ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Εκπαιδευτικού Διαλογικού Βοηθού με Χρήση Μεγάλων Γλωσσικών Μοντέλων και Αρχιτεκτονικής Retrieval-Augmented Generation (RAG)»

Των φοιτητών:

1) Καπαρού Αντωνίου 07/2023
2) Μπάρης Χαρίλαος 13/2023

Επιβλέπων
Κωνσταντίνος Διαμαντάρας
Καθηγητής

Ημερομηνία 08/02/2025

Τίτλος Δ.Ε. «Ανάπτυξη Εκπαιδευτικού Διαλογικού Βοηθού με Χρήση Μεγάλων Γλωσσικών Μοντέλων και Αρχιτεκτονικής Retrieval-Augmented Generation (RAG)»

Κωδικός Δ.Ε. 24250

Ονοματεπώνυμο φοιτητών: Καπαρός Αντώνιος, Μπάρης Χαρίλαος

Ονοματεπώνυμο εισηγητή: Κωνσταντίνος Διαμαντάρας

Ημερομηνία ανάληψης Δ.Ε. 08/10/2024

Ημερομηνία περάτωσης Δ.Ε. 08/02/2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω 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

Πρόλογος

Η ανάπτυξη ενός εκπαιδευτικού chatbot που συνδυάζει τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) και την αρχιτεκτονική Retrieval – Augmented Generation (RAG) προσφέρει σημαντικά οφέλη στη μαθησιακή διαδικασία. Η συνεργασία αυτών των τεχνολογιών επιτρέπει στο chatbot να κατανοεί και να παράγει φυσική γλώσσα με ακρίβεια, ενώ παράλληλα ενισχύει την ποιότητα των απαντήσεων μέσω αναζήτησης αξιόπιστων πληροφοριών. Αυτό συμβάλλει στη μείωση λαθών και παραπληροφόρησης, καθιστώντας τη διάδραση πιο φυσική και αποτελεσματική.

Στη σύγχρονη εκπαιδευτική πραγματικότητα, οι απαιτήσεις της ύλης των μαθημάτων είναι πολύ μεγάλες. Ακόμα, οι μαθητές έχουν αρκετές εξωδιδακτικές δραστηριότητες, με αποτέλεσμα να περιορίζεται ο χρόνος μελέτης τους, ενώ και οι εκπαιδευτικοί αντιμετωπίζουν αυξανόμενο φόρτο εργασίας με την αξιολόγηση των μαθητών, αλλά και την προετοιμασία για τα μαθήματα. Ένα chatbot με LLM και RAG μπορεί να λειτουργήσει ως υποστηρικτικό εργαλείο, προσφέροντας εξατομικευμένη και συστηματική μάθηση, βοηθώντας στην κατανόηση εννοιών μέσω διαλογικής αλληλεπίδρασης. Επιπλέον, παρέχει συνεχή πρόσβαση σε αξιόπιστο εκπαιδευτικό υλικό και συμβάλλει στην αξιολόγηση των μαθητών.

Η δημιουργία αυτού του chatbot δεν ωφελεί μόνο μαθητές και εκπαιδευτικούς, αλλά αποτελεί και μια ευκαιρία για εμάς να εμβαθύνουμε στην τεχνολογία των LLMs και RAG. Με την ενσωμάτωση αυτών των τεχνικών, το chatbot παραμένει ενημερωμένο και χρήσιμο σε ένα δυναμικό εκπαιδευτικό περιβάλλον.

Περίληψη

Η χρήση των Διαλογικών Βοηθών (Chatbots), που υποστηρίζονται με τεχνητή νοημοσύνη, αναδεικνύεται ως ένας από τους πιο σημαντικούς τομείς σε επίπεδο έρευνας και ανάπτυξης τεχνολογικών εφαρμογών τόσο στους κλάδους των εταιρειών, της επιχειρηματικότητας και της βιομηχανίας, όσο και σε επίπεδο οργανισμών, ιδρυμάτων και κρατικών υπηρεσιών, στοχεύοντας στην ενίσχυση της παραγωγικότητας των εργαζομένων. Τα Large Language Models (LLMs) ειδικεύονται στην επεξεργασία και τη δημιουργία φυσικής γλώσσας. Διαπρέπουν σε εργασίες όπως η κατανόηση γλώσσας, η δημιουργία κειμένου, η μετάφραση και η περίληψη. Η αρχιτεκτονική Retrieval – Augmented Generation (RAG) ενσωματώνει στα LLMs εξωτερικές πηγές γνώσης, ενισχύοντάς τα με χρήσιμες πληροφορίες με αποτέλεσμα να εξάγονται απαντήσεις με μεγαλύτερη ακρίβεια. Η αρχιτεκτονική RAG και τα LLMs χρησιμεύουν ως βασικά τεχνολογικά στοιχεία στη δημιουργία Chatbots που βασίζονται σε "Generative-AI". Ωστόσο, η δημιουργία επιτυχημένων διαλογικών βοηθών δεν είναι εύκολη. Απαιτείται σχολαστικός σχεδιασμός της αρχιτεκτονικής RAG. Αυτό περιλαμβάνει λεπτομερή ρύθμιση των LLMs και των σημασιολογικών όρων που ενσωματώνονται στο μοντέλο, εξαγωγή σχετικών εγγράφων από διανυσματικές βάσεις δεδομένων (Vector Databases), αναδιατύπωση ερωτημάτων, ανακατάταξη αποτελεσμάτων, σχεδιασμό αποτελεσματικών προτροπών, ρύθμιση πρόσβασης στα έγγραφα, παροχή συνοπτικών απαντήσεων συμπεριλαμβανομένων σχετικών αναφορών, διαφύλαξη προσωπικών πληροφοριών και δημιουργία πρακτόρων για το συντονισμό όλων αυτών των δραστηριοτήτων. Σε αυτό το έγγραφο, παρουσιάζουμε τον σχεδιασμό και την ανάπτυξη ενός Εκπαιδευτικού Διαλογικού Βοηθού με χρήση Μεγάλων Γλωσσικών Μοντέλων και αρχιτεκτονικής Retrieval – Augmented Generation, ο οποίος μπορεί να αξιοποιηθεί σε σχολικά βιβλία της Πρωτοβάθμιας και Δευτεροβάθμιας Εκπαίδευσης.

« Development of an Educational Conversational Assistant Using Large Language Models and Retrieval-Augmented Generation (RAG) Architecture»

Antonios Kaparos

Harilaos Baris

Abstract

The use of Chatbots, supported by artificial intelligence, is emerging as one of the most important areas in terms of research and development of technological applications in both the corporate, entrepreneurship and industrial sectors, as well as at the level of organizations, institutions and government services, aiming to enhance employee productivity. Large Language Models (LLMs) specialize in the processing and creation of natural language. They excel in tasks such as language understanding, text generation, translation and summarization. The Retrieval – Augmented Generation (RAG) architecture integrates external sources of knowledge into LLMs, enhancing them with useful information, resulting in more accurate answers. The RAG architecture and LLMs serve as key technological elements in the creation of Chatbots based on "Generative-AI". However, creating successful chatbots is not easy. Meticulous design of the RAG architecture is required. This includes fine-tuning the LLMs and semantic terms embedded in the model, extracting relevant documents from vector databases, reformulating queries, reordering results, designing effective prompts, configuring access to documents, providing summary answers including relevant references, preserving personal information, and creating agents to coordinate all these activities. In this paper, we present the design and development of an Educational Dialogue Assistant using Large Language Models and Retrieval – Augmented Generation architecture, which can be utilized in Primary and Secondary Education textbooks.

Ευχαριστίες

Ευχαριστούμε θερμά τις οικογένειές μας για όλη την ηθική στήριξη κατά τη διάρκεια του ΠΜΣ. Επίσης, ευχαριστούμε τον επιβλέπων καθηγητή κ. Διαμαντάρα Κωνσταντίνο για τη συνεργασία και τις συμβουλές του, καθώς και τον κ. Σαρόγλου Στέλιο για τη βοήθειά του σε τεχνικά θέματα.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

AE	Auto – encoding
AI	Artificial Intelligence
API	Application Programming Interface
AR	Auto – regressive
BMP	Bitmap
CAGR	Compound Annual Growth Rate
CLM	Casual Language Modeling
CMCL	Cross – Modal Contrastive Learning
CMMLM	Cross – Modal Masked Language Matching
CMU	Carnegie Mellon University
CSV	Comma-Separated values
DDoS	Distributed Denial – of – Service
DOM	Document Object Model
DL	Deep Learning
DPO	Direct Preference Optimization
ED	Encoder – Decoder
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
HNSW	Hierarchical Navigable Small World
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IRQA	Information Retrieval and Question Answering
ITM	Image – Text Matching
IVF	Inverted File Index
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation

LLMs	Large Language Models
LM	Language Models
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Modeling
MOOCs	Massive Open Online Courses
MRM	Masked Region Modeling
NLP	Natural Language Process
NLU	Natural Language Understanding
OCR	Optical Character Recognition
PDF	Portable Document Format
PNG	Portable Network Graphics
RAG	Retrieval – Augmented Generation
RNNs	Recurrent Neural Networks
SFT	Supervised Fine-Tuning
SUS	System Usability Scale
TXT	Text File Format
VAE	Variational Auto – encoder

Κεφάλαιο 1ο: Τα chatbots, LLMs και RAG στη σημερινή εποχή

1.1 Εισαγωγή

Η Τεχνητή Νοημοσύνη (Artificial Intelligence – AI) έχει μεταμορφώσει τον τρόπο με τον οποίο αλληλεπιδρούμε με την τεχνολογία. Ένας πολλά υποσχόμενος τομέας είναι η ανάπτυξη και η βελτίωση συστημάτων διαλογικών βοηθών (Chatbot). Το chatbot είναι ένα μοντέλο αλληλεπίδρασης ανθρώπου – υπολογιστή που προσομοιώνει μια συνομιλία μεταξύ μηχανών και χρηστών. Το τεράστιο εύρος της έρευνας σε αυτόν τον τομέα και οι πρόσφατες εξελίξεις σε τεχνολογίες όπως η βαθιά μάθηση (Deep Learning - DL), η ανάκτηση πληροφοριών (Information Retrieval) και η εμφάνιση γλωσσικών μοντέλων (Language Models – LM), έχουν οδηγήσει σε έξυπνα συστήματα chatbot που μαθαίνουν και βελτιώνονται συνεχώς με την πάροδο του χρόνου [1].

Ο κλάδος των chatbots έχει σημειώσει σημαντική ανάπτυξη τα τελευταία χρόνια, για να μπορέσει να ανταποκριθεί στις απαιτήσεις των πελατών και στην παροχή εξυπηρέτησης 24/7, μειώνοντας έτσι και το λειτουργικό κόστος των επιχειρήσεων. Σύμφωνα με την έρευνα grand view, το μέγεθος της παγκόσμιας αγοράς chatbot εκτιμήθηκε σε 5.132,8 εκατομμύρια USD το 2022 και από το 2023 έως το 2030 αναμένεται να αυξηθεί σε CAGR (Compound Annual Growth Rate) 23,3%. Σύμφωνα με τις αγορές και την έρευνα αγοράς το μέγεθος της αγοράς chatbot αναμένεται να αυξηθεί από 2,9 δισεκατομμύρια δολάρια ΗΠΑ το 2020 σε 10,5 δισεκατομμύρια δολάρια ΗΠΑ έως το 2026, με σύνθετο ετήσιο ρυθμό ανάπτυξης (CAGR) 23,5%. Τα chatbots έχουν γίνει διαδεδομένα σε διάφορους τομείς, όπως το ηλεκτρονικό εμπόριο, η υγειονομική περίθαλψη, η φιλοξενία, ο τουρισμός, οι τράπεζες και η εξυπηρέτηση πελατών [1].

Στην ψηφιακή εποχή, η εκθετική αύξηση των δεδομένων έχει κάνει απαραίτητη την αποτελεσματική ανάκτηση των πληροφοριών και τις λύσεις διαχείρισης αυτών. Σε αυτό το πλαίσιο, οι τεχνολογίες επεξεργασίας φυσικής γλώσσας (Natural Language Process – NLP) έχουν εξελιχθεί και αποτελούν ουσιαστικά εργαλεία για την εξαγωγή πληροφοριών από μεγάλα σύνολα δεδομένων σε ερωτήματα των χρηστών. Ιδιαίτερα οι διαλογικοί βοηθοί έχουν ελκύσει το ενδιαφέρον για την ικανότητά τους να χειρίζονται περίπλοκα έγγραφα και να απαντούν ταυτόχρονα σε ερωτήσεις των χρηστών. Η αποτελεσματικότητα των chatbots που χρησιμοποιούνται σήμερα για την κάλυψη αυτών των αναγκών έχει αμφισβητηθεί, καθώς παρουσιάζουν σημαντικά μειονεκτήματα που επηρεάζουν τόσο τη λειτουργικότητά τους όσο και την ικανοποίηση των χρηστών [2].

Πριν την κυκλοφορία του Chat-GPT της OpenAI, τον Νοέμβριο του 2022, οι εταιρείες χρησιμοποιούσαν εσωτερικά αναπτυγμένα chatbots βασισμένα σε διαλογικές ροές. Αυτά τα πρώιμα chatbots απαιτούσαν εκτεταμένη εκπαίδευση για την κατανόηση προθέσεων και τον λεπτομερή συντονισμό για τη δημιουργία απαντήσεων. Χτισμένα σε συστήματα διαχείρισης διαλόγου, συνδυασμένα με λύσεις ανάκτησης πληροφοριών και απάντησης ερωτήσεων (Information Retrieval and Question Answering – IRQA), ήταν εύθραυστα και περιορισμένης δυναμικότητας, δε διέθεταν την ακρίβεια, την αντοχή και την αξιοπιστία που απαιτούνται για ευρεία χρήση σε επιχειρήσεις και οργανισμούς [3]. Επίσης, οι παρανοήσεις και η ανακριβής εξαγωγή πληροφοριών, επιδεινώνονταν από τα γλωσσικά χαρακτηριστικά και την εξειδικευμένη ορολογία ενός δεδομένου πεδίου. Έτσι, ο περιορισμός της ικανότητάς τους να προσφέρουν ενδελεχή ανάκτηση πληροφοριών σε διάφορες πηγές τεκμηρίωσης, οδήγησε στην περιορισμένη εστίαση των συστημάτων σε συγκεκριμένα θέματα ή τομείς, καθιστώντας επείγουσα την απαίτηση να ξεπεραστούν αυτές οι προκλήσεις και να δημιουργηθούν

ισχυρότερες και αποτελεσματικότερες λύσεις που μπορούν να χειριστούν την πολυπλοκότητα των σύγχρονων ζητημάτων διαχείρισης πληροφοριών [2].

Η κυκλοφορία του Chat-GPT, η εμφάνιση διανυσματικών βάσεων δεδομένων (Vector DB) και η ευρεία χρήση της αρχιτεκτονικής Retrieval-Augmented Generation (RAG) σηματοδότησαν την αρχή μιας νέας εποχής στον τομέα των chatbots. Σήμερα τα LLMs είναι χτισμένα σε τεράστια νευρωνικά δίκτυα, εκπαιδευμένα σε ποικίλα και εκτεταμένα σύνολα δεδομένων, επιτρέποντάς τα να εκτελούν διάφορες εργασίες NLP, από γλωσσικές μεταφράσεις έως την απάντηση περίπλοκων ερωτήσεων [4]. Η αρχιτεκτονική RAG είναι ένα πλαίσιο εργασίας που έχει σχεδιαστεί, για να βελτιώνει την ακρίβεια των απαντήσεων που δημιουργούνται από τα LLMs. Αυτό το επιτυγχάνει ενσωματώνοντας εξωτερικές πηγές γνώσης, για να συμπληρώσει την εσωτερική κατανόηση του μοντέλου. Οι εξωτερικές πληροφορίες που συλλέγονται προστίθενται στην είσοδο του χρήστη και παρέχονται στο γλωσσικό μοντέλο. Το γλωσσικό μοντέλο χρησιμοποιεί τόσο την εκτεταμένη είσοδο όσο και τις εσωτερικές του γνώσεις, για να παρέχει μια εξατομικευμένη απάντηση στον χρήστη μέσω του chatbot [1], [4].

Η παραπάνω προσέγγιση ενισχύει τη δυνατότητα των LLMs να έχουν πρόσβαση και να ανακτούν σύγχρονο [3] και ενημερωμένο περιεχόμενο με αποτέλεσμα να παρέχουν πιο ακριβείς και σχετικές απαντήσεις, ανακτώντας σχετικές πληροφορίες από τεράστιες βάσεις δεδομένων ή αποθετήρια γνώσης και στη συνέχεια να δημιουργούν απαντήσεις με βάση αυτά τα συγκεκριμένα δεδομένα [4]. Ακόμα, τα LLMs μπορούν να κατανοήσουν καλύτερα τις προθέσεις των χρηστών με απλές προτροπές σε φυσική γλώσσα, να συνθέσουν περιεχόμενο με συνοχή, εξαλείφοντας την ανάγκη πολύπλοκης εκπαίδευσης των μοντέλων, δίνοντας έτσι τη δυνατότητα στα chatbots να βελτιώσουν την αποτελεσματικότητά τους και να αυξήσουν την ικανοποίηση των χρηστών, παρέχοντας πιο λεπτομερείς και συναφείς πληροφορίες [3 – 4].

Η τεχνητή νοημοσύνη αποτελεί αναδυόμενη τάση σε πολλούς κλάδους, συμπεριλαμβανομένης και της εκπαίδευσης, με στόχο την αυτοματοποίηση ορισμένων διαδικασιών [5]. Τα συστήματα ενισχυμένης μάθησης εξελίχθηκαν από το e-learning και το mobile learning σε «έξυπνα» περιβάλλοντα μάθησης, προσφέροντας εξατομικευμένες και ευέλικτες εμπειρίες στους μαθητές [6]. Σύμφωνα με την ανθρώπινη ψυχολογία, κάθε άτομο έχει διαφορετικές ικανότητες και ανάγκες: κάποιοι μαθαίνουν γρήγορα, ενώ άλλοι χρειάζονται περισσότερη υποστήριξη, κάποιοι είναι διστακτικοί στο να εκφράσουν ερωτήσεις, καθώς φοβούνται την κριτική. Παράλληλα, οι συναισθηματικές αντιδράσεις των εκπαιδευτικών (όπως αγένεια ή αποθάρρυνση) μπορούν να επηρεάσουν αρνητικά τους μαθητές. Η τεχνητή νοημοσύνη μπορεί να προσφέρει μια λύση σε αυτά τα θέματα [5].

