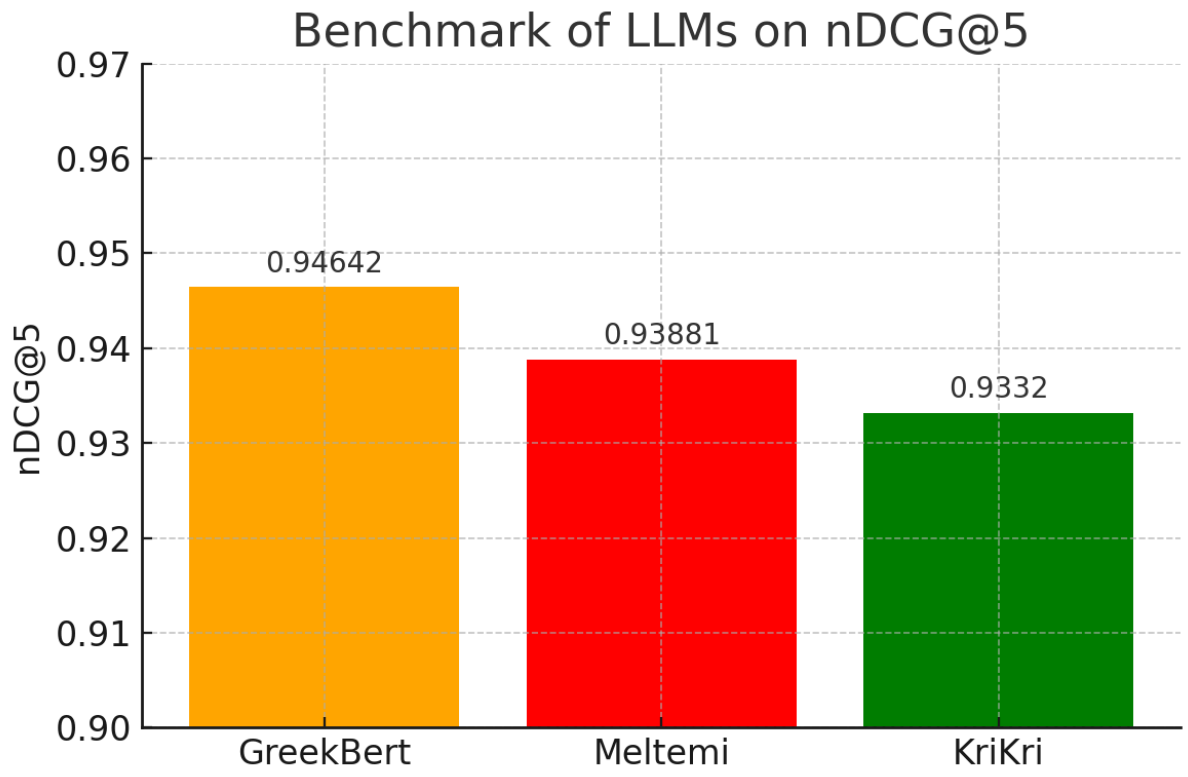
Σχήμα 8.2 : Mean Cumulative Gain (MCG@5)

Από αυτά τα αποτελέσματα, το KriKri φαίνεται να ανακτά συνολικά τα πιο σχετικά έγγραφα στις πρώτες 5 θέσεις, ακολουθούμενο από το Meltemi και τέλος το GreekBERT.

8.4.2 Normalized Discounted Cumulative Gain (nDCG@5)

Το nDCG@5 λαμβάνει υπόψη τόσο τη συνάφεια όσο και τη θέση των αποτελεσμάτων. Ο μέσος όρος nDCG@5 για τις 30 ερωτήσεις ήταν:

- **GreekBERT:** Mean nDCG@5 = 0.94642
- **Meltemi:** Mean nDCG@5 = 0.93881
- **KriKri:** Mean nDCG@5 = 0.93320



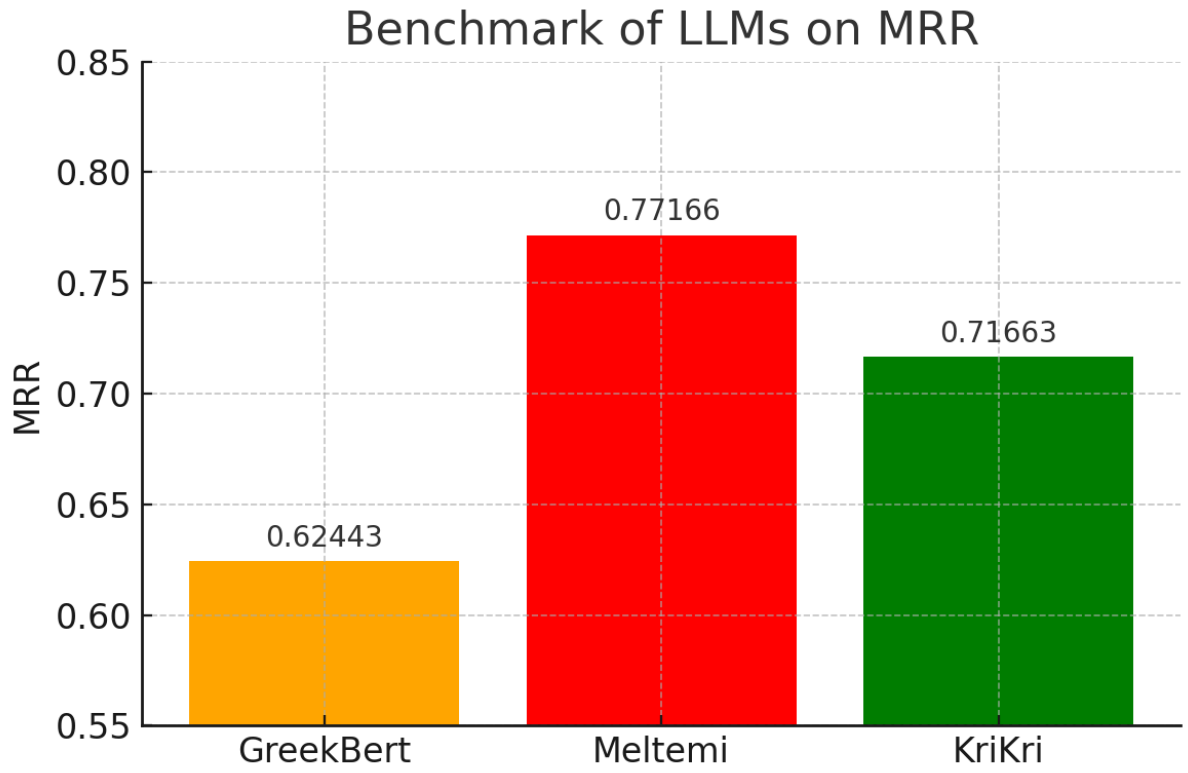
Σχήμα 8.3 : Mean Normalized Discounted Cumulative Gain (nDCG@5)