Η τεχνητή νοημοσύνη προσφέρει σημαντικά οφέλη στους μαθητές, καθώς οι εκπαιδευτικοί διαλογικοί βοηθοί μπορούν να προσαρμόσουν τη διδασκαλία στις ανάγκες του κάθε μαθητή, χωρίς συναισθηματικές διακυμάνσεις, επιτρέποντας την εκμάθηση στον δικό τους χρόνο και ρυθμό. Επιπλέον, οι φοιτητές αισθάνονται πιο άνετα να θέτουν ερωτήσεις χωρίς τον φόβο της κριτικής ή της ταπείνωσης. Τα εργαλεία αυτά προσφέρουν επίσης συνεχόμενη, ποιοτική και άμεση ανατροφοδότηση στους μαθητές [5], [7]. Ωστόσο, η απουσία ανθρώπινων συναισθημάτων και αλληλεπίδρασης μπορεί να οδηγήσει σε έλλειψη κινήτρων, περιορισμένη κοινωνικοποίηση και απώλεια της εμπειρίας που προσφέρει η ανθρώπινη καθοδήγηση [5].

Από την οπτική των εκπαιδευτικών, οι εφαρμογές της τεχνητής νοημοσύνης μπορούν να χρησιμοποιηθούν ως πολύ αποτελεσματικά υποστηρικτικά εργαλεία, επειδή είναι σε θέση να τους απαλλάξουν από κουραστικές, ενεργοβόρες και χρονοβόρες δραστηριότητες, όπως διορθώσεις γραπτών. Οι εκπαιδευτικοί έχουν τη δυνατότητα να δημιουργούν πιο αποτελεσματικά μαθησιακά περιβάλλοντα, να αναλύουν την πρόοδο των μαθητών, να διαγνώσουν το μαθησιακό τους επίπεδο με

αποτέλεσμα να προσαρμόζουν το περιεχόμενο της μάθησης στις ανάγκες των μαθητών [7 – 8]. Ωστόσο, οι εκπαιδευτικοί ενδέχεται να αντιμετωπίσουν δυσκολίες, καθώς η τεχνητή νοημοσύνη αναλαμβάνει όλο και περισσότερες από τις αρμοδιότητές τους. Τα συστήματα AI, τα οποία βασίζονται σε δεδομένα που παρέχουν οι ίδιοι, μπορούν να τους υποκαθιστούν σε μεγάλο βαθμό, περιορίζοντάς τους σε ρόλους που σχετίζονται με τη διαχείριση δεδομένων και τον έλεγχο της σωστής λειτουργίας των συστημάτων [9 – 10]. Παρόλα αυτά, οι εκπαιδευτικοί με συναισθηματική νοημοσύνη θα εξακολουθήσουν να είναι απαραίτητοι, ιδιαίτερα έως ότου οι AI συσκευές αποκτήσουν τη δυνατότητα συναισθηματικής κατανόησης [5].

Τέλος, τα Chatbots με αρχιτεκτονική RAG προσφέρουν εξειδικευμένη υποστήριξη, ενσωματώνοντας προηγμένα μοντέλα επεξεργασίας φυσικής γλώσσας και συγκεκριμένες βάσεις δεδομένων. Αυτά τα εργαλεία μπορούν να ανταποκριθούν στις μοναδικές ανάγκες των μαθητών, βελτιώνοντας την εμπειρία μάθησης και ενισχύοντας τα εκπαιδευτικά αποτελέσματα [4].

Το παρόν έγγραφο αναφέρεται στη δημιουργία ενός εκπαιδευτικού chatbot που συνδυάζει ένα LLM με τεχνικές RAG, σχεδιασμένο για το μάθημα της Γεωγραφίας της Ε' Δημοτικού και τη ζωή του Αριστοτέλη. Στόχος του συστήματος είναι να δέχεται ερωτήσεις μαθητών και να παρέχει απαντήσεις βασισμένες σε σχολικά βιβλία, τα οποία αποτελούν επιμελημένες και αξιόπιστες εκπαιδευτικές πηγές.

Τα εκπαιδευτικά δεδομένα προέρχονται από αρχεία PDF, από τα οποία έχει εξαχθεί το κείμενο, οι εικόνες και οι πίνακες, καθώς και από αρχεία CSV και TXT. Το μοντέλο αξιοποιεί αυτές τις εξωτερικές πηγές δεδομένων ενσωματώνοντάς τες στο RAG, διασπώντας τες σε μικρότερα τμήματα (chunks) και αποθηκεύοντάς τες στη διανυσματική βάση ChromaDB ως ενσωματώσεις (embeddings). Η ανάκτηση των πληροφοριών βασίζεται στην cosine similarity, επιτρέποντας στο LLM, μέσω των τεχνικών RAG, να αντλεί δεδομένα από τα σχολικά βιβλία. Έτσι, ένας μαθητής μπορεί να υποβάλει ερωτήματα και να λάβει σχετικές πληροφορίες σε μορφή κειμένου.

Ακολουθεί μια αναφορά σε σχετικές εργασίες που έχουν γίνει σχετικά με εκπαιδευτικούς διαλογικούς βοηθούς. Στο 2ο κεφάλαιο γίνεται μια παρουσίαση των χαρακτηριστικών, των τεχνικών στοιχείων, της λειτουργίας, των πλεονεκτημάτων και των προκλήσεων των LLMs και της αρχιτεκτονικής RAG. Το κεφάλαιο 3 αναφέρεται στη μεθοδολογία που ακολουθήθηκε για την κατασκευή του μοντέλου και τα στάδια υλοποίησης, όπου περιγράφονται τα εργαλεία που χρησιμοποιήθηκαν. Το 4ο κεφάλαιο αποτελεί ένα εγχειρίδιο χρήσης της εφαρμογής. Υπάρχουν εικόνες από τη διεπαφή χρήστη και περιγραφή του τρόπου χρήσης της εφαρμογής από τους μαθητές. Στο κεφάλαιο 5 γίνεται μια αξιολόγηση της εφαρμογής από τους μαθητές, αλλά και εκπαιδευτικούς, κυρίως της Ε΄ τάξης. Η αξιολόγηση βασίζεται σε ερωτηματολόγιο. Παρουσιάζονται τα αποτελέσματα και γραφικές αναπαραστάσεις. Τέλος, ακολουθεί το κεφάλαιο 6 με συμπεράσματα.

1.2 Σχετικές εργασίες

Οι ερευνητές, όλο και περισσότερο, επικεντρώνουν το ενδιαφέρον τους στην ανάπτυξη διαλογικών βοηθών που βασίζονται σε LLMs, καθώς και σε chatbots που ενσωματώνουν την αρχιτεκτονική Retrieval Augmented Generation με τα LLMs. Πρόσφατες μελέτες έχουν εξερευνήσει διάφορες εφαρμογές αυτών των τεχνικών, οι οποίες αποδεικνύουν την αποτελεσματικότητά τους σε πολλαπλές εφαρμογές.

Οι Neyem et al. [11] διεξήγαγαν μια μελέτη με 150 φοιτητές. Εξέτασαν στρατηγικές όπως άμεσες αναζητήσεις στη βάση δεδομένων μαθησιακών εμπειριών, ερωτήματα στο προ – εκπαιδευμένο μετασχηματιστικό μοντέλο GPT και εμπλουτισμό ερωτημάτων με δεδομένα από τα μαθήματα. Επιπλέον, ενίσχυσαν τα ερωτήματα με δεδομένα από το Stack Overflow πριν από την επεξεργασία από

το GPT. Η ανάλυση έδειξε ότι τα κείμενα που παρήγαγε το Μεγάλο Γλωσσικό Μοντέλο ανταποκρίνονταν στις γλωσσικές προδιαγραφές και τη θεματική συνάφεια των απαιτήσεων πανεπιστημιακών μαθημάτων. Ωστόσο, το μοντέλο τους παρουσίασε έναν περιορισμό, καθώς παρείχε γενικές συστάσεις που δεν ήταν προσαρμοσμένες στο προφίλ του χρήστη.

Οι Dieu, Nguyen Thi και Thao Thi Nguyen [12] ερεύνησαν τη χρήση προηγμένων τεχνικών RAG για τη βελτίωση των AI πρακτόρων σε σχέση με την ακαδημαϊκή συμβουλευτική στην ανώτατη εκπαίδευση. Με την ενσωμάτωση σύγχρονων γλωσσικών μοντέλων με βάσεις γνώσης, εμπλουτισμό μεταδεδομένων, χρήση του ChromaDB για αποθήκευση δεδομένων και του ZeroTier για ασφάλεια, η μελέτη αποδεικνύει βελτιώσεις στην ανταπόκριση και την ακρίβεια. Τα ευρήματα αναδεικνύουν τη μεταμορφωτική δυναμική της τεχνητής νοημοσύνης στην ανώτατη εκπαίδευση και υπογραμμίζουν την ανάγκη για περαιτέρω έρευνα σε αυτόν τον συνεχώς εξελισσόμενο τομέα.

Οι Martinez-Araneda et al. [13] παρουσιάζουν την ανάπτυξη του TutorBot+, ενός εργαλείου ανατροφοδότησης βασισμένου στο ChatGPT, που έχει σχεδιαστεί για φοιτητές προγραμματισμού στο Πανεπιστήμιο Universidad Católica de la Santísima Concepción. Στόχος του είναι η ενίσχυση της μάθησης και της υπολογιστικής σκέψης. Τα αρχικά ευρήματα δείχνουν ότι ενσωματώθηκε επιτυχώς σε ένα LMS και έχει τη δυνατότητα να βελτιώσει την εκπαιδευτική εμπειρία.

Οι Neeraj Singh Amarnath και Rajganesh Nagarajan [14] παρουσιάζουν ένα καινοτόμο chatbot τεχνητής νοημοσύνης, σχεδιασμένο να προσφέρει εξατομικευμένη καθοδήγηση σε μαθητές λυκείου. Αξιοποιώντας τα γλωσσικά μοντέλα της OpenAI, gpt-3.5-turbo και gpt-4.0, το chatbot συγκεντρώνει πληροφορίες από εκπαιδευτικούς οδηγούς και παρέχει προσαρμοσμένες συστάσεις, βοηθώντας τους μαθητές να διαμορφώσουν την εκπαιδευτική τους πορεία. Χάρη στην ικανότητά του να προσφέρει εξατομικευμένη υποστήριξη, να ενισχύει τη διαδικασία λήψης αποφάσεων και να βελτιώνει την κατανόηση, αποτελεί ένα πολύτιμο εργαλείο μάθησης. Συνολικά, το chatbot έχει τη δυνατότητα να μετασχηματίσει την εμπειρία των μαθητών στο λύκειο, παρέχοντας αξιόπιστη, εξατομικευμένη και εξελισσόμενη υποστήριξη.

Οι Bratić D., Šapina M., Jurečić D. και Žiljak Gršić J. [15] προτείνουν ένα υβριδικό μοντέλο που συνδυάζει τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) με chatbots, διευκολύνοντας την πρόσβαση σε εκπαιδευτικό υλικό στην ανώτατη εκπαίδευση. Τονίζεται η σημασία του prompt engineering και της κατανόησης του περιεχομένου για την αποτελεσματική αξιοποίηση των LLMs. Μέσα από πρακτικές εφαρμογές, όπως chatbots ερωτοαπαντήσεων, αναδεικνύεται η χρησιμότητα του προτεινόμενου προγραμματιστικού πλαισίου.

Οι Miladi et al. [16] μελέτησαν τη χρήση της μεθόδου RAG για τη βελτίωση της ακρίβειας των μοντέλων GPT σε εκπαιδευτικά περιβάλλοντα, ιδίως στα Massive Open Online Courses (MOOCs – Μαζικά Ανοικτά Διαδικτυακά Μαθήματα). Μια συγκριτική ανάλυση διαφόρων εκδόσεων των μοντέλων GPT έδειξε σημαντική βελτίωση στην ακρίβεια με την ενίσχυση του RAG, κυρίως το GPT-4. Αυτή η αναβάθμιση αναδεικνύει τη μεγάλη προοπτική των μοντέλων GPT με RAG στη βελτίωση της ακρίβειας παραγωγής περιεχομένου.

Οι Alexander Tobias Neumann et al. [17] παρουσιάζουν το MoodleBot, ένα chatbot ανοιχτού κώδικα που έχει σχεδιαστεί για να παρέχει ανατροφοδότηση σχετικά με το περιεχόμενο των διαλέξεων και τις εργασίες, υποστηρίζοντας την αυτό – ρυθμιζόμενη μάθηση. Αναπτύχθηκε για το Moodle, ένα ευρέως χρησιμοποιούμενο LMS, και εφαρμόστηκε στους φοιτητές του μαθήματος «Βάσεις Δεδομένων και Πληροφοριακά Συστήματα» στο Πανεπιστήμιο RWTH Aachen. Το MoodleBot προσομοιάζει το στιλ συνομιλίας ενός δασκάλου, προσφέροντας παράλληλα άμεσες απαντήσεις και συνεχή διαθεσιμότητα.

Οι Li J., Yuan Y. και Zhang Z. [18] πρότειναν έναν σχεδιασμό συστήματος end – to – end που χρησιμοποιεί τη μέθοδο RAG για τη βελτίωση της πραγματολογικής ακρίβειας των μεγάλων γλωσσικών μοντέλων (LLMs) σε εξειδικευμένα και χρονικά ευαίσθητα ερωτήματα που σχετίζονται με ιδιωτικές βάσεις γνώσεων. Εφάρμοσαν αυτήν την προσέγγιση RAG χρησιμοποιώντας δοκιμαστικά δεδομένα ιστού από ιστοσελίδες του Carnegie Mellon University (CMU), που συλλέχθηκαν μέσω web crawlers. Τα πειράματά τους απέδειξαν την αποτελεσματικότητα του συστήματος στην παραγωγή πιο ακριβών απαντήσεων για εξειδικευμένα και χρονικά κρίσιμα ερωτήματα. Η μελέτη αποκάλυψε επίσης τους περιορισμούς της προσαρμογής των LLMs όταν χρησιμοποιούνται μικρής κλίμακας και μη αντιπροσωπευτικά σύνολα δεδομένων.

Οι Udayan et al. [19] ανέπτυξαν ένα chatbot χρησιμοποιώντας την αρχιτεκτονική Long Short-Term Memory (LSTM) για να απαντά σε ερωτήσεις φοιτητών. Το chatbot εκπαιδεύτηκε σε λέξεις-κλειδιά, τις οποίες χρησιμοποιεί για να κατανοήσει το περιεχόμενο της ερώτησης του χρήστη. Όταν ο χρήστης εισάγει ένα ερώτημα, το σύστημα εντοπίζει τις λέξεις-κλειδιά, αναλύει την ερώτηση και ανακτά τις προκαθορισμένες απαντήσεις. Ωστόσο, τα συστήματα LSTM έχουν εξελιχθεί και αντικατασταθεί από την αρχιτεκτονική των μετασχηματιστών (transformers), λόγω της ανώτερης ικανότητάς τους στην επεξεργασία φυσικής γλώσσας, κάτι που τα καθιστά πιο ευέλικτα και αποτελεσματικά για τους τελικούς χρήστες. Επιπλέον, το chatbot τους δεν χρησιμοποιεί Μεγάλα Γλωσσικά Μοντέλα (LLMs) για τη δημιουργία απαντήσεων, γεγονός που μειώνει σημαντικά την ικανότητα του συστήματος να κατανοεί τις ερωτήσεις, να αντιλαμβάνεται το απαιτούμενο περιεχόμενο και να παρέχει πιο ακριβείς απαντήσεις.

Οι Thway et al. [20] ανέπτυξαν τον Professor Leodar, ένα προσαρμοσμένο chatbot βασισμένο στη μέθοδο RAG, το οποίο επικοινωνεί στη διάλεκτο Singlish και ενσωματώνει δυνατότητες Μεγάλων Γλωσσικών Μοντέλων (LLM) για τη βελτίωση της εκπαίδευσης. Ο Professor Leodar αναπτύχθηκε στο Πανεπιστήμιο Τεχνολογίας Nanyang στη Σιγκαπούρη και παρέχει πληροφορίες με σχετικό περιεχόμενο, βοηθώντας τους φοιτητές στη μάθηση, τη συμμετοχή και την προετοιμασία για εξετάσεις. Η υλοποίηση του συστήματος ανέδειξε τα οφέλη των εξατομικευμένων απαντήσεων σε εκπαιδευτικά περιβάλλοντα. Το σύστημα χρησιμοποιεί το Claude 3 και απαντά κυρίως στη διάλεκτο Singlish. Ωστόσο, δεδομένου ότι το Singlish είναι μια υποκατηγορία των αγγλικών που τα περισσότερα LLMs δεν έχουν εκπαιδευτεί ειδικά να επεξεργάζονται, η χρήση βασικών αγγλικών πιθανότατα θα απέδιδε καλύτερα αποτελέσματα στη δημιουργία απαντήσεων.

Οι Odede και Frommholz [21] παρουσίασαν το JayBot, ένα chatbot βασισμένο σε Μεγάλα Γλωσσικά Μοντέλα με στόχο τη βελτίωση της εμπειρίας των υποψήφιων και νυν φοιτητών, του διδακτικού προσωπικού και του προσωπικού διοίκησης σε ένα πανεπιστήμιο του Ηνωμένου Βασιλείου. Το JayBot παρέχει πληροφορίες σχετικά με γενικές ερωτήσεις για τα πανεπιστημιακά προγράμματα, τη διάρκειά τους, τα δίδακτρα, τις προϋποθέσεις εισαγωγής, τους διδάσκοντες, τις πρακτικές ασκήσεις, τις επαγγελματικές προοπτικές και τη δυνατότητα απορρόφησης στην αγορά εργασίας. Χρησιμοποιώντας AI, το chatbot κατασκευάστηκε με το προηγμένο μεγάλο γλωσσικό μοντέλο GPT-3.5 turbo της OpenAI. Για την αντιμετώπιση προκλήσεων όπως οι "παραισθήσεις" (hallucinations), η ακρίβεια και η επικαιρότητα των απαντήσεων, ενσωματώθηκε ένα μοντέλο μετασχηματιστή ενσωματώσεων (embedding transformer) σε συνδυασμό με μια διανυσματική βάση δεδομένων και με διανυσματική αναζήτηση. Η παρουσίαση του συστήματος αναδεικνύει την αποτελεσματικότητα του συνδυασμού LLM και RAG στην ανάκτηση σχετικών πληροφοριών, διευκολύνοντας τους φοιτητές στην πλοήγηση σε ένα εκπαιδευτικό περιβάλλον γεμάτο πληροφορίες και συχνά δύσκολο στην κατανόηση.

Ο Nakhod [22] παρουσιάζει ένα μοντέλο RAG που ενσωματώνει πληροφορίες σε LLMs για την υποστήριξη προγραμματιστών low-code. Το μοντέλο χρησιμοποιεί μια βάση δεδομένων διανυσμάτων

για την αποθήκευση και ανάκτηση εξειδικευμένων πληροφοριών, βελτιώνοντας τις απαντήσεις με σχετικό περιεχόμενο. Μέσω της μεθόδου ομοιότητας συνημιτόνου (cosine similarity) για την ανάκτηση εγγράφων, αυξάνει την ποιότητα των απαντήσεων. Τα αποτελέσματα δείχνουν ότι τα μοντέλα RAG ξεπερνούν τα βασικά LLMs σε ακρίβεια και συνάφεια. Ωστόσο, η μελέτη αναγνωρίζει περιορισμούς στο εύρος της και προτείνει μελλοντικές βελτιώσεις στις τεχνικές αποθήκευσης και ανάκτησης δεδομένων.

Το Al-TA είναι ένα σύστημα που στοχεύει στην αυτοματοποίηση των απαντήσεων σε ερωτήσεις φοιτητών σε πλατφόρμες όπως το Piazza, μειώνοντας τον φόρτο εργασίας σε μεγάλα μαθήματα πληροφορικής και παρουσιάζεται από τους Y. Ricke, A. Agarwal, Q. Ma και P. Denny [23]. Χρησιμοποιεί σύγχρονες μεθόδους, όπως RAG, Supervised Fine-Tuning (SFT) και Direct Preference Optimization (DPO), βασισμένες σε ανοιχτού κώδικα LLMs της οικογένειας LLaMA-2. Το RAG συμβάλλει κατά 30% στη βελτίωση της ποιότητας των απαντήσεων σε ένα εισαγωγικό μάθημα Πληροφορικής. Αν και το σύστημα έχει σημειώσει επιτυχία, βρίσκεται ακόμα στα στάδια της δομημένης εισόδου και της χειροκίνητης βελτιστοποίησης, με προοπτικές περαιτέρω αυτοματοποίησης και προσαρμογής σε διάφορα εκπαιδευτικά περιβάλλοντα.

Οι Tianshi Zheng et al. [24] Η παρούσα μελέτη ασχολείται με την ανάπτυξη ενός συστήματος RAG βασισμένου σε ένα μεγάλο γλωσσικό μοντέλο, προσαρμοσμένου για μαθήματα πληροφορικής σε προπτυχιακό και μεταπτυχιακό επίπεδο. Διερευνάται η δυνατότητα του RAG να λειτουργεί ως εικονικός βοηθός διδασκαλίας και ως εργαλείο υποστήριξης για το διδακτικό προσωπικό στην ανώτατη εκπαίδευση. Τέλος, αναλύονται οι απαιτούμενες βελτιώσεις για την αποτελεσματική ενσωμάτωση του RAG στην ψηφιακή εκπαίδευση, βάσει αξιολογήσεων και σχολίων των εκπαιδευτικών.

Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο

2.1 Εισαγωγή

Το κεφάλαιο αυτό εισάγει τις έννοιες του LLM και της αρχιτεκτονικής RAG, ενημερώνοντας τον αναγνώστη και παρέχοντάς του μια θεμελιώδη πληροφόρηση για τα χαρακτηριστικά των εννοιών και των υπολογιστικών μεθόδων που χρησιμοποιούν. Αυτή η επισκόπηση θέτει τις βάσεις για την καλύτερη κατανόηση των μεθοδολογιών και των εφαρμογών που υλοποιήθηκαν στην εργασία, οι οποίες παρουσιάζονται στις επόμενες ενότητες, εστιάζοντας στη δομή και τη λειτουργία τους, χωρίς να εμβαθύνουμε σε τεχνικές λεπτομέρειες.

2.2 Μοντέλα

Στον τομέα της μηχανικής μάθησης (Machine Learning - ML) και της τεχνητής νοημοσύνης, ένα μοντέλο αναφέρεται σε μια μαθηματική ή υπολογιστική αναπαράσταση ενός συστήματος ή μιας διαδικασίας. Τα μοντέλα έχουν σχεδιαστεί για να καταγράφουν μοτίβα και σχέσεις μέσα στα δεδομένα, επιτρέποντάς τα να κάνουν προβλέψεις, ταξινομήσεις ή να παράγουν αποτελέσματα με βάση νέες εισόδους [25].

Τα μοντέλα συνήθως εκπαιδεύονται σε μεγάλα σύνολα δεδομένων, χρησιμοποιώντας διάφορους αλγόριθμους και τεχνικές, όπως νευρωνικά δίκτυα, δέντρα αποφάσεων ή μηχανές υποστήριξης διανυσμάτων. Η διαδικασία εκπαίδευσης περιλαμβάνει την προσαρμογή των παραμέτρων του μοντέλου για να ελαχιστοποιηθεί η ασυνέπεια μεταξύ των προβλέψεών του και των πραγματικών, επιθυμητών αποτελεσμάτων. Αυτό συχνά περιλαμβάνει τεχνικές από επαναληπτική και σταδιακή ανάπτυξη στη μηχανική λογισμικού, οι οποίες έχουν προσαρμοστεί για τη βελτιστοποίηση των διαδικασιών μάθησης σε συστήματα τεχνητής νοημοσύνης [25].

Τα νευρωνικά δίκτυα, ειδικότερα, είναι μια κατηγορία μοντέλων εμπνευσμένων από τη δομή και τη λειτουργία των βιολογικών νευρωνικών δικτύων στον ανθρώπινο εγκέφαλο. Αποτελούνται από διασυνδεδεμένους κόμβους (νευρώνες) που επεξεργάζονται και μεταδίδουν πληροφορίες, επιτρέποντάς τους να μαθαίνουν και να κάνουν προβλέψεις από δεδομένα. Η αρχιτεκτονική και η λειτουργία των νευρωνικών δικτύων έχουν επηρεαστεί σημαντικά από τις εξελίξεις στην κατανόηση του τρόπου με τον οποίο μεμονωμένες μονάδες σε ένα δίκτυο συμβάλλουν στη συνολική του απόδοση [26].

Επιπλέον, η εφαρμογή των νευρωνικών δικτύων σε εκπαιδευτικά περιβάλλοντα έχει αποδείξει την ικανότητά τους να βελτιώνουν τις εξατομικευμένες μαθησιακές εμπειρίες, επιδεικνύοντας την προσαρμοστικότητά τους σε διάφορους τομείς. Τα μοντέλα νευρωνικών δικτύων όχι μόνο μιμούνται τις γνωστικές λειτουργίες, αλλά επεκτείνονται και σε εφαρμογές όπου διευκολύνουν την επεξεργασία δεδομένων και τη λήψη αποφάσεων σε πραγματικό χρόνο [26].

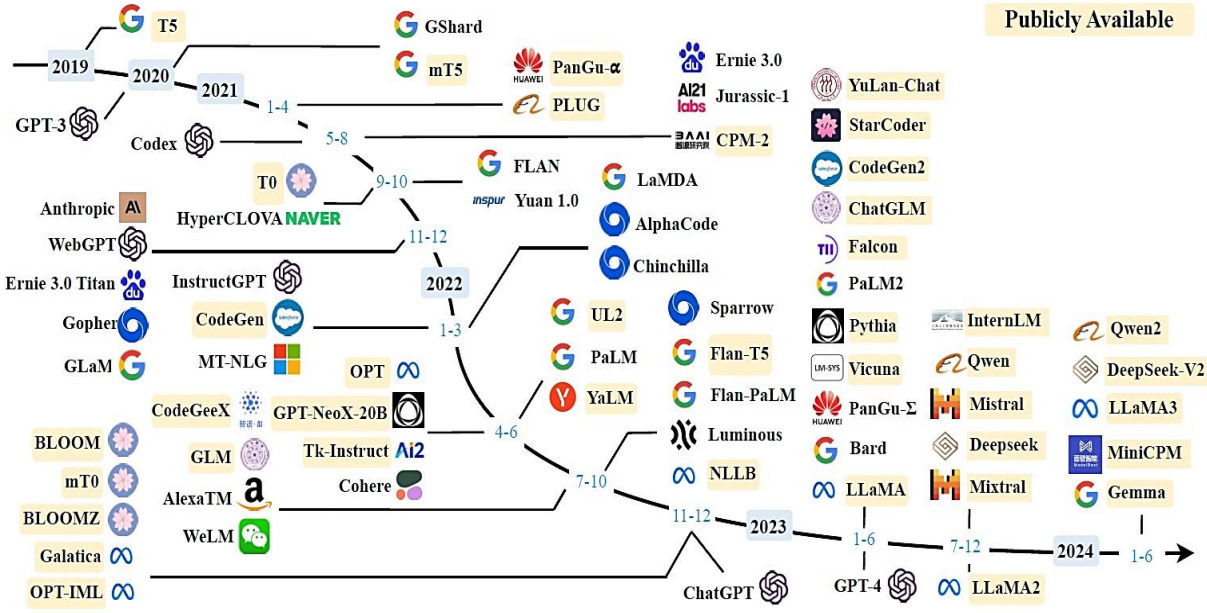
2.3 Large Language Models (LLMs - Μεγάλα Γλωσσικά Μοντέλα)

Τα LLMs διατηρούν τα βασικά χαρακτηριστικά ενός μοντέλου, όπως αναφέρθηκε παραπάνω. Είναι μοντέλα βαθιάς μάθησης που έχουν σχεδιαστεί για την επεξεργασία της φυσικής γλώσσας, γεγονός που τους δίνει τη δυνατότητα να κατανοούν και να δημιουργούν κείμενο που μοιάζει ανθρώπινο, μοντελοποιώντας περίπλοκα γλωσσικά μοτίβα και σημασιολογικές σχέσεις. Χαρακτηρίζονται από το τεράστιο μέγεθός τους, που συνήθως περιλαμβάνει εκατοντάδες εκατομμύρια έως δισεκατομμύρια παραμέτρους και την εκπαίδευσή τους σε τεράστιες ποσότητες δεδομένων κειμένου από διαφορετικές πηγές. Η χρήση τους έχει ξεπεράσει τα όρια των παραδοσιακών γλωσσικών εργασιών, διεισδύοντας σε

διάφορους τομείς, όπως η υγειονομική περίθαλψη, τα οικονομικά, η εξυπηρέτηση πελατών, η εκπαίδευση και οι μεταφορές [27 – 28].

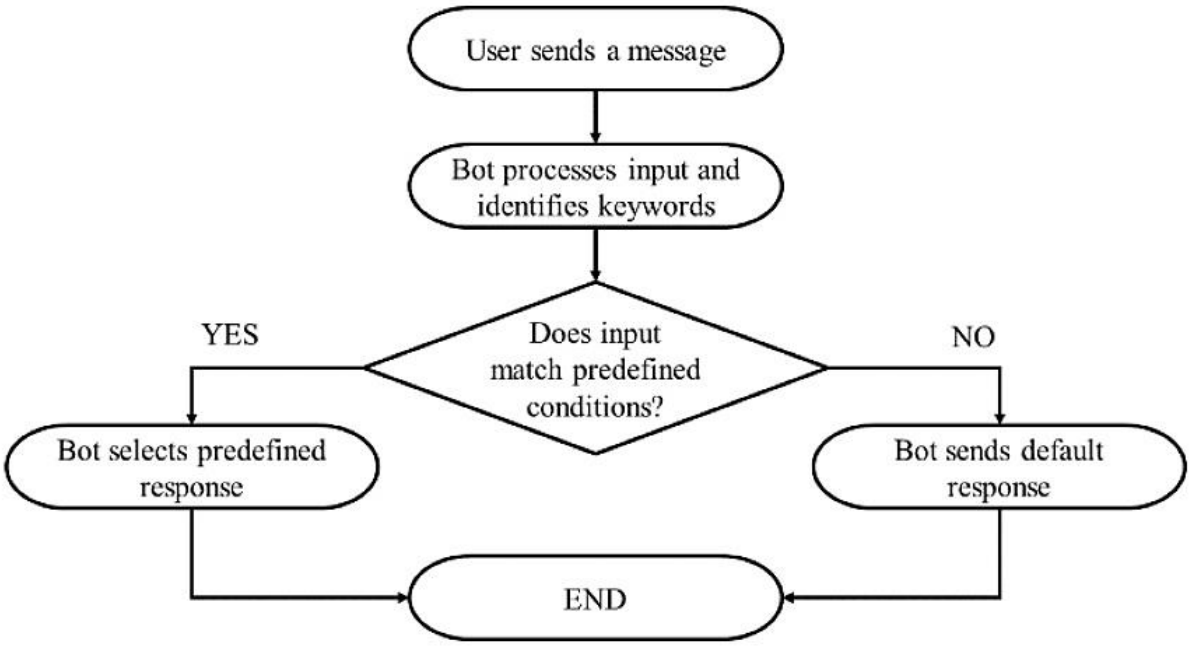
Τα LLMs έχουν εξελιχθεί σημαντικά με τα χρόνια, το σχήμα 2.1 παρουσιάζει την εξέλιξη των LLMs με περισσότερες από 10 δις παραμέτρους [29], με την ιστορία τους να εκτείνεται από συστήματα που βασίζονται σε κανόνες (Rule Based), σχήμα 2.2 [30], έως τα σημερινά μοντέλα που βασίζονται σε βαθιά μάθηση. Η ιστορική τους πορεία είναι η εξής:

- Πρώιμα γλωσσικά μοντέλα: Τα πρώιμα γλωσσικά μοντέλα βασίζονταν σε προκαθορισμένους γραμματικούς και γλωσσικούς κανόνες. Είχαν περιορισμένες δυνατότητες στην κατανόηση του περιεχομένου (context) και δεν είχαν την ικανότητα να μαθαίνουν από τα δεδομένα.
- Στατιστικά γλωσσικά μοντέλα: Με την εμφάνιση των στατιστικών μεθόδων, τα γλωσσικά μοντέλα άρχισαν να χρησιμοποιούν πιθανολογικές προσεγγίσεις, για να προβλέψουν την πιθανότητα μιας λέξης ή φράσης μετά από ένα δεδομένο πλαίσιο κειμένου. Παραδείγματα τέτοιων μοντέλων είναι τα N-grams και τα Hidden Markov Models.
- Μεγάλα γλωσσικά μοντέλα που βασίζονται σε βαθιά μάθηση: Η σύγχρονη εποχή των γλωσσικών μοντέλων χαρακτηρίζεται από την άνοδο των τεχνικών βαθιάς μάθησης. Συγκεκριμένα, τα Recurrent Neural Networks (RNNs – Επαναλαμβανόμενα Νευρωνικά Δίκτυα), τα δίκτυα Long Short-Term Memory (LSTM – Μακροπρόθεσμης Βραχυπρόθεσμης Μνήμης) και τα Transformer-based models (μοντέλα που βασίζονται σε μετασχηματιστές), όπως τα BERT, GPT-2 και GPT-3, αποτελούν τους πρωτοπόρους του συγκεκριμένου τομέα. Αυτά τα μοντέλα μπορούν να αποτυπώσουν το περιεχόμενο (context), τη σημασιολογία και τις συντακτικές δομές, καθιστώντας τα εξαιρετικά αποτελεσματικά στην κατανόηση και τη δημιουργία φυσικής γλώσσας [27]. Το σχήμα 2.3 παρουσιάζει κάποιες δημοφιλείς οικογένειες των LLMs. [31]



Σχήμα 2.1: Ιστορική εξέλιξη των LLMs με περισσότερες από 10 δις παραμέτρους

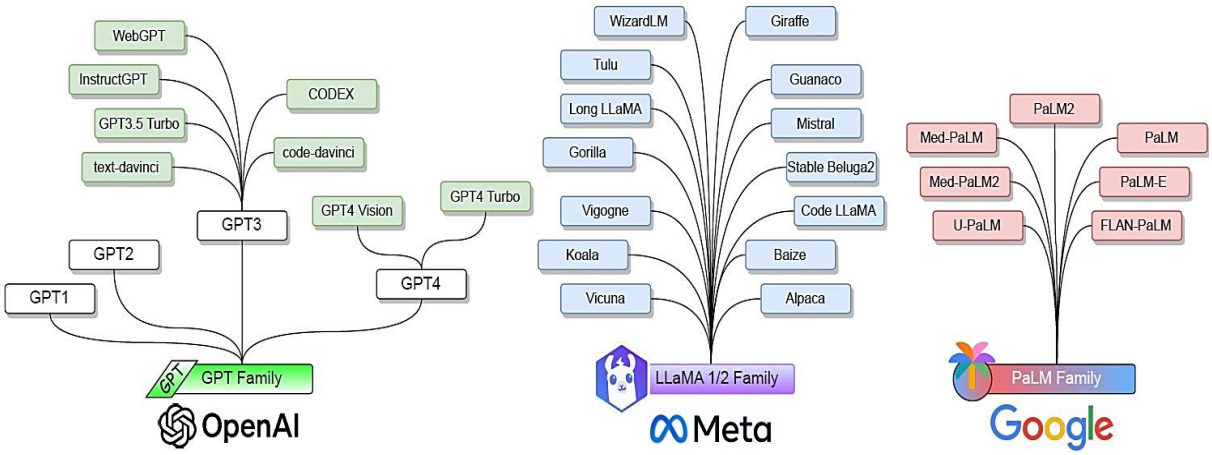
Η εξέλιξη των LLMs, διάρκειας αρκετών ετών, όχι μόνο απέδειξε τη δύναμή τους σε γλωσσικά καθήκοντα, αλλά και την ευελιξία τους στο χειρισμό διαφορετικών μορφών δεδομένων, οδηγώντας στην επεξεργασία και τη δημιουργία πολυμεσικού περιεχομένου. Οι δυνατότητες, σχήμα 2.4 [31] και οι εφαρμογές τους καλύπτουν ένα ευρύ φάσμα πεδίων, συμπεριλαμβανομένης της δημιουργίας οπτικού περιεχομένου, της παραγωγής ήχου, της δημιουργίας κειμένου, της δημιουργίας κώδικα, της αυτοματοποιημένης σχεδίασης και των ρομποτικών διεπαφών.