Εδώ, το GreekBERT παρουσιάζει την υψηλότερη μέση τιμή nDCG@5, υποδεικνύοντας ότι, κατά μέσο όρο, κατατάσσει τα σχετικά έγγραφα σε υψηλότερες θέσεις πιο αποτελεσματικά από τα άλλα δύο μοντέλα.

8.4.3 Mean Reciprocal Rank (MRR)

Το MRR αξιολογεί την ικανότητα του συστήματος να επιστρέφει το πρώτο σχετικό έγγραφο σε όσο το δυνατόν υψηλότερη θέση. Οι δυαδικές αξιολογήσεις συνάφειας (0 ή 1) χρησιμοποιήθηκαν για τον υπολογισμό του RR κάθε ερώτησης.

- **GreekBERT:** Συνολικό άθροισμα RR = 18.733, MRR = 18.733/30≈0.62443
- **Meltemi:** Συνολικό άθροισμα RR = 23.150, MRR = 23.150/30≈0.77166
- **KriKri:** Συνολικό άθροισμα RR = 21.499, MRR = 21.499/30≈0.71663



Σχήμα 8.4 : Mean Reciprocal Rank (MRR)

Το Meltemi πέτυχε το υψηλότερο MRR, υποδεικνύοντας ότι ήταν το πιο αποτελεσματικό στο να φέρνει ένα τουλάχιστον άκρως σχετικό αποτέλεσμα στην κορυφή της λίστας.

8.5 Ποιοτική ανάλυση σφαλμάτων

Πέρα από τις ποσοτικές μετρικές, πραγματοποιήθηκε και μια ποιοτική ανάλυση των αποτελεσμάτων για τον εντοπισμό συχνών τύπων σφαλμάτων ή προβληματικών συμπεριφορών των συστημάτων.

- Σφάλματα Συνάφειας: Σε ορισμένες περιπτώσεις, ειδικά με πιο ασαφή ή υπερβολικά σύντομα ερωτήματα, τα συστήματα επέστρεφαν έγγραφα που ήταν οριακά σχετικά ή άσχετα. Αυτό ήταν λιγότερο πρόβλημα για ερωτήματα με πολύ συγκεκριμένους όρους όπου η λεξική συνιστώσα της υβριδικής αναζήτησης μπορούσε να εντοπίσει ακριβείς αντιστοιχίες.
- Επίδραση "Θορύβου" από την Εξαγωγή Κειμένου: Παρόλο που τα PDF χαρακτηρίστηκαν ως "searchable", η διαδικασία εξαγωγής κειμένου δεν είναι πάντα τέλεια. Λάθη στην αναγνώριση

διάταξης, σπασμένες λέξεις στα όρια γραμμών ή παραγράφων, ή η λανθασμένη ερμηνεία δομημένων στοιχείων όπως πίνακες ως συνεχές κείμενο, μπορούσαν να οδηγήσουν στη δημιουργία "θορυβωδών" chunks. Αυτός ο θόρυβος ενδέχεται να επηρέασε την ποιότητα των embeddings ή τη λεξική αντιστοίχιση, οδηγώντας σε λιγότερο ακριβή αποτελέσματα.

- Διαφορές ανά Μοντέλο: Παρατηρήθηκε ότι τα μεγαλύτερα μοντέλα Meltemi και KriKri έτειναν να είναι καλύτερα στην κατανόηση πιο σύνθετων ή εννοιολογικά διατυπωμένων ερωτημάτων όπου η σημασιολογική ομοιότητα ήταν κυρίαρχη. Το GreekBERT, από την άλλη πλευρά, έδειχνε να υπερτερεί όταν υπήρχε ισχυρή λεξική επικάλυψη μεταξύ ερωτήματος και εγγράφου, πιθανώς λόγω της φύσης της αρχιτεκτονικής του (BERT) που είναι ισχυρή στην κατανόηση πλαισίου λέξεων.