Σχήμα 2.2: Rule based Chatbot

Η ευελιξία και η αποτελεσματικότητα των LLMs συνέβαλαν στη σημαντική πρόοδο αυτών των τομέων, με πολλές εταιρείες να αναπτύσσουν αξιόλογα προϊόντα, αξιοποιώντας τις δυνατότητές τους. Αξιοσημείωτες εργασίες, στις οποίες χρησιμοποιούνται τα LLMs είναι [32]:

- Η δημιουργία συνεκτικού κειμένου από εισαγωγή δομημένων δεδομένων, έπειτα από τη λήψη σχετικών οδηγιών. Περιλαμβάνει εφαρμογές, όπως η δημιουργική γραφή και τα chatbots.
- Παραγωγή Κώδικα: Τα LLMs έχουν κάνει σημαντικά βήματα στον τομέα της δημιουργίας κώδικα, βοηθώντας τους προγραμματιστές στη σύνταξη και στη βελτιστοποίηση του κώδικα.
- Λογικός συλλογισμός (Logical Reasoning): Ανάλυση και συμπέρασμα που βασίζεται σε λογική εγγενή σε ένα δεδομένο σενάριο.
- Μηχανική μετάφραση σε γλωσσικά πλαίσια εργασίας.
- Σύνοψη εγγράφων και εξαγωγή πληροφοριών: Οι δυνατότητες των LLMs εκτείνονται πέρα από την κατανόηση μεμονωμένων προτάσεων. Μπορούν να συνοψίσουν μεγάλα έγγραφα, να εξάγουν βασικές πληροφορίες και να προσδιορίσουν σχετικές οντότητες, διευκολύνοντας την αποτελεσματική έρευνα και ανάλυση [28].
- Κατανόηση οπτικού περιεχομένου. Εκτός από το κείμενο και τον κώδικα, τα LLMs είναι καθοριστικά για την κατανόηση οπτικού περιεχομένου. Αυτό περιλαμβάνει εργασίες, όπως περιγραφή εικόνας βάσει κειμένου, δημιουργία λεζάντας και ανάγνωση εγγράφων μέσω OCR (Optical Character Recognition - Οπτική Αναγνώριση Χαρακτήρων).
- Πολυτροπική (Multimodal) υποστήριξη: Πέρα από το περιεχόμενο κειμένου, τα LLMs διευκολύνουν τις εισόδους και τις εξόδους σε διάφορες μορφές, συμπεριλαμβανομένων εικόνων, βίντεο, ήχου και αλληλεπιδράσεων σε ρομποτικά περιβάλλοντα, αξιοποιώντας πολλούς τύπους δεδομένων.

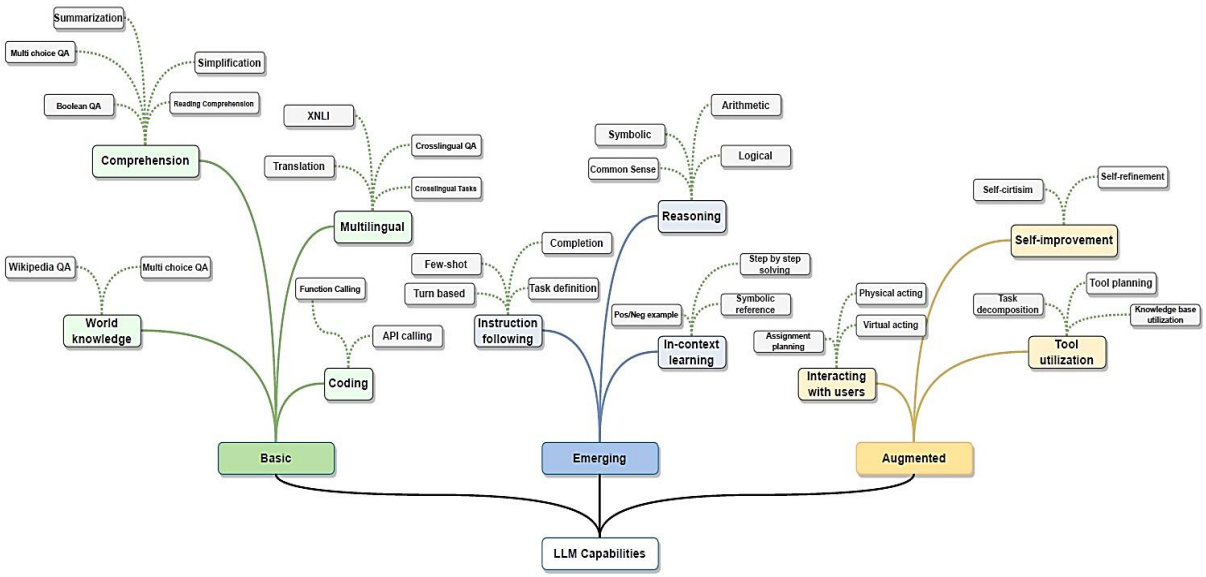


Σχήμα 2.3: Δημοφιλείς οικογένειες των LLMs

Κάποια χαρακτηριστικά στοιχεία της δομής και λειτουργίας των LLMs αναφέρονται παρακάτω.

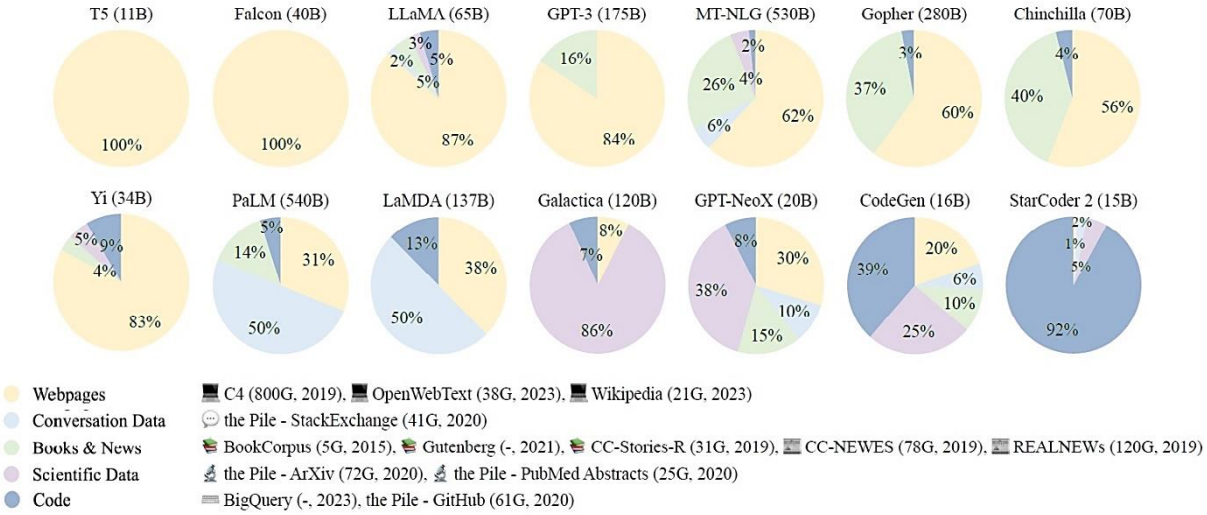
2.3.1 Συλλογή και Προετοιμασία Δεδομένων

Σε σύγκριση με τα γλωσσικά μοντέλα μικρής κλίμακας, τα LLMs απαιτούν δεδομένα υψηλής ποιότητας για την προ – εκπαίδευσή τους. Η ποιότητα της απόδοσής τους βασίζεται σε μεγάλο βαθμό στον τρόπο προ – επεξεργασίας των δεδομένων και στην προ – εκπαίδευσή τους με τα δεδομένα αυτά. Ακολουθεί μια αναφορά στη συλλογή και στην επεξεργασία των δεδομένων, συμπεριλαμβανομένων των πηγών, των μεθόδων προ – επεξεργασίας και της σημαντικής ανάλυσης του τρόπου με τον οποίο τα δεδομένα προ – κατάρτισης επηρεάζουν την απόδοση των LLMs.



Σχήμα 2.4: Δυνατότητες LLMs

Πηγή δεδομένων. Για την ανάπτυξη ενός ικανού LLM, είναι σημαντική η συλλογή ενός μεγάλου όγκου υλικού φυσικής γλώσσας από διάφορες πηγές δεδομένων. Τα υφιστάμενα LLMs αξιοποιούν κυρίως ένα μείγμα από διάφορα δημόσια σύνολα δεδομένων κειμένου ως σώμα πριν από την κατάρτιση. Το σχήμα 2.5 [29] δείχνει την κατανομή των πηγών δεδομένων προ – εκπαίδευσης για έναν αριθμό αντιπροσωπευτικών LLMs.



Σχήμα 2.5: Κατανομή των πηγών δεδομένων προ – εκπαίδευσης αντιπροσωπευτικών LLMs

Οι πηγές του προ – εκπαιδευτικού σώματος μπορούν να κατηγοριοποιηθεί ευρέως σε δύο τύπους: γενικά και εξειδικευμένα δεδομένα κειμένου.

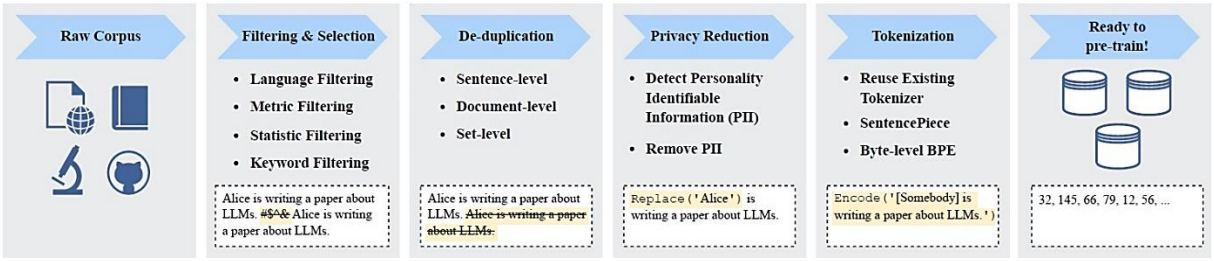
1. Γενικά δεδομένα κειμένου. Όπως μπορούμε να δούμε στο Σχήμα 6, η συντριπτική πλειονότητα των LLMs υιοθετεί δεδομένα προ – εκπαίδευσης γενικού σκοπού, όπως ιστοσελίδες, βιβλία και κείμενο συνομιλίας, τα οποία παρέχουν πηγές πλούσιου κειμένου για μια ποικιλία θεμάτων, βελτιώνοντας τη μοντελοποίηση της γλώσσας και τις ικανότητες γενίκευσης των LLMs. Υπό το πρίσμα των εντυπωσιακών δυνατοτήτων γενίκευσης που παρουσιάζουν τα LLMs, υπάρχουν μελέτες που επεκτείνουν το σώμα τους πριν από την κατάρτιση σε πιο εξειδικευμένα σύνολα δεδομένων, όπως πολύγλωσσα δεδομένα, επιστημονικά δεδομένα και κώδικας, δίνοντας στα LLMs συγκεκριμένες δυνατότητες επίλυσης εργασιών. Στη συνέχεια, συνοψίζουμε εν συντομία τρία σημαντικά είδη γενικών δεδομένων [29].
    - Ιστοσελίδες. Λόγω της εξάπλωσης του Διαδικτύου, έχουν δημιουργηθεί διάφοροι τύποι δεδομένων, τα οποία επιτρέπουν στα LLMs να αποκτήσουν ποικίλες γλωσσικές γνώσεις και να ενισχύσουν τις δυνατότητές τους. Ωστόσο, τα δεδομένα ιστού τείνουν να περιέχουν κείμενο υψηλής ποιότητας, όπως η Wikipedia και κείμενο χαμηλής ποιότητας, όπως ανεπιθύμητα μηνύματα. Επομένως είναι σημαντικό οι ιστοσελίδες να φιλτράρονται για τη βελτίωση της ποιότητας των δεδομένων [29], [33].
    - Κείμενο συνομιλίας. Τα δεδομένα συνομιλίας μπορούν να ενισχύσουν την ικανότητα συνομιλίας των LLMs και ενδεχομένως να βελτιώσουν την απόδοσή τους σε μια σειρά εργασιών ερωτήσεων απαντήσεων. Οι ερευνητές μπορούν να χρησιμοποιήσουν υποσύνολα του σώματος δημόσιας συνομιλίας (π.χ. PushShift.io Reddit corpus) ή να συλλέξουν δεδομένα συνομιλιών από μέσα κοινωνικής δικτύωσης. Δεδομένου ότι τα διαδικτυακά δεδομένα συνομιλίας συχνά περιλαμβάνουν συζητήσεις μεταξύ πολλών συμμετεχόντων, ένας αποτελεσματικός τρόπος επεξεργασίας είναι η μετατροπή μιας συνομιλίας σε δομή δέντρου. Με αυτόν τον τρόπο, το δέντρο συνομιλίας πολλών μερών μπορεί να χωριστεί σε πολλαπλές δευτερεύουσες συνομιλίες, οι οποίες μπορούν να συλλεχθούν στο σώμα προ – εκπαίδευσης.
    - Βιβλία. Σε σύγκριση με άλλα σώματα, τα βιβλία παρέχουν μια σημαντική πηγή επίσημων μακροσκελών κειμένων, τα οποία είναι ωφέλιμα στα LLMs για την απόκτηση γλωσσικών γνώσεων και εξαρτήσεων και τη δημιουργία αφηγηματικών και συνεκτικών κειμένων. Για τη λήψη δεδομένων από βιβλία ανοιχτού κώδικα, οι υπάρχουσες μελέτες συνήθως υιοθετούν τα σύνολα δεδομένων Books3 και Bookcorpus2, τα οποία είναι διαθέσιμα στο σύνολο δεδομένων Pile.
2. Εξειδικευμένα δεδομένα κειμένου. Τα εξειδικευμένα σύνολα δεδομένων είναι χρήσιμα για τη βελτίωση των ειδικών δυνατοτήτων των LLMs. Παρουσιάζονται τρία είδη εξειδικευμένων δεδομένων.

- Πολύγλωσσο κείμενο. Εκτός από το κείμενο στη γλώσσα – στόχο, η ενσωμάτωση ενός πολυγλωσσικού σώματος μπορεί να ενισχύσει τις πολυγλωσσικές ικανότητες κατανόησης και δημιουργίας γλωσσών. Αυτά τα μοντέλα επιδεικνύουν εντυπωσιακή απόδοση σε πολυγλωσσικές εργασίες, όπως μετάφραση, σύνοψη κειμένου προερχόμενη από πολλές γλώσσες και απαντήσεις σε πολυγλωσσικές ερωτήσεις [29].
- Επιστημονικό κείμενο. Η εξερεύνηση της επιστήμης από τους ανθρώπους έχει παρατηρηθεί από την αύξηση των επιστημονικών δημοσιεύσεων. Προκειμένου να βελτιωθεί η κατανόηση της επιστημονικής γνώσης από τα LLMs, είναι χρήσιμο να ενσωματωθεί ένα επιστημονικό σώμα (corpus) για την προ – εκπαίδευση των μοντέλων [29], [34]. Για τη συγκρότηση του επιστημονικού σώματος, συγκεντρώνονται εργασίες από το arXiv, επιστημονικά εγχειρίδια, ιστοσελίδες μαθηματικών και άλλες συναφείς πηγές. Η πολυπλοκότητα των δεδομένων σε επιστημονικά πεδία, όπως τα μαθηματικά σύμβολα και οι αλληλουχίες πρωτεϊνών, απαιτεί εξειδικευμένες τεχνικές δημιουργίας tokens και προ – επεξεργασίας, ώστε να μετατραπούν οι διάφορες μορφές δεδομένων σε μια ενοποιημένη μορφή που μπορεί να επεξεργαστεί από τα γλωσσικά μοντέλα. Μέσω της προ – εκπαίδευσης σε έναν τεράστιο όγκο επιστημονικού κειμένου, τα LLMs μπορούν να επιτύχουν υψηλές επιδόσεις σε επιστημονικές και συλλογιστικές εργασίες [34].
- Κωδικοποίηση. Η σύνθεση προγραμμάτων έχει μελετηθεί ευρέως στην ερευνητική κοινότητα ωστόσο, παραμένει πρόκληση η δημιουργία υψηλής ποιότητας προγραμμάτων. Πρόσφατες μελέτες αποδεικνύουν ότι η εκπαίδευση των LLMs σε ένα τεράστιο σώμα κώδικα μπορεί να οδηγήσει σε ουσιαστική βελτίωση της ποιότητας των σύνθετων προγραμμάτων. Γενικά, δύο τύποι σωμάτων κώδικα χρησιμοποιούνται συνήθως για την προ – εκπαίδευση των LLMs. Η πρώτη πηγή προέρχεται από κοινότητες που απαντούν σε ερωτήσεις προγραμματισμού όπως το Stack Exchange. Η δεύτερη πηγή προέρχεται από δημόσια αποθετήρια λογισμικού όπως το GitHub, όπου συλλέγονται δεδομένα κώδικα, συμπεριλαμβανομένων σχολίων και εγγράφων, για χρήση [29].