8.6 Ανάλυση στην παράμετρο α

Ο συντελεστής α, ο οποίος καθορίζει τη στάθμιση μεταξύ της σημασιολογικής (α * semantic_score) και της λεξικής ((1-α) * lexical_score) βαθμολογίας στην υβριδική αναζήτηση, αποτελεί μια κρίσιμη παράμετρο του συστήματος. Στην παρούσα υλοποίηση, χρησιμοποιήθηκαν δύο προκαθορισμένες τιμές: ALPHA_BASE = 0.2 για συντομότερα ερωτήματα και ALPHA_LONGQ = 0.35 για μεγαλύτερα, δίνοντας γενικά μεγαλύτερη βαρύτητα στη λεξική συνιστώσα.

Μια περαιτέρω ανάλυση στην παράμετρο α θα περιελάμβανε τη συστηματική μεταβολή της τιμής της και την επανεκτέλεση της αξιολόγησης για κάθε τιμή του α και για κάθε ένα από τα τρία γλωσσικά μοντέλα. Τα αποτελέσματα αυτής της ανάλυσης θα μπορούσαν να απεικονιστούν γραφικά, δείχνοντας πώς μεταβάλλεται η απόδοση του συστήματος καθώς αλλάζει η σχετική βαρύτητα της σημασιολογικής έναντι της λεξικής αναζήτησης.

- Θα μπορούσε να αποκαλύψει τη βέλτιστη τιμή του α για το συγκεκριμένο dataset και μοντέλο και τύπο ερωτήσεων, δυνητικά οδηγώντας σε βελτιωμένη απόδοση σε σχέση με τις αρχικά επιλεγμένες τιμές.
- Θα έδειχνε πόσο ευαίσθητη είναι η απόδοση του συστήματος στις μεταβολές αυτού του συντελεστή, παρέχοντας πληροφορίες για την βελτιστοποίηση της υβριδικής προσέγγισης.

Παρόλα αυτά επιλέχθηκε να κρατηθεί και στα τρία μοντέλα, το ίδιο μοτίβο έτσι ώστε να επιτευχθεί μία πιο αντικειμενική προσέγγιση και στα τρία LLM.

8.7 Αποτελέσματα – πλεονεκτήματα & μειονεκτήματα κάθε μοντέλου

Η συνδυαστική ανάλυση των ποσοτικών μετρικών και των ποιοτικών παρατηρήσεων μας επιτρέπει την εξαγωγή συμπερασμάτων σχετικά με την απόδοση, τα πλεονεκτήματα και τα μειονεκτήματα κάθε ενός από τα τρία εξεταζόμενα ελληνικά LLMs.

- **GreekBERT (nlpaueb/bert-base-greek-uncased-v1):**
    1. **Πλεονεκτήματα:** Πέτυχε την υψηλότερη μέση τιμή nDCG@5 (0.946), υποδεικνύοντας την καλύτερη συνολική ποιότητα κατάταξης των σχετικών εγγράφων στις κορυφαίες θέσεις. Ως μικρότερο μοντέλο, έχει σημαντικά χαμηλότερες απαιτήσεις σε υπολογιστικούς πόρους (VRAM, χρόνος inference για το query embedding) σε σύγκριση με τα Meltemi και KriKri, καθιστώντας το πιο εύκολα αξιοποιήσιμο σε περιβάλλοντα με περιορισμένους πόρους.

    2. **Μειονεκτήματα:** Είχε το χαμηλότερο MCG@5 (9.57) και MRR (0.624). Αυτό υποδηλώνει ότι, ενώ η κατάταξή του ήταν καλή όταν έβρισκε σχετικά έγγραφα, μπορεί να δυσκολευόταν περισσότερο να ανακτήσει συνολικά τον ίδιο αριθμό σχετικών εγγράφων ή να φέρει το πρώτο άκρως σχετικό έγγραφο τόσο ψηλά όσο τα μεγαλύτερα μοντέλα, ειδικά για ερωτήματα που απαιτούσαν βαθύτερη σημασιολογική κατανόηση.
- **Meltemi (ilsp/Meltemi-7B-Instruct-v1.5):**
    1. **Πλεονεκτήματα:** Επέδειξε την υψηλότερη τιμή MRR (0.772), δείχνοντας ότι ήταν το πιο αποτελεσματικό στο να επιστρέφει τουλάχιστον ένα άκρως σχετικό αποτέλεσμα σε πολύ υψηλή θέση κατάταξης. Είχε επίσης καλή απόδοση στο MCG@5 (9.70), ελαφρώς καλύτερη από το GreekBERT. Η φύση του ως instruct-tuned μοντέλο μπορεί να συμβάλλει στην καλύτερη κατανόηση της πρόθεσης του ερωτήματος για την παραγωγή ποιοτικών embeddings.

    2. **Μειονεκτήματα**: Είχε ελαφρώς χαμηλότερο Mean nDCG@5 (0.939) από το GreekBERT, υποδηλώνοντας ότι η συνολική κατάταξη των 5 κορυφαίων αποτελεσμάτων μπορεί να μην ήταν πάντα τόσο βέλτιστη. Ως μοντέλο 7 δισεκατομμυρίων παραμέτρων, έχει σημαντικά υψηλότερες απαιτήσεις σε VRAM και χρόνο επεξεργασίας.
- **KriKri (ilsp/Llama-Krikri-8B-Base):**
    1. **Πλεονεκτήματα:** Πέτυχε το υψηλότερο MCG@5 (9.90), υποδεικνύοντας ότι συνολικά ανέκτησε τα πιο σχετικά, με βάση το άθροισμα των βαθμών συνάφειας έγγραφα στις 5 κορυφαίες θέσεις. Το MRR του (0.717) ήταν επίσης ανταγωνιστικό, καλύτερο από του GreekBERT.

    2. **Μειονεκτήματα:** Είχε το χαμηλότερο Mean nDCG@5 (0.933) από τα τρία μοντέλα, πράγμα που σημαίνει ότι ενώ μπορεί να έφερνε πολλά σχετικά έγγραφα, η σειρά κατάταξής τους δεν ήταν πάντα η ιδανική. Ως το μεγαλύτερο μοντέλο (8 δισεκατομμύρια παράμετροι), είναι το πιο απαιτητικό σε υπολογιστικούς πόρους.

Συνολικά, η υβριδική προσέγγιση φαίνεται να αποδίδει καλά, με όλες τις τιμές Mean nDCG@5 να είναι πάνω από 0.93, επιδεικνύοντας υψηλή ποιότητα κατάταξης. Η επιλογή του "καλύτερου" μοντέλου εξαρτάται από κάποιες προτεραιότητες σαν αυτές:

- Αν η βέλτιστη κατάταξη όλων των κορυφαίων αποτελεσμάτων είναι πρωταρχικής σημασίας, το GreekBERT φαίνεται να υπερτερεί ελαφρώς λόγω υψηλότερου nDCG@5.

- Αν η γρήγορη εύρεση του πρώτου πιο σχετικού εγγράφου είναι ο στόχος, το Meltemi έχει το πλεονέκτημα με υψηλότερο MRR, συγκριτικά πάντα με τα άλλα.
- Αν η συνολική ανάκτηση όσο το δυνατόν περισσότερης σχετικής πληροφορίας στις πρώτες θέσεις είναι το ζητούμενο, το KriKri δείχνει να είναι ελαφρώς καλύτερο κάτι που φάνηκε από το υψηλότερο MCG@5.

Φυσικά, οι υπολογιστικοί πόροι και ο επιθυμητός χρόνος απόκρισης παίζουν επίσης καθοριστικό ρόλο στην τελική επιλογή για μια παραγωγική εφαρμογή στην πραγματικότητα.

8.8 Επίλογος

Η πειραματική αξιολόγηση που διεξήχθη στο παρόν κεφάλαιο παρέχει πολύτιμες ενδείξεις για την απόδοση των τριών επιλεγμένων ελληνικών LLM στο πλαίσιο ενός συστήματος υβριδικής ανάκτησης πληροφοριών για δημόσια έγγραφα. Μέσω ενός συνόλου 30 ερωτήσεων και της εφαρμογής καθιερωμένων μετρικών αξιολόγησης, όπως το Mean Cumulative Gain, το Normalized Discounted Cumulative Gain και το Mean Reciprocal Rank, κατέστη δυνατή η ποσοτική σύγκριση της αποτελεσματικότητάς τους.

Τα αποτελέσματα έδειξαν ότι και τα τρία μοντέλα μπορούν να επιτύχουν υψηλή απόδοση, με το καθένα να παρουσιάζει πλεονεκτήματα σε διαφορετικές πτυχές της ανάκτησης. Το KriKri επέδειξε την καλύτερη συνολική ανάκτηση σχετικότητας, το Meltemi την καλύτερη ικανότητα εντοπισμού του πρώτου άκρως σχετικού εγγράφου, ενώ το GreekBERT την καλύτερη συνολική ποιότητα κατάταξης των σχετικών αποτελεσμάτων , όντας παράλληλα το πιο αποδοτικό από άποψη υπολογιστικών πόρων.

Κεφάλαιο 9ο: Συμπεράσματα & Προτάσεις

9.1 Εισαγωγή

Η παρούσα διπλωματική εργασία επικεντρώθηκε στον σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός chatbot βασισμένου σε Μεγάλα Γλωσσικά Μοντέλα (LLMs) και εν μέρη την αρχιτεκτονική Ανάκτησης Επαυξημένης Παραγωγής (RAG), με στόχο τη διευκόλυνση της αναζήτησης και ανάκτησης πληροφοριών από ελληνικά δημόσια έγγραφα, συγκεκριμένα από προκηρύξεις εντεταλμένων διδασκόντων πανεπιστημίων. Μέσα από μια συστηματική διαδικασία συλλογής δεδομένων, προεπεξεργασίας, δημιουργίας διανυσματικών αναπαραστάσεων, υλοποίησης ενός υβριδικού μηχανισμού αναζήτησης και ανάπτυξης μιας λειτουργικής διεπαφής χρήστη, διερευνήθηκε η απόδοση τριών διαφορετικών ελληνικών LLMs στο συγκεκριμένο πλαίσιο.

Το παρόν κεφάλαιο αποσκοπεί στη σύνοψη των κυριότερων θεμάτων της εργασίας, στην αναγνώριση των περιορισμών που προέκυψαν κατά την έρευνα και την υλοποίηση, καθώς και στην παράθεση συγκεκριμένων προτάσεων για μελλοντικές βελτιώσεις και επεκτάσεις του αναπτυχθέντος συστήματος.