Στάδια προ – επεξεργασίας. Η προ – επεξεργασία δεδομένων είναι ένα σημαντικό μέρος της Επεξεργασίας Φυσικής Γλώσσας που καθιστά τα δεδομένα έτοιμα για περαιτέρω ανάλυση και μοντελοποίηση. Τα διάφορα στάδια που εμπλέκονται στην προ – επεξεργασία δεδομένων, σχήμα 2.6 [29], είναι [1]:

- Μετατροπή δεδομένων σε πεζά: Το κείμενο μετατρέπεται σε πεζά που βοηθά στην τυποποίηση του συνόλου δεδομένων.
- Αφαίρεση σημείων στίξης. Όλα τα σημεία στίξης στο κείμενο αφαιρούνται, γιατί συνήθως δεν έχουν νόημα και κάποια στιγμή μπορεί να προσθέσουν θόρυβο.
- Διατήρηση σημαντικών λέξεων: Σε ένα chatbot, η διατήρηση των σημαντικών λέξεων, είναι ζωτικής σημασίας για τη βελτίωση της κατανόησης των ερωτημάτων των χρηστών. Αυτό το βήμα συμβάλλει στην καλύτερη γνώση του περιεχομένου, συμβάλλοντας σε μια πιο αποτελεσματική και εξατομικευμένη αλληλεπίδραση μεταξύ του χρήστη και του chatbot.
- Stop words Removal: Κατάργηση των πιο συχνά εμφανιζόμενων λέξεων σε μια πρόταση, που δεν συμβάλλουν στο νόημα της πρότασης, όπως "α", "αν", "το" και "σε". Κάνει την πρόταση πιο διαχειρίσιμη και βοηθά στην καλύτερη ανάλυση και σε καλύτερα μηχανικά γλωσσικά μοντέλα.
- Tokenization: Το κείμενο μετατρέπεται σε μεμονωμένες μονάδες που ονομάζονται tokens, οι οποίες αναλύονται και επεξεργάζονται από αλγόριθμους. Έτσι παρέχεται μια δομημένη αναπαράσταση δεδομένων κειμένου για υπολογιστική επεξεργασία [1], [31]
- De – duplication: Η διαδικασία της αφαίρεσης διπλοτύπων στοχεύει στην εξάλειψη των διπλότυπων δεδομένων από ένα σύνολο δεδομένων, προκειμένου να αποφευχθούν μεροληψίες κατά την εκπαίδευση των μοντέλων. Τα επαναλαμβανόμενα δεδομένα μπορούν να μειώσουν την ποικιλία των δειγμάτων και να οδηγήσουν σε υπερπροσαρμογή, καθώς το μοντέλο ενδέχεται να μάθει υπερβολικά από συγκεκριμένα παραδείγματα. Έρευνες έχουν δείξει ότι η αφαίρεση διπλοτύπων ενισχύει την ικανότητα των μοντέλων να προσαρμόζονται σε νέα, μη ορατά δεδομένα. Η ανάγκη για τη διαδικασία είναι ιδιαίτερα

έντονη σε μεγάλα σύνολα δεδομένων, καθώς η ύπαρξη διπλότυπων μπορεί να δώσει αδικαιολόγητη βαρύτητα σε ορισμένα πρότυπα. Στον τομέα της Επεξεργασίας Φυσικής Γλώσσας, η ποικιλία και η αντιπροσωπευτικότητα των δεδομένων είναι απαραίτητες για την ανάπτυξη ανθεκτικών γλωσσικών μοντέλων. Οι μέθοδοι αφαίρεσης διπλοτύπων διαφέρουν ανάλογα με το είδος των δεδομένων και τις απαιτήσεις του μοντέλου, συχνά περιλαμβάνοντας τη σύγκριση πλήρων δεδομένων ή συγκεκριμένων χαρακτηριστικών, όπως η επικάλυψη μεταξύ εγγράφων βάσει n-grams [29], [31].



Σχήμα 2.6: Στάδια προ – επεξεργασίας δεδομένων

2.3.2 Προ – εκπαιδευτική Διαδικασία

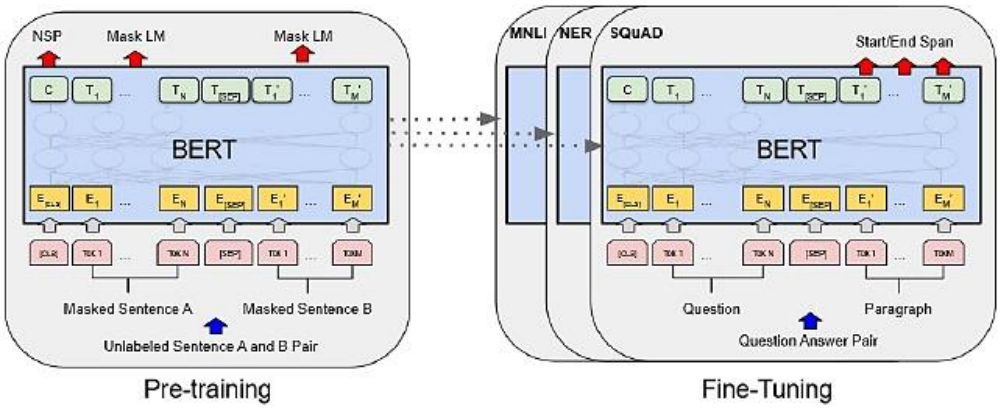
Η προ – εκπαίδευση των LLMs είναι ένα κρίσιμο βήμα με σκοπό να επιτύχουν εξαιρετική απόδοση στις εργασίες που τους ανατίθενται. Περιλαμβάνει την εκπαίδευση του μοντέλου χωρίς επίβλεψη, σε ένα τεράστιο σύνολο διαφορετικών δεδομένων κειμένου, το οποίο περιλαμβάνει όλες τις γλώσσες των χωρών, τις οποίες αντλεί από το Διαδίκτυο. Το Casual Language Modeling (CLM) και το Masked Language Modeling (MLM) είναι δύο δημοφιλείς προσεγγίσεις που χρησιμοποιούνται στα LLMs πριν από την κατάρτισή τους. Στο CLM, ένα LLM εκπαιδεύεται να προβλέπει την επόμενη λέξη σε μια πρόταση, δεδομένου του πλαισίου των προηγούμενων λέξεων. Κατά τη διάρκεια της προ – εκπαίδευσης, το LLM επεξεργάζεται ένα μεγάλο σύνολο δεδομένων κειμένου και προσπαθεί να μάθει τις υποκείμενες γραμματικές, συντακτικές και σημασιολογικές σχέσεις, καθιστώντας το ικανό στη δημιουργία συναφούς και κατάλληλου κειμένου με βάση το περιεχόμενο. Μοντέλα μόνο για αποκωδικοποιητές (Decoder – only), όπως το GPT-3, χρησιμοποιούν αυτήν τη λειτουργία κατά τη φάση της προ – εκπαίδευσης [28].

Σε αντίθεση με το CLM, το MLM είναι μια ελαφρώς διαφορετική προσέγγιση που έγινε δημοφιλής από τον BERT, σχήμα 2. 7 [31]. Κατά τη διάρκεια της προ – εκπαίδευσης, το MLM κρύβει τυχαία ορισμένες λέξεις ή διακριτικά στο εισαγόμενο κείμενο και το μοντέλο έχει την αποστολή να προβλέψει αυτά τα καλυμμένα διακριτικά. Αυτή η προσέγγιση διαφέρει από την CLM, καθώς απαιτεί από το μοντέλο να χρησιμοποιεί το πλαίσιο τόσο πριν όσο και μετά τη συγκαλυμμένη λέξη, για να κάνει ακριβείς προβλέψεις. Μετά την ολοκλήρωση της προ – εκπαίδευσης, αυτά τα μοντέλα μπορούν να βελτιστοποιηθούν (Fine – Tuning), ώστε να ακολουθούν τις οδηγίες ή να εκτελούν συγκεκριμένες εργασίες, όπως η απάντηση σε ερωτήσεις ή η σύνοψη κειμένου [28].

2.3.3 Βελτιστοποίηση (Fine – Tuning)

Για τη βελτιστοποίηση των LLMs σε ένα συγκεκριμένο τομέα, συλλέγεται ένα σύνολο δεδομένων από τον τομέα αυτόν. Αυτό το σύνολο δεδομένων περιέχει παραδείγματα με ετικέτα σχετικά με την εργασία που προορίζεται να εκτελέσει το μοντέλο για τον συγκεκριμένο τομέα. Για παράδειγμα, εάν ο στόχος είναι να χρησιμοποιηθεί το LLM για νομικά κείμενα, συντάσσεται ένα σύνολο δεδομένων νομικών κειμένων με ετικέτα (π.χ. δικαστικές γνωμοδοτήσεις, νομικά άρθρα, συμβάσεις). Η βελτιστοποίηση βοηθάει το LLM να προσαρμοστεί στο «πνεύμα» του τομέα. Το μοντέλο μαθαίνει να αναγνωρίζει την ορολογία, το πλαίσιο και τη χρήση γλώσσας του συγκεκριμένου τομέα. Για παράδειγμα, στην

επεξεργασία νομικών κειμένων το τελειοποιημένο μοντέλο εξοικειώνεται με τη νομική ορολογία, τη νομολογία και το ύφος της "γλώσσας" των συμβάσεων. Ακόμη και μετά τη βελτιστοποίηση, τα LLMs διατηρούν τη γενική τους γλωσσική κατανόηση από τη φάση της προ – εκπαίδευσης. Έτσι, συνδυάζοντας την προ – εκπαιδευμένη γνώση με την προσαρμογή για συγκεκριμένο τομέα, τα LLMs μπορούν να παρέχουν κατάλληλες απαντήσεις στο πλαίσιο αυτού του τομέα, ενώ παράλληλα είναι σε θέση να χειρίζονται γενικές γλωσσικές εργασίες. Αυτό τα επιτρέπει να είναι ευέλικτα και να χειρίζονται αποτελεσματικότερα πολλούς τομείς. Η βελτιστοποίηση είναι ένα κρίσιμο βήμα για την πλήρη αξιοποίηση των δυνατοτήτων των LLMs για NLP σε συγκεκριμένο τομέα [28].



Σχήμα 2.7: Διαδικασία προ-εκπαίδευσης και βελτιστοποίησης του BERT

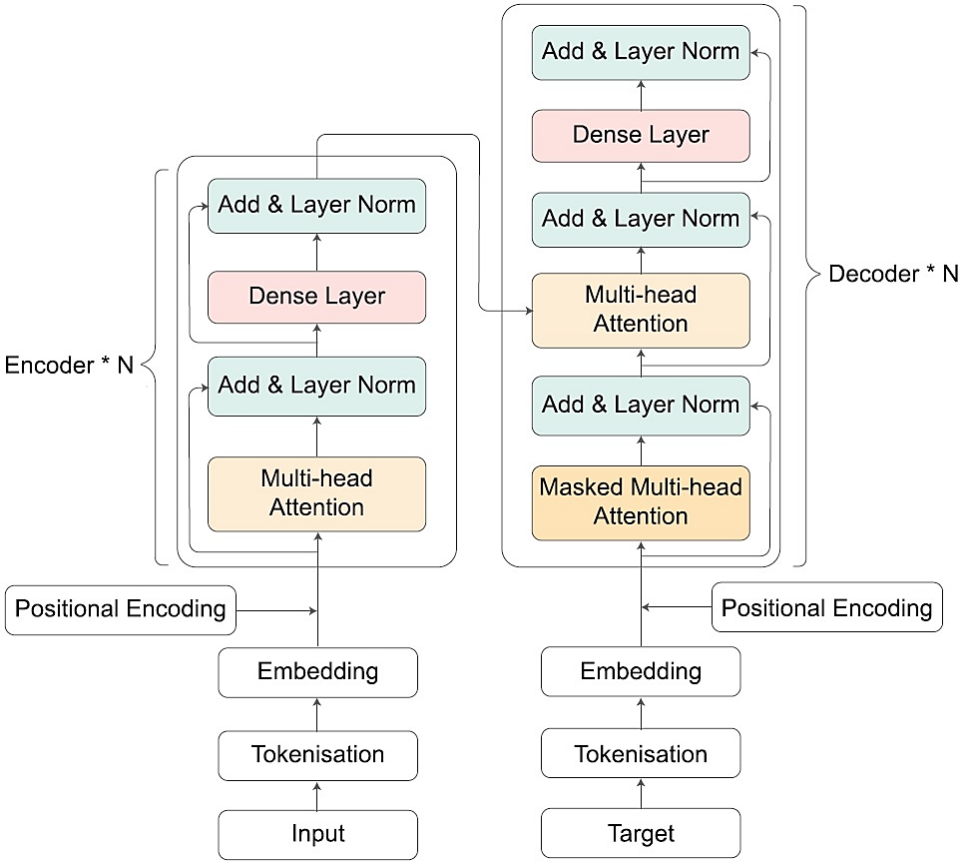
2.3.4 Prompt Engineering (Μηχανική Προτροπών)

Μια άλλη βασική τεχνική που επιταχύνει την κατανόηση του περιεχομένου από τα LLMs είναι η μηχανική των προτροπών, η οποία διατυπώνει στρατηγικά ερωτήματα εισόδου που περιλαμβάνουν περιεχόμενο και οδηγίες. Αυτή η τεχνική είναι απλούστερη από την προ – εκπαίδευση και τη βελτιστοποίηση και επιτρέπει στους χρήστες να αλληλεπιδρούν με το LLM για να ελέγχουν τη ροή δεδομένων των tokens. Οι στρατηγικές προτροπών μειώνουν την ανθρώπινη μεροληψία στα δεδομένα εκπαίδευσης, διευκολύνοντας τους χρήστες να βρίσκουν σχετικά αποτελέσματα. Αυτό οφείλεται κυρίως στην αλληλεπίδραση μεταξύ του χρήστη και του συστήματος. Επίσης, οι προτροπές των χρηστών, έχουν υψηλή πυκνότητα πληροφοριών σε σύγκριση με τα δεδομένα εκπαίδευσης που χρησιμοποιούνται για προ – εκπαίδευση και βελτιστοποίηση. Τέλος, η μηχανική προτροπών μπορεί να προσαρμοστεί, δίνοντας τη δυνατότητα στους χρήστες να μπορούν να επιτύχουν εξαιρετική απόδοση ακρίβειας [32].

2.3.5 Transformers (Μετασχηματιστές)

Η σύγχρονη προσέγγιση των LLMs χρησιμοποιεί κυρίως την αρχιτεκτονική Transformer (Μετασχηματιστής), που εισήχθη από τους Vaswani et al. το 2017. Η βασική καινοτομία αυτής της αρχιτεκτονικής είναι ότι αντικαθιστά τα παραδοσιακά επίπεδα του Recurrent Neural Network (RNN), στα οποία η επεξεργασία κάθε token είναι σειριακή και εξαρτάται από τα προηγούμενα, με μηχανισμούς επίβλεψης, επιτρέποντας στο μοντέλο να επεξεργάζεται τα tokens εισόδου παράλληλα. Οι transformers επιτυγχάνουν αυτήν την επεξεργασία μέσα από τον μηχανισμό multi-head self-attention, που επιτρέπει την παράλληλη εκπαίδευση μοντέλων και έχουν αποδειχθεί εξαιρετικά αποτελεσματικοί στην επεξεργασία διαδοχικών δεδομένων, όπως προτάσεις και παράγραφοι, επιταχύνοντας σημαντικά την εκπαίδευση και την εξαγωγή συμπερασμάτων [28], [32].

Οι transformers αποτελούνται από μια αρχιτεκτονική encoder – decoder (κωδικοποιητή – αποκωδικοποιητή). Ο encoder αντιστοιχίζει ακολουθίες εισόδου σε έναν χώρο υψηλότερων διαστάσεων, ενώ ο decoder παράγει ακολουθίες εξόδου από αυτές τις ενσωματώσεις. Συνήθως, ένα μοντέλο Transformer περιλαμβάνει πολλαπλά επίπεδα από encoders και decoders. Το σχήμα 2.8 παρουσιάζει την αρχιτεκτονική των Transformers. Στην περίπτωση των LLMs που χρησιμοποιούνται για εργασίες, όπως η δημιουργία και η κατανόηση κειμένου, συνήθως χρησιμοποιείται μόνο το τμήμα του decoder (αποκωδικοποιητή). Ο decoder περιλαμβάνει πολλαπλά επίπεδα νευρωνικών δικτύων αυτοελέγχου (self – attention) και τροφοδοσίας (feed – forward). Αυτά τα επίπεδα συνεργάζονται, για να κωδικοποιήσουν το κείμενο εισόδου σε πλούσιες ενσωματώσεις (embeddings) με βάση το περιεχόμενο (context), καταγράφοντας τόσο τοπικές όσο και καθολικές εξαρτήσεις μέσα στο κείμενο. Έτσι επιτυγχάνουν ταχύτερη και αποδοτικότερη, παράλληλη επεξεργασία, διαχειρίζοντας όλα τα μέρη των δεδομένων εισόδου ταυτόχρονα [28], [32].



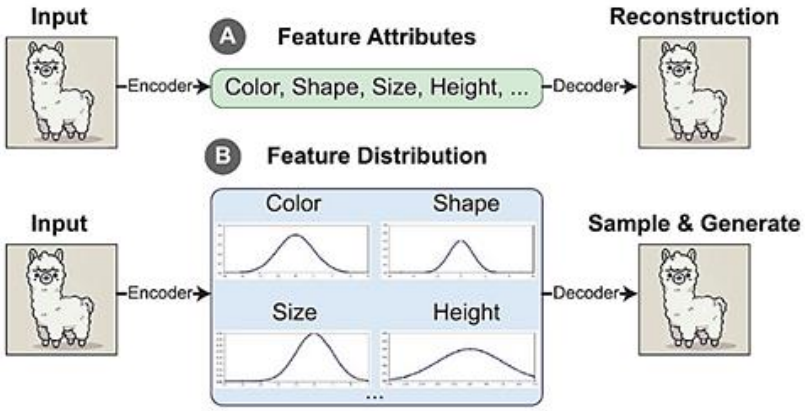
Σχήμα 2.8: Αρχιτεκτονική των Transformers

Για να διατηρηθεί η πληροφορία της ακολουθίας των tokens χωρίς την ανάγκη σειριακής επεξεργασίας, οι transformers χρησιμοποιούν μια τεχνική γνωστή ως positional encoding. Αυτή η μέθοδος επιτρέπει σε κάθε token, όπως μια λέξη σε μια πρόταση, να κωδικοποιεί τη σχετική του θέση μέσα στην ακολουθία. Το positional encoding είναι απαραίτητο, καθώς χωρίς αυτό, το transformer θα αντιλαμβανόταν την πρόταση ως ένα απλό σύνολο λέξεων, χωρίς πληροφορία για τη σειρά τους. Η τεχνική αυτή βασίζεται σε μια μαθηματική φόρμουλα που χρησιμοποιεί συναρτήσεις ημίτονου και συνημίτονου, διασφαλίζοντας ότι κάθε θέση στην ακολουθία λαμβάνει μοναδική κωδικοποίηση. Προσθέτοντας αυτήν την κωδικοποίηση στην ενσωμάτωση του token, το μοντέλο αποκτά πληροφορία σχετικά με τη θέση του μέσα στην ακολουθία. Οι συγκεκριμένες εξισώσεις έχουν σχεδιαστεί, έτσι ώστε να παρέχουν ένα μοναδικό σήμα θέσης για κάθε πιθανή θέση στην είσοδο, επιτρέποντας στο μοντέλο

να ερμηνεύει τη σειρά των λέξεων αποτελεσματικά, ακόμα και με παράλληλη επεξεργασία των δεδομένων [32].