9.2 Περιορισμοί

Παρά τις προσπάθειες για μια ολοκληρωμένη προσέγγιση, η παρούσα διπλωματική εργασία ενέχει ορισμένους περιορισμούς, οι οποίοι πρέπει να αναγνωριστούν:

- **Πεδίο Εφαρμογής και Μέγεθος Dataset:** Η έρευνα εστιάστηκε αποκλειστικά σε έναν τύπο δημοσίου εγγράφου (προκηρύξεις εντεταλμένων διδασκόντων) και σε ένα σχετικά περιορισμένο σύνολο 114 αρχείων PDF. Αυτό ενδέχεται να περιορίζει τη γενικευσιμότητα των συμπερασμάτων σχετικά με την απόδοση των μοντέλων σε άλλους τύπους εγγράφων ή σε πολύ μεγαλύτερα σύνολα δεδομένων.
- **Ποικιλομορφία Διάταξης των PDF**: Η ευρετική μέθοδος που χρησιμοποιήθηκε για την αφαίρεση επικεφαλίδων/υποσέλιδων και η γενικότερη εξαγωγή κειμένου από PDF με ποικίλες και ενίοτε μη τυποποιημένες διατάξεις ενδέχεται να μην ήταν πάντοτε απόλυτα ακριβής. Αυτό μπορεί να οδήγησε στην εισαγωγή "θορύβου" ή στην απώλεια τμημάτων κειμένου κατά την προεπεξεργασία, επηρεάζοντας την ποιότητα του τελικού dataset και, κατ' επέκταση, την απόδοση του συστήματος ανάκτησης. Η διαχείριση πινάκων και άλλων σύνθετων δομών ήταν επίσης στοιχειώδης και πολυσύνθετη στην αποδόμηση τους καθώς αποτελούνταν από διαφορετικά μήκη και πλάτη.
- **Υπολογιστικοί Πόροι (VRAM):** Οι σημαντικές απαιτήσεις σε VRAM, ειδικά για τα μεγαλύτερα μοντέλα όπως το Meltemi και το KriKri , αποτέλεσαν έναν πρακτικό περιορισμό. Η ανάγκη προσαρμογής των μεγεθών batch και η χρήση τεχνικών όπως torch.float16 ή device_map="auto" ήταν απαραίτητες για τη λειτουργία τους σε περιβάλλοντα όπως τα Hugging Face Spaces, αλλά ενδέχεται να μην αντικατοπτρίζουν την απόδοση σε συστήματα με απεριόριστους πόρους.
- **Περιορισμοί Αξιολόγησης:** Η αξιολόγηση βασίστηκε σε ένα σύνολο 30 ερωτήσεων και χειροκίνητη κρίση συνάφειας. Αν και σχεδιάστηκε για να είναι αντιπροσωπευτικό, ένα

μεγαλύτερο σύνολο ερωτήσεων και η αξιολόγηση από πολλαπλούς κριτές θα μπορούσαν να προσφέρουν πιο στατιστικά ισχυρά αποτελέσματα.

9.3 Προτάσεις

Με βάση τα ευρήματα και τους περιορισμούς της παρούσας εργασίας, προτείνονται οι ακόλουθες κατευθύνσεις για μελλοντική έρευνα και βελτίωση του συστήματος:

- **Fine-tuning των LLMs**: Το fine tuning των LLMs σε ένα ευρύτερο και πιο εξειδικευμένο σώμα κειμένων από τον χώρο της ελληνικής δημόσιας διοίκησης ή συγκεκριμένα από προκηρύξεις, θα μπορούσε να βελτιώσει σημαντικά την ποιότητα των παραγόμενων embeddings και την κατανόηση της εξειδικευμένης ορολογίας, οδηγώντας σε ακριβέστερη σημασιολογική αναζήτηση.
- **Ενσωμάτωση Μηχανισμών Επανακατάταξης (Re-rankers):** Μετά την αρχική ανάκτηση υποψήφιων τμημάτων από τον υβριδικό μηχανισμό, θα μπορούσε να εφαρμοστεί ένα δεύτερο στάδιο επανακατάταξης με χρήση πιο ισχυρών, αλλά υπολογιστικά πιο απαιτητικών, μοντέλων (cross-encoders). Αυτό θα μπορούσε να βελτιώσει την ακρίβεια των κορυφαίων αποτελεσμάτων.
- **Βελτιώσεις στη Διεπαφή Χρήστη (UI):** Προηγμένες Επιλογές Φιλτραρίσματος: Προσθήκη δυνατοτήτων φιλτραρίσματος των αποτελεσμάτων βάσει μεταδεδομένων (π.χ., φορέας, ημερομηνία ανάρτησης, τύπος πράξης).
- **FeedBack:** Ενσωμάτωση ενός συστήματος όπου οι χρήστες θα μπορούσαν να αξιολογούν τη συνάφεια των αποτελεσμάτων, παρέχοντας πολύτιμα δεδομένα για περαιτέρω βελτίωση του αλγορίθμου.
- **Παραγωγή Περίληψης**: Εξέταση της δυνατότητας χρήσης του LLM για την παραγωγή μιας σύντομης περίληψης των ανακτηθέντων σχετικών τμημάτων, αντί της απλής επιστροφής των chunks.
- **Εξαντλητική Βελτιστοποίηση** : Διεξαγωγή εκτεταμένης ανάλυσης ευαισθησίας για τον συντελεστή α της υβριδικής αναζήτησης, καθώς και για άλλες παραμέτρους όπως το CHUNK_SIZE, CHUNK_OVERLAP και ο αριθμός των ανακτηθέντων εγγράφων k, για κάθε μοντέλο ξεχωριστά.

9.4 Επέκταση σε άλλες κατηγορίες Δημοσίου

Η μεθοδολογία και η αρχιτεκτονική που αναπτύχθηκαν στην παρούσα εργασία για τις προκηρύξεις εντεταλμένων διδασκόντων έχουν τη δυνατότητα να επεκταθούν και να προσαρμοστούν για την ανάκτηση πληροφοριών και από άλλες σημαντικές κατηγορίες εγγράφων του ελληνικού δημόσιου τομέα.

- **Φύλλο Εφημερίδας της Κυβερνήσεως** : Τα ΦΕΚ περιέχουν νόμους, προεδρικά διατάγματα, υπουργικές αποφάσεις και άλλες κανονιστικές πράξεις. Η δημιουργία ενός συστήματος RAG για την αναζήτηση σε ΦΕΚ θα ήταν εξαιρετικά χρήσιμη για νομικούς, δημοσιογράφους, ερευνητές και πολίτες. Οι προκλήσεις εδώ περιλαμβάνουν την πολύ τυποποιημένη και συχνά αρχαϊζουσα γλώσσα, καθώς και την ανάγκη για ακριβή αναγνώριση άρθρων, παραγράφων και τροποποιήσεων.

- **Κεντρικό Ηλεκτρονικό Μητρώο Δημοσίων Συμβάσεων:** Η πλατφόρμα αυτή περιέχει πληθώρα εγγράφων σχετικά με δημόσιους διαγωνισμούς και συμβάσεις. Ένα σύστημα RAG θα μπορούσε να διευκολύνει τις επιχειρήσεις στην εύρεση σχετικών προκηρύξεων και την παρακολούθηση των διαδικασιών. Η δομή των εγγράφων και η εξειδικευμένη ορολογία αποτελούν και εδώ σημεία που απαιτούν προσοχή.
- **Αποφάσεις άλλων Δημοσίων Φορέων:** Πλήθος άλλων αποφάσεων από δήμους, περιφέρειες, υπουργεία και ανεξάρτητες αρχές αναρτώνται στη ΔΙΑΥΓΕΙΑ. Η επέκταση του συστήματος σε αυτές τις κατηγορίες θα απαιτούσε την προσαρμογή της προεπεξεργασίας για την αντιμετώπιση της ποικιλομορφίας των μορφών και του περιεχομένου.

9.5 Επίλογος

Η παρούσα διπλωματική εργασία κατέδειξε τη δυνατότητα αξιοποίησης των Μεγάλων Γλωσσικών Μοντέλων και της αρχιτεκτονικής RAG για την ανάπτυξη ενός αποτελεσματικού συστήματος ανάκτησης πληροφοριών από εξειδικευμένα ελληνικά δημόσια έγγραφα. Μέσω της συγκριτικής αξιολόγησης των GreekBERT, Meltemi και KriKri, αναδείχθηκαν οι επιδόσεις και οι ιδιαιτερότητες κάθε μοντέλου, παρέχοντας χρήσιμα συμπεράσματα για την επιλογή του καταλληλότερου εργαλείου ανάλογα με τις εκάστοτε απαιτήσεις. Η υλοποίηση ενός υβριδικού μηχανισμού αναζήτησης απέδειξε την αξία του συνδυασμού λεξικών και σημασιολογικών προσεγγίσεων.

Παρά τους περιορισμούς που εντοπίστηκαν, κυρίως όσον αφορά την ποικιλομορφία των PDF και τις υπολογιστικές απαιτήσεις, η εργασία συνεισφέρει στην έρευνα για την Επεξεργασία Φυσικής Γλώσσας στην ελληνική γλώσσα και προσφέρει μια πρακτική λύση σε ένα υπαρκτό πρόβλημα πρόσβασης στην πληροφορία. Οι προτάσεις για μελλοντικές βελτιώσεις, ανοίγουν τον δρόμο για περαιτέρω έρευνα και ανάπτυξη. Η συνεχής εξέλιξη των LLMs και των συναφών τεχνολογιών υπόσχεται να μεταμορφώσει τον τρόπο με τον οποίο αλληλεπιδρούμε με την πληροφορία, και η εφαρμογή τους στον ελληνικό δημόσιο τομέα μπορεί να συμβάλλει καθοριστικά στην ενίσχυση της διαφάνειας, της προσβασιμότητας και της αποδοτικότητας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ελίζα Τριανταφύλλου, “Στα άδυτα της Διαύγειας, των δημοσίων συμβάσεων και του εμπορικού μητρώου” , 6 Σεπτεμβρίου 2017. [Online]. Available: https://opengov.ellak.gr/2017/09/06/sta-adita-tis-diavgias-ton-dimosion-simvaseon-ke-tou-emporikou-mitroou/

[2] Ioannis Pavlopoulos, greek-nlp/benchmark: “The GenA of Greek NLP - GitHub”, November 27, 2024. [Online]. Available: https://github.com/greek-nlp/benchmark

[3] V. Saketos, D.-A. Pantazi, and M. Koubarakis, “The Large Language Model GreekLegalRoBERTa,” in Proc. 13th Conf. on Artificial Intelligence (SETN 2024), Piraeus, Greece, Sep. 11–13, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3688671.3688770

[4] John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis and Ion Androutsopoulos, “A Greek version of BERT pre-trained language model.”, nlpaueb/bert-base-greek-uncased-v1, Published on Aug 27, 2020. [Online]. Available: https://huggingface.co/nlpaueb/bert-base-greek-uncased-v1

[5] John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, Ion Androutsopoulos, “GREEK-BERT: The Greeks visiting Sesame Street”, Department of Informatics, Athens University of Economics and Business, 3 Sep 2020, [Online]. Available: https://arxiv.org/abs/2008.12014

[6] I. Veneti, I. Varlamis, and G. Th. Papadopoulos, “Classification of Greek News Articles by Genre Using Open Large Language Models”, September 11–13, 2024, Piraeus, Greece. [Online]. Available: https://www.researchgate.net/publication/389390004_Classification_of_Greek_News_Articles_by_Genre_Using_Open_Large_Language_Models

[7] Pranaydeep Singh, "A Pilot Study for BERT Language Modelling and Morphological Analysis for Ancient and Medieval Greek", Ancient-Greek-BERT, 24 Sep 2021, [Online].Available: https://github.com/pranaydeeps/Ancient-Greek-BERT