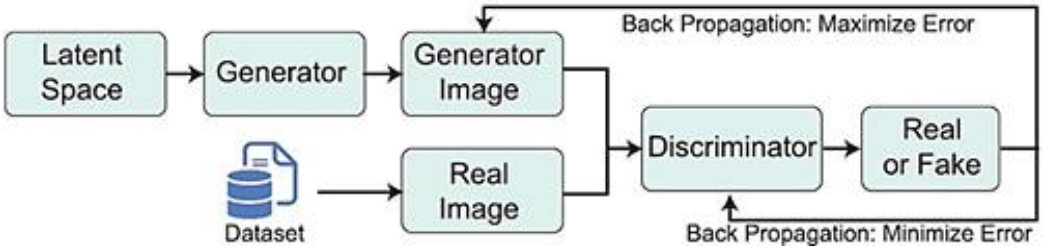
Τα σύγχρονα LLMs που αξιοποιούν τη θεμελιώδη αρχιτεκτονική του Transformer μπορούν να ομαδοποιηθούν ως εξής:

- Auto – encoding (AE - Αυτόματη κωδικοποίηση): Αυτά τα μοντέλα βασίζονται σε κωδικοποιητές. Έχουν σχεδιαστεί και είναι προσαρμοσμένα κυρίως για εργασίες που αφορούν τη φυσική γλώσσα. Μέσω τεχνικών εκπαίδευσης, όπως η bi – directional learning και το masking, διακρίνονται στην κατανόηση περιεχομένου. Παραδείγματα μοντέλων αυτής της κατηγορίας είναι τα ERNIE, ALBERT, BERT και τα παράγωγά του. Παρόλα αυτά, τα μοντέλα που λειτουργούν με αυτόματη κωδικοποίηση, έχουν ορισμένους περιορισμούς, όπως ότι οι είσοδοί τους είναι σταθερού μήκους, ότι η εξάρτηση από το περιεχόμενο μπορεί να είναι εμπόδιο για την παραγωγή κειμένου και ότι η απουσία decoder απαιτεί fine – tuning για την προσαρμογή σε downstream tasks [31 – 32].
- Auto – regressive (AR - Αυτό–παλινδρονούμενα): Με επίκεντρο τον αποκωδικοποιητή, αυτά τα μοντέλα είναι βελτιστοποιημένα για εργασίες παραγωγής. Το τελευταίο διάστημα τα μοντέλα αυτά, όπως το GPT και η σειρά LLaMA, έχουν κερδίσει έδαφος. Η auto - regressive λειτουργία τους βασίζεται στη δημιουργία tokens με βάση τα προηγούμενα, καθιστώντας τα κατάλληλα για παραγωγικές εργασίες. Τα μοντέλα αυτής της κατηγορίας παρέχουν ευελιξία σε μεταβλητά μήκη εισόδου, καθιστώντας τα κατάλληλα για παραγωγή εκτεταμένων δεδομένων, ικανότητα σε εργασίες few – shot ή zero – shot, χωρίς ανάγκη για συγκεκριμένο fine – tuning, αλλά παρουσιάζουν αδυναμία στην αποτύπωση του συνολικού νοήματος του κειμένου μιας και εξάγονται πληροφορίες μόνο από τα προηγούμενα tokens [31].
- Encoder – Decoder (ED Κωδικοποιητής – Αποκωδικοποιητής): Αναφέρονται και ως sequence to sequence (Seq2Seq) μοντέλα. Συνδυάζοντας δομές κωδικοποιητή και αποκωδικοποιητή, αυτά τα LLMs αξιοποιούν τα πλεονεκτήματα των δύο προηγούμενων τύπων με ορισμένους συμβιβασμούς. Είναι εξαιρετικά για εργασίες όπως η σύνοψη, η μετάφραση και η απάντηση ερωτήσεων, όπου η έξοδος εξαρτάται στενά από την είσοδο. Παραδείγματα αποτελούν τα μοντέλα T5, GLM και η σειρά Pangu. Αν και η ενσωμάτωση encoder και decoder επιτρέπει στα Seq2Seq μοντέλα να διαχειρίζονται πολύπλοκες εισόδους, ωστόσο παρουσιάζουν προκλήσεις, όπως ότι η ενσωμάτωση αυξάνει τον αριθμό παραμέτρων, επηρεάζοντας την απόδοση και η εκπαίδευση απαιτεί σημαντικούς υπολογιστικούς πόρους, λόγω του πολύπλοκου συνδυασμού της ακολουθίας εισόδου – εξόδου [31 – 32].
- Variational Auto – encoder (VAE): Τα Variational Auto – encoder [13] είναι εξελιγμένα δημιουργικά μοντέλα που βασίζονται στους παραδοσιακούς auto – encoders και ενσωματώνουν πιθανοτική μοντελοποίηση για τη δημιουργία ενός ευέλικτου χώρου latents. Σε αντίθεση με τους AE που συμπιέζουν δεδομένα εισόδου σε μια στατική αναπαράσταση, οι VAE παράγουν κατανομές πιθανότητας, ορισμένες από μέσους όρους και διακυμάνσεις. Το σχήμα 2.9 [32] δείχνει τη διαφορά στη λειτουργία των AE και VAE. Με τη χρήση πιθανοτικής κωδικοποίησης, οι VAE δημιουργούν δυναμικό χώρο latents, επιτρέποντας όχι μόνο την ανακατασκευή δεδομένων, αλλά και τη δημιουργία νέων, μέσω δειγματοληψίας από τις κατανομές πιθανότητας. Εφαρμόζουν το "reparameterization trick" για τη διατήρηση των gradients κατά τη διάρκεια της δειγματοληψίας και εξισορροπούν την απώλεια ανακατασκευής με την απόκλιση Kullback – Leibler (KL), διασφαλίζοντας ακριβή ανακατασκευή εισόδου και ομαλό χώρο latents. Αυτό καθιστά τα VAE ισχυρά εργαλεία για εφαρμογές όπως δημιουργία εικόνων, εμπλουτισμός δεδομένων και ανίχνευση ανωμαλιών [32].



Σχήμα 2.9: Διαφορά λειτουργίας μεταξύ Auto-encoder και Variational Auto-encoder

- Generative Adversarial Network (GAN): Τα (GANs) αποτελούν μια κατηγορία πλαισίων DL που εισήχθησαν από τους Goodfellow et al. Τα GANs περιλαμβάνουν δύο νευρωνικά δίκτυα, έναν generator και έναν discriminator, τα οποία εκπαιδεύονται ταυτόχρονα μέσω ανταγωνιστικών διαδικασιών. Ο generator στοχεύει στη δημιουργία σύνθετων δεδομένων που μοιάζουν με τα πραγματικά, για παράδειγμα ο generator λαμβάνει ένα διάνυσμα θορύβου (τυχαία δειγματοληψία) και τον χαρτογραφεί σε έναν χώρο δεδομένων, με στόχο να παράγει δείγματα που μοιάζουν με τα δεδομένα εκπαίδευσης, ενώ ο ρόλος του discriminator είναι να διακρίνει τα πραγματικά δεδομένα από τα σύνθετα, για παράδειγμα, λαμβάνει ένα δείγμα (πραγματικό ή σύνθετο) και προβλέπει αν είναι πραγματικό (από το σύνολο εκπαίδευσης) ή ψεύτικο (δημιουργημένο από τον generator), έτσι ώστε να μεγιστοποιήσει την πιθανότητα σωστής ταξινόμησης των δειγμάτων. Με την πρόοδο της εκπαίδευσης, ο generator βελτιώνεται στη δημιουργία ρεαλιστικών δεδομένων, ενώ ο discriminator βελτιώνεται στη διάκριση μεταξύ πραγματικών και ψεύτικων δεδομένων, όπως φαίνεται στο σχήμα 2.10 [31 – 32].



Σχήμα 2.10: Αρχιτεκτονική του Generative Adversarial Network (GAN)

2.3.6 Προκλήσεις των LLMs και τρόποι αντιμετώπισης

Ενώ τα LLMs έχουν πρόσβαση σε τεράστιες ποσότητες δεδομένων κατά τη διάρκεια της προ – εκπαίδευσης, τους λείπει η πραγματική γνώση. Σε αντίθεση με τους ανθρώπους, που διαθέτουν γενικές γνώσεις για τον κόσμο και μπορούν να συλλογιστούν χρησιμοποιώντας αυτή τη γνώση, τα LLMs βασίζονται αποκλειστικά σε μοτίβα που μαθαίνουν από τα δεδομένα εκπαίδευσης. Αυτός ο περιορισμός μπορεί να εμποδίσει την ικανότητά τους να κατανοούν πολύπλοκα σενάρια που απαιτούν εις βάθος γνώση του θέματος. Ειδικά αν τα δεδομένα είναι πολυσύνθετα ή περιλαμβάνουν υπερβολικές επαναλήψεις. Σε τέτοιες περιπτώσεις, το μοντέλο μπορεί να αντιμετωπίσει δυσκολίες στην πλήρη κατανόηση της πρότασης ή μπορεί να παράγει αποτελέσματα με μειωμένη συνοχή. Ακόμα, η ποιότητα των δεδομένων μπορεί να οδηγήσει σε ανακριβείς, παρωχημένες και αναξιόπιστες γνώσεις κατά την εκπαίδευση του μοντέλου [28], [32].

Τα μεγάλα γλωσσικά μοντέλα περιλαμβάνουν δισεκατομμύρια παραμέτρους, γεγονός που επιβαρύνει τη μνήμη των υπολογιστών τόσο κατά την εκπαίδευση όσο και κατά την ανάπτυξή τους. Τρεις κύριες τεχνικές συμπίεσης είναι:

- Το κλάδεμα (Pruning). Το κλάδεμα μειώνει το μέγεθος του μοντέλου αφαιρώντας λιγότερο σημαντικά στοιχεία. Υπάρχουν δύο τύποι: το δομημένο κλάδεμα, που αφαιρεί ολόκληρα τμήματα του μοντέλου, όπως κανάλια, στρώματα και βάρη και το μη δομημένο κλάδεμα, που αφαιρεί μεμονωμένα βάρη ή κόμβους που έχουν τη χαμηλότερη σημασία [29], [32].
- Κβαντοποίηση (Quantization). Η κβαντοποίηση μειώνει τις απαιτήσεις σε υπολογιστικούς πόρους μετατρέποντας παραμέτρους υψηλής ακρίβειας σε χαμηλότερη ακρίβεια. Για παράδειγμα, τα βάρη των 32-bit μετατρέπονται σε 8-bit, βελτιώνοντας την αποθήκευση και την ταχύτητα επεξεργασίας [29], [31].
- Knowledge Distillation (Απόσταξη Γνώσης). Αυτή η τεχνική συμπιέζει τις γνώσεις ενός μεγαλύτερου, πιο σύνθετου μοντέλου ("καθηγητής") σε ένα μικρότερο και αποδοτικότερο μοντέλο ("μαθητής"). Έτσι, το μικρότερο μοντέλο διατηρεί υψηλή απόδοση, καθιστώντας το ιδανικό για συσκευές με περιορισμένους πόρους. Δύο βασικές προσεγγίσεις χρησιμοποιούνται σε αυτήν την τεχνική. Η White-Box και η Black-Box KD [29], [31 – 32].

Λόγω του αυξανόμενου μεγέθους των LLMs, η εκπαίδευση τους πραγματοποιείται πλέον σε συστοιχίες υψηλής απόδοσης αντί για τοπικές συσκευές. Αυτό δημιουργεί προκλήσεις στην κατανομή της υπολογιστικής ισχύος. Οι τρεις κύριες μέθοδοι κατανεμημένου υπολογισμού είναι η data parallelism (παραλληλισμός δεδομένων) που ενισχύει την ταχύτητα εκπαίδευσης μοντέλων, ενώ η tensor parallelism (παραλληλισμός ροών τάσεων) και η pipeline parallelism (παραλληλισμός αγωγών) επιτρέπουν την εκπαίδευση μοντέλων που υπερβαίνουν τη χωρητικότητα μνήμης των συσκευών [28].

Τα LLMs δεν περιορίζονται μόνο στην επεξεργασία φυσικής γλώσσας, αλλά χρειάζεται να χειρίζονται και δεδομένα διαφορετικών μορφών. Η υποστήριξη της πολυτροπικότητας αποτελεί μια σημαντική πρόκληση. Ορισμένες βασικές τεχνικές αντιμετώπισης, οι οποίες βελτιώνουν την ικανότητα των LLMs να συνδυάζουν κείμενο και εικόνες με ακρίβεια, είναι [28]:

- Αντιστοιχία Εικόνας-Κειμένου. Η τεχνική Image – Text Matching (ITM) επιτρέπει στο μοντέλο να προβλέπει τη σχέση μεταξύ μιας εικόνας και του αντίστοιχου κειμένου, βελτιώνοντας την κατανόηση οπτικών πληροφοριών.
- Cross – Modal Contrastive Learning (CMCL). Αυτή η μέθοδος βελτιώνει τη σύνδεση μεταξύ εικόνων και κειμένου, διακρίνοντας άσχετα ζεύγη και ενισχύοντας τη σύνδεση σχετικών ζευγών δεδομένων.
- Cross – Modal Masked Language Matching (CMMLM). Το CMMLM λειτουργεί καλύπτοντας τμήματα των εισόδων (εικόνες ή κείμενο), υποχρεώνοντας το μοντέλο να προβλέψει τις καλυμμένες πληροφορίες κατά την εκπαίδευση.
- Masked Region Modeling (MRM). Το MRM επικεντρώνεται στην κάλυψη οπτικών στοιχείων της εισόδου, επιτρέποντας στο μοντέλο να προβλέψει τις κρυμμένες περιοχές μέσω ταξινόμησης ή παλινδρόμησης.

Τα LLMs μπορούν να εξετάσουν έναν σταθερό αριθμό tokens στην ακολουθία εισαγωγής του ερωτήματος. Τα μακροσκελή και πολύπλοκα έγγραφα αποτελούν μια πρόκληση. Οι μεγάλες προτάσεις μπορεί να περικοπούν ή να χωριστούν σε μικρότερα μέρη, οδηγώντας δυνητικά στην απώλεια του σημαντικού περιεχομένου. Παρά τον περιορισμό αυτό, μπορούν ακόμα να αξιοποιήθουν οι μηχανισμοί ελέγχου για την επίβλεψη των σχετικών τμημάτων της πρότασης και την καταγραφή σημαντικών εξαρτήσεων για την κατανόηση του περιεχομένου [28].

Τα Μεγάλα Γλωσσικά Μοντέλα εκπαιδεύονται σε εκτενείς συλλογές δεδομένων από το Διαδίκτυο, τα οποία συχνά περιέχουν μεροληπτικές πληροφορίες και αντικατοπτρίζουν κοινωνικές προκαταλήψεις.

Ως εκ τούτου, αυτά τα μοντέλα ενδέχεται να απορροφήσουν και να αναπαράγουν τέτοιες προκαταλήψεις στις απαντήσεις τους, συμπεριλαμβανομένων εκείνων που αφορούν το φύλο, τη φυλή, τον πολιτισμό, τη θρησκεία, το επάγγελμα, τις ιδεολογίες και την κοινωνικοοικονομική κατάσταση. Η αντιμετώπιση της μεροληψίας στα LLMs αποτελεί κρίσιμο ηθικό ζήτημα, καθιστώντας απαραίτητη την εστίαση της έρευνας και ανάπτυξης σε τεχνικές που μειώνουν αυτές τις προκαταλήψεις κατά την εκπαίδευση και τη βελτίωση του μοντέλου. Μέθοδοι όπως η επαύξηση δεδομένων, η εκπαίδευση με αντιπαράθεση και η μάθηση με επίγνωση της δικαιοσύνης μπορούν να συμβάλουν στον μετριασμό των στρεβλώσεων. Επιπλέον, η ενσωμάτωση ποικιλόμορφων και αντιπροσωπευτικών δεδομένων μπορεί να οδηγήσει σε πιο δίκαια και αμερόληπτα μοντέλα [28], [32].

Ένα άλλο σημαντικό ζήτημα που εγείρει η χρήση των LLMs είναι η προστασία της ιδιωτικότητας. Είναι απαραίτητο να διασφαλιστεί η ύπαρξη κατάλληλων μέτρων απορρήτου, ώστε να προστατεύονται οι ευαίσθητες πληροφορίες. Τεχνικές όπως τα ανώνυμα δεδομένα, το διαφορικό απόρρητο, το οποίο είναι ένας μαθηματικός τρόπος για την προστασία των ατόμων, όταν τα στοιχεία τους χρησιμοποιούνται σε σύνολα δεδομένων και η ομοσπονδιακή μάθηση, μπορούν να συμβάλουν στη διασφάλιση της εμπιστευτικότητας των δεδομένων. Επιπλέον, η συμμόρφωση με κανονισμούς, όπως ο Γενικός Κανονισμός Προστασίας Δεδομένων (General Data Protection Regulation – GDPR), είναι κρίσιμη για την υπεύθυνη διαχείριση προσωπικών πληροφοριών στις εφαρμογές των LLMs.

Τα LLMs εκπαιδεύονται σε μεγάλα σύνολα δεδομένων που είναι δημόσια διαθέσιμα, γεγονός που μπορεί να περιλαμβάνει περιεχόμενο προστατευμένο από πνευματικά δικαιώματα. Αυτό δημιουργεί ανησυχία σχετικά με πιθανές νομικές παραβιάσεις. Η χρήση υλικού που υπόκειται σε πνευματικά δικαιώματα χωρίς την κατάλληλη άδεια μπορεί να οδηγήσει σε νομικά προβλήματα τόσο για τους προγραμματιστές όσο και για τους χρήστες αυτών των μοντέλων. Οι ερευνητές και οι οργανισμοί που δραστηριοποιούνται στον τομέα των LLMs θα πρέπει να λαμβάνουν υπόψη ζητήματα αδειοδότησης και πνευματικής ιδιοκτησίας. Η αξιοποίηση ανοικτών και δημόσια προσβάσιμων δεδομένων μπορεί να μειώσει αυτούς τους κινδύνους, ενώ η συνεργασία με δημιουργούς περιεχομένου και κατόχους πνευματικών δικαιωμάτων μπορεί να προωθήσει τη νόμιμη και ηθική χρήση προστατευμένου υλικού [28].