[8] “Meltemi: Ένα μεγάλο γλωσσικό μοντέλο ανοιχτού λογισμικού για τα Ελληνικά από το ΙΕΛ Αθηνά”, 02/04/2024, [Online].Available: https://digi.gov.gr/meltemi-ena-megalo-glossiko-montelo-gia-ta-ellinika-apo-to-iel-athina/

[9] Dimitris Roussis, Leon Voukoutis, Georgios Paraskevopoulos, Sokratis Sofianopoulos, Prokopis Prokopidis, Vassilis Papavasileiou, Athanasios Katsamanis, Stelios Piperidis, Vassilis Katsouros, ilsp/Llama-Krikri-8B-Base, “Llama-Krikri-8B-Base: A large foundation Language Model for the Greek language”, Institute for Language and Speech Processing, [Online].Available: https://huggingface.co/ilsp/Llama-Krikri-8B-Base

[10] Dimitris Roussis, Leon Voukoutis, Georgios Paraskevopoulos, Sokratis Sofianopoulos, Prokopis Prokopidis, Vassilis Papavasileiou, Athanasios Katsamanis, Stelios Piperidis, Vassilis Katsouros, “Krikri: Advancing Open Large Language Models for Greek”, Institute for Speech and Language Processing, Athena Research Center Artemidos 6 & Epidavrou, Athens, Greece, 19 May 2025, [Online].Available: https://arxiv.org/abs/2505.13772

[11] “ Meltemi 7B V1”, [Online].Available: https://dataloop.ai/library/model/ilsp_meltemi-7b-v1/

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space,", arXiv:1301.3781, 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", arXiv:1908.10084, 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[14] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv, arXiv:2312.10997, 2023. [Online]. Available: https://arxiv.org/abs/2312.10997

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv, arXiv:2005.11401, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

[16] Tiago Carneiro, Raul V. Medeiros da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C. de Albuquerque, Pedro P. Rebouças Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," DOI 10.1109/ACCESS.2018.2874767, 2018. [Online].Available:https://www.researchgate.net/publication/328158184_Performance_Analysis_of_Google_Colaboratory_as_a_Tool_for_Accelerating_Deep_Learning_Applications

[17] "The PyPDF2 project is going back to its roots. PyPDF2==3.0.X will be the last version of PyPDF2.", Dec 31, 2022. [Online]. Available: https://pypi.org/project/PyPDF2/

[18] "Protect_dates", Anashel, [Online]. Available: https://github.com/Anashel-RPG/echoai/blob/main/utils.py

[19] Gene Diaz, " Stopwords ISO", Philippines, [Online].Available: https://github.com/stopwords-iso/stopwords-el/blob/master/stopwords-el.txt

[20] Georgios Ntais, "Development of a Stemmer for the Greek Language", February 2006, Stockholm University. [Online]. Available: https://people.dsv.su.se/~hercules/papers/Ntais_greek_stemmer_thesis_final.pdf

[21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander M. Rush, "Transformers: State-of-the-Art Natural Language Processing," arXiv, Brooklyn, USA , 14 Jul 2020. [Online].Available: https://arxiv.org/abs/1910.03771

[22] Byungsoo Ko, Han-Gyu Kim, Byeongho Heo, Sangdoo Yun, Sanghyuk Chun, Geonmo Gu, Wonjae Kim, "Group Generalized Mean Pooling for Vision Transformer," arXiv, 8 Dec 2022. [Online]. Available: https://arxiv.org/abs/2212.04114

[23] Yu. A. Malkov, D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," arXiv,14 Aug 2018. [Online].Available: https://arxiv.org/abs/1603.09320

[24] Vyacheslav Efimov, "Similarity Search, Part 4: Hierarchical Navigable Small World (HNSW)",Jun 16, 2023. [Online].Available: https://medium.com/data-science/similarity-search-part-4-hierarchical-navigable-small-world-hnsw-2aad4fe87d37

[25] John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, “Charagram: Embedding Words and Sentences via Character n-grams”, Chicago, USA, 10 Jul 2016, [Online].Available: https://arxiv.org/abs/1607.02789

[26] Zhuolin Jiang, Manaj Srivastava, Sanjay Krishna, David Akodes, Richard Schwartz, “Combining Word Embeddings and N-grams for Unsupervised Document Summarization”, Cambridge, MA, 25 Apr 2020, [Online].Available: https://arxiv.org/abs/2004.14119

[27] Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi, Daniela Petrelli, “Hybrid Search: Effectively Combining Keywords and Semantic Searches”, Sheffield United Kingdom, [Online].Available: https://link.springer.com/chapter/10.1007/978-3-540-68234-9_41

[28] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, James Zou, “Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild”, 6 Jun 2019, [Online].Available: https://arxiv.org/abs/1906.02569

[29] “Model memory estimator”, [Online].Available: https://huggingface.co/docs/accelerate/usage_guides/model_size_estimator

[30] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, Jimmy Lin, “A Large-Scale Study of Relevance Assessments with Large Language Models: An Initial Look”, USA and Canada, 13 Nov 2024, [Online].Available: https://arxiv.org/abs/2411.08275

[31] Hao Kang, “ResearchArena: Benchmarking LLMs’ Ability to Collect and Organize Information as Research Agents”, Carnegie Mellon University Pittsburgh, PA, 15213, 13 Jun 2024, [Online].Available: https://arxiv.org/html/2406.10291v1