Λόγω της εξαιρετικά περίπλοκης δομής τους, τα LLMs, ιδίως εκείνα με δισεκατομμύρια παραμέτρους, συχνά θεωρούνται «μαύρα κουτιά», καθώς ο τρόπος λήψης αποφάσεων τους δεν είναι πάντα διαφανής. Η έλλειψη διαφάνειας καθιστά δύσκολο να κατανοηθεί το σκεπτικό υπό το οποίο καταλήγουν σε συγκεκριμένες απαντήσεις. Τεχνικές όπως η οπτικοποίηση των μηχανισμών προσοχής του μοντέλου, οι χάρτες σημασίας και οι επεξηγήσεις που βασίζονται σε κανόνες, μπορούν να προσφέρουν χρήσιμες πληροφορίες για τη λειτουργία τους.

Η διαμόρφωση ενός ολοκληρωμένου ρυθμιστικού πλαισίου είναι απαραίτητη για την αντιμετώπιση ζητημάτων όπως η προστασία δεδομένων, η μεροληψία, η διαφάνεια και τα πνευματικά δικαιώματα. Η συνεργασία μεταξύ ειδικών τεχνητής νοημοσύνης, νομικών, φορέων χάραξης πολιτικής και άλλων ενδιαφερόμενων μερών είναι κρίσιμη για την ανάπτυξη κατευθυντήριων γραμμών που διασφαλίζουν την υπεύθυνη και ηθική χρήση των LLMs. Δίνοντας προτεραιότητα στη δικαιοσύνη, την ιδιωτικότητα, τη διαφάνεια και την υπεύθυνη ανάπτυξη, μπορούμε να αξιοποιήσουμε τις δυνατότητες των LLMs με τρόπο που προάγει τις αξίες της ισότητας και της δικαιοσύνης [28].

2.3.7 Χρήση AI Πρακτόρων και LLMs στην Εξατομικευμένη Εκπαίδευση

Οι πράκτορες (agents) αποτελούν αυτόνομα συστήματα που έχουν την ικανότητα να αντιλαμβάνονται το περιβάλλον τους, να λαμβάνουν αποφάσεις και να ενεργούν με στόχο την επίτευξη συγκεκριμένων

στόχων. Σε εφαρμογές τεχνητής νοημοσύνης, ο όρος "πράκτορας" χρησιμοποιείται για να περιγράψει συστήματα που λειτουργούν βάσει αλγορίθμων, που συχνά συνδυάζουν μηχανική μάθηση, επεξεργασία φυσικής γλώσσας και λήψη αποφάσεων [35].

Η ενσωμάτωση LLM μοντέλων στην υλοποίηση πρακτόρων έχει επιφέρει μια σημαντική επανάσταση στην αυτοματοποίηση διαλόγου και στην παραγωγή περιεχομένου. Μοντέλα όπως το GPT-3, το ChatGPT και τα νεότερα συστήματα που βασίζονται στην αρχιτεκτονική "Transformer" επιτρέπουν στους πράκτορες να κατανοούν πολύπλοκα ερωτήματα, να παράγουν ακριβείς και φυσικές απαντήσεις, αλλά και να προσαρμόζουν τη συμπεριφορά τους βάσει του συμφραζομένου [28], [35].

Η χρήση τέτοιων μοντέλων, σε συνδυασμό με πλαίσια όπως το LangChain, δίνει τη δυνατότητα για την υλοποίηση διαδραστικών συστημάτων (π.χ., "δασκάλων πράκτορες") που διεξάγουν κουίζ, αξιολογούν απαντήσεις και παρέχουν ανατροφοδότηση σε πραγματικό χρόνο. Αυτοί οι πράκτορες όχι μόνο υποστηρίζουν τη διαδικασία μάθησης αλλά και προάγουν την εξατομίκευση της διδασκαλίας, επιτρέποντας προσαρμοσμένες εκπαιδευτικές εμπειρίες [32], [35].

Η εφαρμογή αυτών των τεχνολογιών έχει ήδη δειχθεί σε πολλαπλά εκπαιδευτικά περιβάλλοντα, όπου οι πράκτορες λειτουργούν ως βοηθοί διδασκαλίας, ενισχύοντας την αλληλεπίδραση μεταξύ εκπαιδευτών και μαθητών. Συγκεκριμένα, για το επίπεδο της πέμπτης τάξης του δημοτικού και για θεματικές ενότητες που αφορούν τη γεωγραφία ή τη φιλοσοφία (π.χ., το έργο του Αριστοτέλη), η χρήση LLM μοντέλων προσφέρει τη δυνατότητα παραγωγής ερωτήσεων και αξιολόγησης απαντήσεων που είναι τόσο εκπαιδευτικές όσο και διασκεδαστικές.

Η υλοποίηση τέτοιων πρακτόρων συνδυάζει τεχνολογίες που έχουν εξελιχθεί τα τελευταία χρόνια, ενσωματώνοντας τόσο κλασικές τεχνικές λήψης αποφάσεων όσο και νεότερες προσεγγίσεις βασισμένες σε βαθιά μάθηση και επεξεργασία φυσικής γλώσσας. Αυτή η πολυδιάστατη προσέγγιση επιτρέπει τη δημιουργία ολοκληρωμένων συστημάτων που μπορούν να προσαρμοστούν σε ποικίλα εκπαιδευτικά περιβάλλοντα και να υποστηρίξουν την εξατομικευμένη μάθηση [35].

2.4 Retrieval – Augmented Generation (RAG)

Η αρχιτεκτονική RAG είναι μια προηγμένη τεχνική για την ανάπτυξη γλωσσικών και λογικών μοντέλων που χρησιμοποιούν αποτελεσματικά εκτεταμένες, εξωτερικές πηγές γνώσης για την παραγωγή ολοκληρωμένων απαντήσεων [12], [36]. Έτσι, η αρχιτεκτονική RAG δίνει τη δυνατότητα σε ένα LLM να κατανοήσει τις εξωτερικές πηγές που μεταβιβάζονται ως αρχεία PDF, αρχεία κειμένου, αρχεία βίντεο κ.λπ. και να χρησιμοποιεί αυτή τη γνώση, για να κάνει συγκεκριμένες εργασίες [37]. Οι σχετικές πληροφορίες ανακτώνται από διανυσματικές βάσεις δεδομένων μέσω σημασιολογικής αντιστοίχισης και στη συνέχεια τροφοδοτούνται στο LLM για τη δημιουργία της απόκρισης, διασφαλίζοντας από κοινού ότι η απάντηση περιέχει ενημερωμένη γνώση. Γενικά, το RAG είναι μια μέθοδος για την προσθήκη νέων δεδομένων στα LLMs, η οποία ενισχύει τη γνώση τους [3], [37] τα επεκτείνει εκτός των δεδομένων εκπαίδευσής τους, γεγονός το οποίο συντελείται ανεξάρτητα ή χωρίς την επανεκπαίδευση των μοντέλων [8].

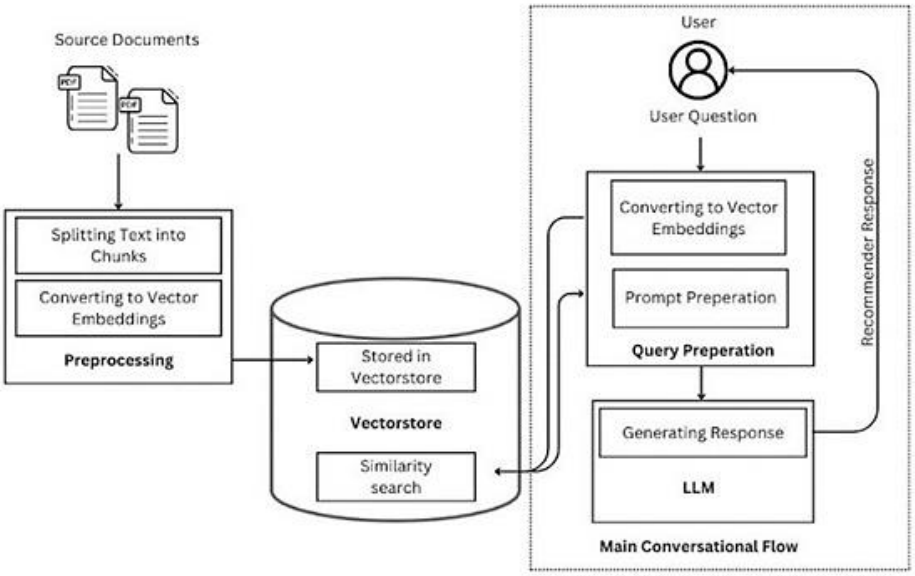
Το RAG είναι μια τεχνική ενσωμάτωσης γνώσης για τη βελτίωση της τεχνητής νοημοσύνης με προσανατολισμό τον άνθρωπο. Χρησιμοποιείται, για να αντιμετωπιστούν διάφορα προβλήματα των LLMs, όπως οι απαρχαιωμένες πληροφορίες, η παροχή ψευδών πληροφοριών, η κακή ποιότητα των απαντήσεων ή η μη σχετικότητα του περιεχομένου με το δεδομένο πρόβλημα [8]. Βρίσκει πολλές εφαρμογές στην Επεξεργασία φυσικής Γλώσσας, συμπεριλαμβάνοντας τη δημιουργία διαλόγου, την απάντηση ερωτήσεων και τη σύνοψη. Ένα LLM που χρησιμοποιεί την αρχιτεκτονική RAG, βελτιώνει την απόδοσή του και αυξάνει τον όγκο των δεδομένων του σε συγκεκριμένους τομείς ή εσωτερικές

πηγές/πληροφορίες, χωρίς αυτά να διατηρούνται στο μοντέλο. Αυτή η συνεργασία οδηγεί σε αποκρίσεις που είναι πλούσιες και ενημερωμένες σε πληροφορίες, καθώς και ευθυγραμμισμένες με το περιεχόμενο [36], [38].

Η λειτουργία της αρχιτεκτονικής RAG βασίζεται στην ανάκτηση εξωτερικών δεδομένων για τη δημιουργία ακριβέστερων απαντήσεων από ένα Μεγάλο Γλωσσικό Μοντέλο (LLM). Αντί το LLM να περιορίζεται μόνο στα δεδομένα που έχει εκπαιδευτεί, η εφαρμογή RAG αναζητά σχετικές πληροφορίες από εξωτερικές πηγές, όπως βάσεις δεδομένων, έγγραφα, API και τις μεταφέρει στο μοντέλο [36].

Για να γίνει αυτό πιο αποτελεσματικά, όλα τα δεδομένα εισόδου μετατρέπονται σε μια κοινή μορφή μέσω τεχνικών ενσωμάτωσης (embeddings), αποθηκεύονται σε μια διανυσματική βάση δεδομένων (VectorDB) και διευκολύνουν την επεξεργασία από το LLM. Στη συνέχεια, το μοντέλο πραγματοποιεί αναζήτηση συνάφειας, συγκρίνοντας τα τρέχοντα και τα αποθηκευμένα δεδομένα, ώστε να ενσωματώσει τις σχετικές πληροφορίες στο γενικότερο πλαίσιο της απόκρισης [36]. Το σχήμα 2.11 προβάλλει την αρχιτεκτονική RAG [14].

Η τεχνική άμεσης μηχανικής (prompt engineering) επιτρέπει στο μοντέλο να επικοινωνεί αποτελεσματικότερα με τα δεδομένα, βελτιώνοντας την ακρίβεια των απαντήσεων του. Ένα σημαντικό βήμα είναι η συνεχής ενημέρωση των εξωτερικών δεδομένων, είτε μέσω περιοδικής τροφοδοσίας νέων πληροφοριών είτε με αυτοματοποιημένες τεχνικές σε πραγματικό χρόνο. Αυτή η διαδικασία διασφαλίζει ότι το LLM βασίζεται πάντα σε επικαιροποιημένα και χρήσιμα δεδομένα, ενισχύοντας την απόδοση και την αξιοπιστία του στη διαχείριση ερωτημάτων [36]. Ακολουθεί μια ανάλυση των παραπάνω βημάτων της αρχιτεκτονικής RAG σχετικά με τα στοιχεία της ευρετηρίασης (Indexing), των διανυσματικών βάσεων δεδομένων (VectorDB), της ανάκτησης (Retrieval) και της παραγωγής απόκρισης (Generation). Έπειτα ακολουθεί μια μαθηματική περιγραφή των παραπάνω βημάτων και το κεφάλαιο ολοκληρώνεται με μια αναφορά των πλεονεκτημάτων της χρήσης του RAG στα LLMs, καθώς και των προκλήσεων που αντιμετωπίζει [37].



Σχήμα 2.11: Αρχιτεκτονική RAG

2.4.1 Ευρετηρίαση (Indexing)

Η ευρετηρίαση περιλαμβάνει τη συλλογή δεδομένων από την πηγή, χρησιμοποιώντας προγράμματα φόρτωσης εγγράφων για την εισαγωγή δεδομένων ως Documents. Ένα Document αποτελείται από

κείμενο και τα αντίστοιχα μεταδεδομένα (metadata) του. Για παράδειγμα, υπάρχουν προγράμματα φόρτωσης που επιτρέπουν τη μεταφόρτωση ενός αρχείου TXT, την ανάκτηση περιεχομένου από ιστοσελίδες ή ακόμα και τη φόρτωση αντιγράφου από βίντεο του YouTube. Στη συνέχεια, τα έγγραφα υποβάλλονται σε διαδικασία διαχωρισμού, όπου ένα μεγάλο κείμενο διασπάται σε μικρότερα, πιο εύχρηστα τμήματα (chunks). Αυτό το στάδιο είναι καίριας σημασίας για την αποτελεσματική οργάνωση και ανάλυση γλωσσικών δεδομένων, διευκολύνοντας την εξαγωγή ουσιωδών πληροφοριών, την αναγνώριση μοτίβων και την εκτέλεση διάφορων εργασιών Επεξεργασίας Φυσικής Γλώσσας (NLP) [37].

Τα τμήματα του κειμένου μετατρέπονται σε αριθμητικές αναπαραστάσεις μέσω μοντέλων ενσωμάτωσης, καθιστώντας τα κατάλληλα για επεξεργασία από αλγόριθμους μηχανικής μάθησης. Αυτές οι ενσωματώσεις (embeddings) χρησιμοποιούνται σε εφαρμογές NLP, όπως η κατανόηση κειμένου, η ανάλυση συναισθημάτων και η μετάφραση. Στο Langchain, τα μοντέλα αυτά δημιουργούν ενσωματώσεις τόσο για ερωτήματα όσο και για έγγραφα.

Όταν ένα ερώτημα ενσωματώνεται, η συμβολοσειρά κειμένου μετατρέπεται σε έναν αριθμητικό πίνακα, όπου κάθε αριθμός αντιστοιχεί σε μία διάσταση του χώρου ενσωμάτωσης. Για την αποθήκευση και την αναζήτηση μη δομημένων δεδομένων, τα διανύσματα ενσωμάτωσης καταχωρούνται και αποθηκεύονται. Κατά την εκτέλεση ενός ερωτήματος, η ενσωματωμένη αναπαράστασή του συγκρίνεται με τα αποθηκευμένα διανύσματα, ώστε να ανακτηθούν εκείνα που είναι πιο συναφή. Η αποθήκευση αυτών των δεδομένων και η αναζήτησή τους πραγματοποιείται μέσω του Vector Store. Το σχήμα 2.12 παρουσιάζει τη διαδικασία της ευρετηρίασης [37].

2.4.2 Embeddings (Ενσωματώσεις)

Οι λέξεις ή το κείμενο πριν χρησιμοποιηθούν από το LLM, πρέπει να μετατραπούν σε μια μορφή που το μοντέλο μπορεί να την επεξεργαστεί αποτελεσματικά. Η διαδικασία μετατροπής δεδομένων ή κειμένου σε έναν συνεχή διανυσματικό χώρο ονομάζεται Embedding. Το αποτέλεσμα αυτής της διαδικασίας είναι τα tokens. Ο αριθμός αυτών, που μπορεί να επεξεργαστεί ένα μοντέλο εξαρτάται από το μέγεθος του token και τους περιορισμούς του μοντέλου. Μετά το tokenization, τα tokens αντιστοιχίζονται ως μια αρχική ενσωμάτωση που δεν είναι τίποτε άλλο από μια προ – εκπαιδευμένη αναπαράσταση που έμαθε το μοντέλο. Γενικά, τα Embeddings χρησιμοποιούνται για διάφορες διαδικασίες επεξεργασίας φυσικής γλώσσας (NLP), αλλά μπορούν επίσης να χρησιμοποιηθούν και για άλλους τύπους αρχείων όπως εικόνες, ήχους και άλλα [8].

2.4.3 Διανυσματικές βάσεις δεδομένων (Vector DB)

Τα tokens πρέπει να είναι μεγάλα, για να τροφοδοτούν απευθείας το LLM, ωστόσο εάν επεξεργαζόμαστε τεράστια δεδομένα, τότε η διαδικασία διαφέρει. Οι ενσωματώσεις «μαθαίνονται» από το μοντέλο κατά τη διάρκεια της εκπαίδευσής του σε μεγάλα σύνολα δεδομένων. Η διαδικασία κατηγοριοποιεί ή τοποθετεί τα διανύσματα, έτσι ώστε οι λέξεις ή τα στοιχεία με παρόμοια σημασία να έχουν παρόμοια διανύσματα. Η αποθήκευση αυτών των διανυσμάτων γίνεται σε διανυσματικές βάσεις δεδομένων (VectorDB). Εκεί αποθηκεύονται οι ενσωματώσεις (embeddings). Οι VectorDB είναι βελτιστοποιημένες για διατήρηση μεγάλου όγκου διανυσμάτων πολλών διαστάσεων. Αυτά τα διανύσματα θα μπορούσαν να περιλαμβάνουν κείμενο, εικόνα κ.λπ. [8], [39].