[32] Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, “STaRK: Benchmarking LLM Retrieval on Textual and Relational Knowledge Bases”, Department of Computer Science, Stanford University, 20 May 2024, [Online].Available: https://arxiv.org/html/2404.13207v2

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α: Υβριδικός αλγόριθμος:

```
# ---------------------- HYBRID SEARCH (Κύρια Λογική) ---
def hybrid_search_gradio(query, k=5):
    if not setup_successful or not ids or not col or not model or not tok:
        return "Σφάλμα: Η εφαρμογή δεν αρχικοποιήθηκε σωστά. Τα δεδομένα ή το
μοντέλο δεν φορτώθηκαν. Ελέγξτε τα logs εκκίνησης."
    if not query.strip():
        return "Παρακαλώ εισάγετε μια ερώτηση."

    q_pre = preprocess(query)
    words = q_pre.split()
    alpha = ALPHA_LONGQ if len(words) > 30 else ALPHA_BASE
    exact_ids_set = {ids[i] for i, t in enumerate(pre_chunks) if q_pre in t}
    q_emb_np = cls_embed([q_pre], tok, model)
    q_emb_list = q_emb_np.tolist()

    try:
        sem_results = col.query(query_embeddings=q_emb_list,
n_results=min(150, len(ids)), include=["distances"])
    except Exception as e:
        # Εκτύπωση του σφάλματος στα logs του server για διάγνωση
        print(f"ERROR during ChromaDB query in hybrid_search_gradio:
{type(e).__name__}: {e}")
        return "Σφάλμα κατά την σημασιολογική αναζήτηση. Επικοινωνήστε με τον
διαχειριστή."

    sem_sims = {doc_id: 1 - dist for doc_id, dist in
zip(sem_results["ids"][0], sem_results["distances"][0])}
    q_char_sparse = char_vec.transform([q_pre])
    q_char_normalized = sk_normalize(q_char_sparse)
    char_sim_scores = (q_char_normalized @ X_char.T).toarray().flatten()
    q_word_sparse = word_vec.transform([q_pre])
    q_word_normalized = sk_normalize(q_word_sparse)
    word_sim_scores = (q_word_normalized @ X_word.T).toarray().flatten()
    lex_sims = {}
```

```
    for idx, (c_score, w_score) in enumerate(zip(char_sim_scores,
word_sim_scores)):
        if c_score > 0 or w_score > 0:
            if idx < len(ids): lex_sims[ids[idx]] = 0.85 * c_score + 0.15 *
w_score
            else: print(f"Warning (hybrid_search): Lexical score index {idx}
out of bounds for ids list (len: {len(ids)}).")

    all_chunk_ids_set = set(sem_sims.keys()) | set(lex_sims.keys()) |
exact_ids_set
    scored = []
    for chunk_id_key in all_chunk_ids_set:
        s = alpha * sem_sims.get(chunk_id_key, 0.0) + (1 - alpha) *
lex_sims.get(chunk_id_key, 0.0)
        if chunk_id_key in exact_ids_set: s = 1.0
        scored.append((chunk_id_key, s))
    scored.sort(key=lambda x: x[1], reverse=True)
    hits_output = []
    seen_doc_main_ids = set()
    for chunk_id_val, score_val in scored:
        try: idx_in_lists = ids.index(chunk_id_val)
        except ValueError: print(f"Warning (hybrid_search): chunk_id
'{chunk_id_val}' not found in loaded ids. Skipping."); continue
        doc_meta = metas[idx_in_lists]
        doc_main_id = doc_meta['id']
        if doc_main_id in seen_doc_main_ids: continue
        original_url_from_meta = doc_meta.get('url', '#')
        pdf_gcs_url = "#"
        pdf_filename_display = "N/A"
        if original_url_from_meta and original_url_from_meta != '#':
            pdf_filename_extracted = os.path.basename(original_url_from_meta)
            if pdf_filename_extracted and
pdf_filename_extracted.lower().endswith(".pdf"):
                pdf_gcs_url =
f"{GCS_PUBLIC_URL_PREFIX}{pdf_filename_extracted}"
                pdf_filename_display = pdf_filename_extracted
            elif pdf_filename_extracted: pdf_filename_display = "Source is not
a PDF"

        hits_output.append({
            "score": score_val, "title": doc_meta.get('title', 'N/A'),
            "snippet": raw_chunks[idx_in_lists][:500] + " ...",
            "original_url_meta": original_url_from_meta, "pdf_serve_url":
pdf_gcs_url,
            "pdf_filename_display": pdf_filename_display
        })
```

```
        seen_doc_main_ids.add(doc_main_id)
        if len(hits_output) >= k: break
    if not hits_output: return "Δεν βρέθηκαν σχετικά αποτελέσματα."
    output_md = f"Βρέθηκαν **{len(hits_output)}** σχετικά αποτελέσματα:\n\n"
    for hit in hits_output:
        output_md += f"### {hit['title']} (Score: {hit['score']:.3f})\n"
        snippet_wrapped = textwrap.fill(hit['snippet'].replace("\n", " "),
width=100)
        output_md += f"**Απόσπασμα:** {snippet_wrapped}\n"
        if hit['pdf_serve_url'] and hit['pdf_serve_url'] != '#':
            output_md += f"**Πηγή (PDF):** <a href='{hit['pdf_serve_url']}'
target='_blank'>{hit['pdf_filename_display']}</a>\n"
        elif hit['original_url_meta'] and hit['original_url_meta'] != '#':
            output_md += f"**Πηγή (αρχικό από metadata):**
[{hit['original_url_meta']}]({hit['original_url_meta']})\n"
        output_md += "---\n"
    return output_md
```