Τα ενσωματωμένα tokens ευρετηριάζονται (indexed) χρησιμοποιώντας προηγμένες τεχνικές ευρετηρίασης, όπως KD-trees, γραφήματα Hierarchical Navigable Small World (HNSW) ή Inverted File Index (IVF), για να επιτρέψουν την αποτελεσματική και γρήγορη αναζήτηση σε παρόμοια

διανύσματα. Οι διανυσματικές βάσεις δεδομένων έχουν σχεδιαστεί για να χειρίζονται δεδομένα μεγάλης κλίμακας και μπορούν να διαχειριστούν αποτελεσματικά εκατομμύρια ή δισεκατομμύρια διανύσματα, καθιστώντας τες κατάλληλες για χρήση σε διάφορούς τομείς, αλλά και για ακαδημαϊκούς σκοπούς [8], [39].

Κατά τη ρύθμιση του LLM, το ίδιο το μοντέλο λειτουργεί ανεξάρτητα, ενώ η διανυσματική βάση δεδομένων χρησιμεύει ως αποθήκη για τα ενσωματωμένα δεδομένα. Το LLM δεν μπορεί να συμπεριλάβει όλα τα δεδομένα και δεν είναι πρακτικό να επανεκπαιδευτεί σε νέα δεδομένα. Η διανυσματική βάση δεδομένων παίζει καθοριστικό ρόλο, παρέχοντας συνεχώς δεδομένα, διασφαλίζοντας έτσι ότι το LLM έχει πάντα πρόσβαση σε ενημερωμένο περιεχόμενο για επεξεργασία [8].

Ακολουθεί μια επισκόπηση της δομής και των βασικών χαρακτηριστικών της διανυσματικής βάσης ChromaDB, η οποία χρησιμοποιήθηκε στην εργασία και είναι ειδικά σχεδιασμένη για τη διαχείριση και την αναζήτηση διανυσματικών ενσωματώσεων (Vector Embeddings).

2.4.4 ChromaDB

Η αρχιτεκτονική της ChromaDB έχει τα ακόλουθα στοιχεία [4], [40 – 41]:

- Collections (Συλλογές): Το ChromaDB είναι οργανωμένο σε συλλογές, οι οποίες είναι παρόμοιες με τους πίνακες σε μια παραδοσιακή βάση δεδομένων. Κάθε συλλογή μπορεί να αποθηκεύσει ένα σύνολο σχετικών ενσωματώσεων (related embeddings).
- Vectors (Διανύσματα): Μέσα σε κάθε συλλογή, το ChromaDB αποθηκεύει διανύσματα (ενσωματώσεις), τα οποία είναι αριθμητικές αναπαραστάσεις των δεδομένων. Αυτά υποβάλλονται στο ερώτημα. Τα διανύσματα μπορούν να αντιπροσωπεύουν κείμενο, εικόνες ή οποιονδήποτε άλλο τύπο δεδομένων που μπορεί να κωδικοποιηθεί σε διάνυσμα.
- Metadata (Μεταδεδομένα): Παράλληλα με τα διανύσματα, το ChromaDB επιτρέπει την αποθήκευση μεταδεδομένων για κάθε διάνυσμα. Αυτά τα μεταδεδομένα μπορεί να είναι οποιαδήποτε πρόσθετη πληροφορία που σχετίζεται με το διάνυσμα, όπως το αρχικό κείμενο ή άλλα αναγνωριστικά.
- Indexes (Ευρετήρια): Το ChromaDB χρησιμοποιεί διάφορες τεχνικές ευρετηρίασης, για να βελτιστοποιήσει την αποθήκευση και την ανάκτηση διανυσμάτων. Αυτοί οι δείκτες έχουν σχεδιαστεί για να χειρίζονται αποτελεσματικά δεδομένα πολλών διαστάσεων.

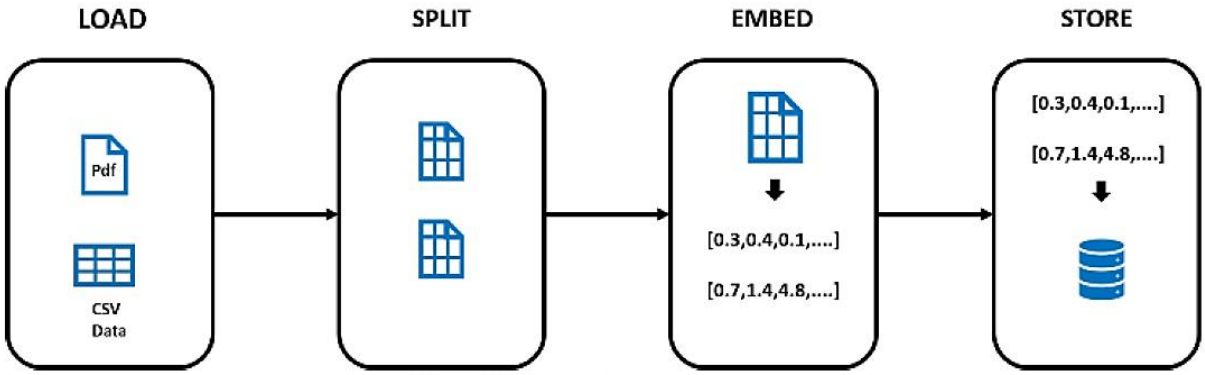
Ορισμένα εξαιρετικά χαρακτηριστικά της ChromaDB είναι [4], [40 – 41]:

- Vector Search (Αναζήτηση διανυσμάτων): Το ChromaDB παρέχει αποτελεσματικές δυνατότητες εύρεσης της ομοιότητας μεταξύ διανυσμάτων. Επιτρέπει τις αναζητήσεις πλησιέστερου γείτονα (nearest neighbor), οι οποίες είναι κρίσιμες για εργασίες, όπως η εύρεση παρόμοιων εγγράφων, εικόνων ή άλλων τύπων δεδομένων.
- Scalability (Επεκτασιμότητα): Το ChromaDB έχει δημιουργηθεί για να χειρίζεται σύνολα δεδομένων μεγάλης κλίμακας. Μπορεί να διαχειριστεί εκατομμύρια διανύσματα, ενώ εξακολουθεί να παρέχει γρήγορες απαντήσεις ερωτημάτων.
- MultiModal Data Support (Υποστήριξη πολλαπλών τύπων δεδομένων): Μπορεί να χειριστεί διαφορετικούς τύπους δεδομένων (π.χ. κείμενο, εικόνες), αποθηκεύοντας τις αντίστοιχες ενσωματώσεις τους στην ίδια ή σε διαφορετικές συλλογές.
- Integration with Machine Learning Models (Ενσωμάτωση με μοντέλα μηχανικής μάθησης): Το ChromaDB έχει σχεδιαστεί, για να λειτουργεί απρόσκοπτα με μοντέλα μηχανικής μάθησης, ιδιαίτερα με εκείνα που δημιουργούν ενσωματώσεις (embeddings), όπως transformers ή συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks).
- API και Libraries: Παρέχει API και βιβλιοθήκες που διευκολύνουν την ενσωμάτωση του ChromaDB στις εφαρμογές, είτε σε Python είτε σε Java ή σε άλλες γλώσσες.

- Metadata Filtering (Φιλτράρισμα μεταδεδομένων): Η ChromaDB δίνει τη δυνατότητα πραγματοποίησης αναζητήσεων, όχι μόνο με βάση την ομοιότητα των διανυσμάτων, αλλά και φιλτράροντας τα μεταδεδομένα που σχετίζονται με τα διανύσματα. Αυτό επιτρέπει πιο σύνθετα ερωτήματα και καλύτερη ακρίβεια στην ανάκτηση.
- Versioning (Εκδόσεις): Το ChromaDB υποστηρίζει τις εκδόσεις στις συλλογές, επιτρέποντας την αποτελεσματική διαχείριση των αλλαγών στα σύνολα δεδομένων με την πάροδο του χρόνου.
- Data Persistence and Durability (Αντοχή και ανθεκτικότητα δεδομένων): Το ChromaDB διασφαλίζει ότι τα δεδομένα σας αποθηκεύονται με ασφάλεια και μπορούν να ανακτηθούν σε περίπτωση αποτυχίας. Υποστηρίζει διάφορους μηχανισμούς, ανάλογα με τις ανάγκες του χρήστη.
- Distributed and Cloud – Ready: Το ChromaDB μπορεί να αναπτυχθεί σε κατανεμημένα περιβάλλοντα και είναι βελτιστοποιημένο για ανάπτυξη cloud, καθιστώντας το κατάλληλο για εφαρμογές μεγάλης κλίμακας και υψηλής διαθεσιμότητας.
- Real-Time Updates (Ενημερώσεις σε πραγματικό χρόνο): Υποστηρίζει ενημερώσεις σε πραγματικό χρόνο για τις αποθηκευμένες ενσωματώσεις και τα μεταδεδομένα, επιτρέποντας τις εφαρμογές να παραμένουν ενημερωμένες με ελάχιστη καθυστέρηση.

Η χρήση του ChromaDB στην Επεξεργασία Φυσικής Γλώσσας βοηθάει στη διεκπεραίωση εργασιών, όπως η ομοιότητα προτάσεων, η ομαδοποίηση κειμένου ή η μοντελοποίηση θεμάτων. Το ChromaDB είναι ιδιαίτερα ισχυρό σε σενάρια όπου απαιτείται η διαχείριση και η αποτελεσματική αναζήτηση μεγάλου όγκου διανυσματικών δεδομένων, καθιστώντας το πολύτιμο εργαλείο για σύγχρονες εφαρμογές τεχνητής νοημοσύνης και μηχανικής μάθησης [4].

Η διαδικασία και τα στοιχεία που αναφέρθηκαν παραπάνω, παρουσιάζονται στο σχήμα 2.12 [37].



Σχήμα 2.12: Διαδικασία ευρετηρίασης των δεδομένων σε διανυσματική βάση

2.4.5 Ανάκτηση (Retrieval)

Κατά τη φάση της ανάκτησης, το μοντέλο χρησιμοποιεί ένα Retriever, για να ανακτήσει το πιο σχετικό περιεχόμενο από τη διανυσματική βάση δεδομένων, με βάση το ερώτημα εισαγωγής από τον χρήστη. Το ερώτημα είναι τύπου string και επιστρέφει μια λίστα από τις πιο σχετικές ενσωματώσεις περιεχομένου (embeddings) που βρίσκονται στη διανυσματική βάση δεδομένων. Η επιλογή αυτών των εγγράφων βασίζεται στην ομοιότητά τους με το ερώτημα εισαγωγής. Η ομοιότητα συνημιτόνου (Cosine similarity) χρησιμοποιείται για τον έλεγχο της ομοιότητας μεταξύ του περιεχομένου και του ερωτήματος. Υπολογίζει το συνημίτονο της γωνίας μεταξύ δύο διανυσμάτων, εξίσωση (2.1).

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{2.1}$$

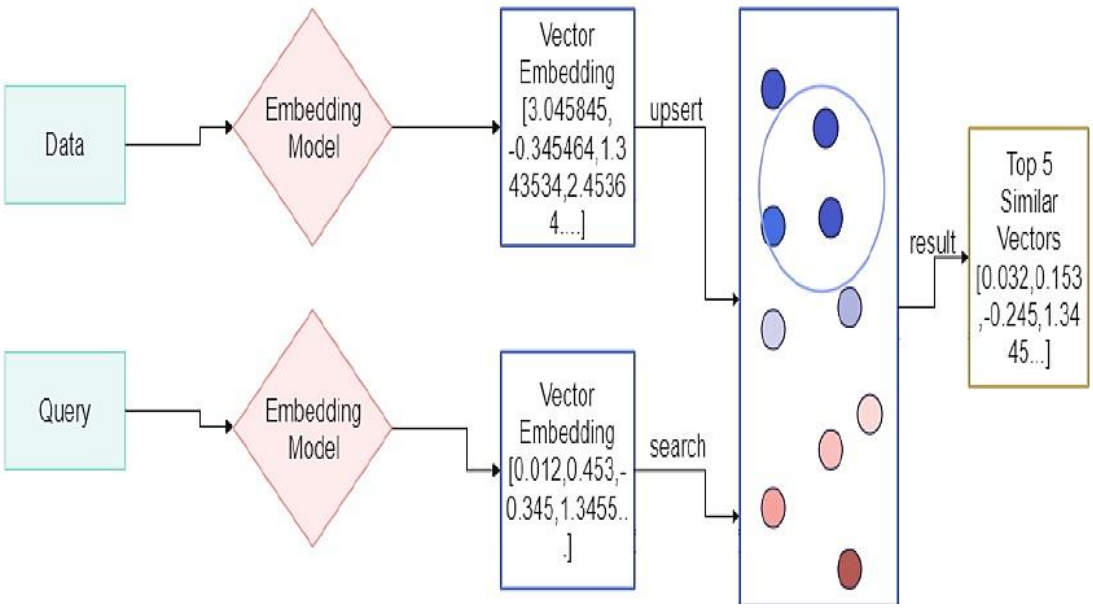
όπου A και B είναι οι αριθμητικές διανυσματικές αναπαραστάσεις των δύο κειμένων.

Όταν ένα ερώτημα δίνεται ως είσοδος, ο retriever υπολογίζει την ομοιότητα συνημιτόνου μεταξύ της ενσωμάτωσης του ερωτήματος και των ενσωματώσεων όλων των εγγράφων στη βάση δεδομένων. Τα έγγραφα με την υψηλότερη ομοιότητα συνημιτόνου θεωρούνται παρόμοια με το ερώτημα και τα κορυφαία n (top n) επιστρέφονται ως έξοδος. Έπειτα το προωθεί ως είσοδο στο LLM, το οποίο δημιουργεί την απάντηση [30], [37]. Το σχήμα 2-13 [8] δείχνει τη διαδικασία της ανάκτησης.

2.4.6 Παραγωγή απόκρισης

Κατά την παραγωγή απόκρισης δημιουργείται ένα πρότυπο προτροπής (prompt) που παρέχει μια τυποποιημένη μορφή για την υποβολή ερωτημάτων στο μέρος της ανάκτησης. Έτσι, εξασφαλίζεται η συνέπεια στον τρόπο διατύπωσης των ερωτημάτων, καθιστώντας ευκολότερη την ανάκτηση σχετικών πληροφοριών από εξωτερικές πηγές. Με τη χρήση ενός προτύπου προτροπής, η ποιότητα των ερωτημάτων βελτιώνεται και αποστέλλεται στον retriever. Το πρότυπο περιλαμβάνει λέξεις-κλειδιά, κομμάτια κειμένου ή δομημένες πληροφορίες που βοηθούν τον retriever να κατανοήσει την πρόθεση του χρήστη και τον καθοδηγούν στην επιλογή κατάλληλων πηγών ή υποσυνόλων δεδομένων για ανάκτηση, έτσι ώστε να ανακτήσει τα πιο ακριβή και σχετικά δεδομένα. Αυτό το πλαίσιο μπορεί να είναι ζωτικής σημασίας για την κατάλληλη ανάκτηση πληροφοριών, ανάλογα με το ερώτημα του χρήστη [37].

Το μοντέλο παραγωγής της απόκρισης, συνήθως ένα LLM που βασίζεται σε μετασχηματιστή (transformer), χρησιμοποιεί τα ανακτημένα έγγραφα για να δημιουργήσει μια συνεκτική και σχετική με το περιεχόμενο (context) απόκριση. Συνδυάζει τις ανακτηθείσες πληροφορίες με τις εξόδους του μοντέλου παραγωγής, για την εξαγωγή μιας κατάλληλης και σημασιολογικά συνεπής απάντησης στον χρήστη σε μορφή string [4], [30].



Σχήμα 2.13: Η διαδικασία της ανάκτησης

2.4.7 Μαθηματική περιγραφή της διαδικασίας

Η όλη διαδικασία μπορεί να περιγραφεί μαθηματικά ως εξής [4]:

1. Tokenization: Το κείμενο T διαχωρίζεται σε ένα σύνολο από tokens {t1, t2, ..., tn}.
2. Embedding: Τα tokens μετατρέπονται σε ενσωματώσεις {e1, e2, ..., en}.
3. Transformer Layers: Οι ενσωματώσεις μετατρέπονται σε αναπαραστάσεις {h1, h2, ..., hn}.
4. Aggregation (Συσσώρευση): Οι αναπαραστάσεις {h1, h2, ..., hn} συνδυάζονται σε έναν τελικό διανύσμα v που αναπαριστά το κείμενο T στον σημασιολογικό χώρο.
5. Εφαρμογή σε πολλαπλά έγγραφα: Αν το σύνολο των εισερχόμενων κειμένων είναι {T1, T2, ..., Tn}, τότε κάθε έγγραφο T_i μετατρέπεται σε διανύσμα v_i, με βάση την εξίσωση 2.2, μέσω του μοντέλου intfloat/multilingual-e5-large:

$$v_i = f(T_i) \tag{2.2}$$

Όπου f είναι η συνάρτηση που αντιστοιχίζει το κείμενο σε διανύσματα. Τα διανύσματα {v1, v2, ..., v_n} αποθηκεύονται στη βάση ChromaDB.

Διαδικασία αναζήτησης:

1. Όταν λαμβάνεται ένα ερώτημα Q μετατρέπεται σε διάνυσμα, με βάση την εξίσωση 2.3:

$$q = f(Q) \tag{2.3}$$

2. Υπολογίζεται η cosine similarity, με βάση την εξίσωση 1, μεταξύ του διανύσματος του ερωτήματος q και κάθε αποθηκευμένου διανύσματος vi.
3. Τα διανύσματα ταξινομούνται κατά φθίνουσα σειρά ομοιότητας. Επιστρέφονται τα κείμενα με την υψηλότερη ομοιότητα (Top-k).
4. Αποθήκευση διανυσμάτων στο ChromaDB:
5. Τα διανύσματα μπορούν να αποθηκευτούν με διάφορους τρόπους:
   - Ως πίνακες (array): [1, 2, 3, 4].
   - Ως συμβολοσειρές (string): '1234'.
   - Ως γραμμές/στήλες μήτρας.
   - Ως JSON: { "vector": [1, 2, 3, 4] }.
   - Ως blobs: Μη κωδικοποιημένα δεδομένα.

Η επιλογή τρόπου αποθήκευσης εξαρτάται από παράγοντες όπως η απόδοση, η ευελιξία και η ανάγκη ανάκτησης δεδομένων.

2.4.8 Πλεονεκτήματα συνεργασία των LLMs με εφαρμογή της αρχιτεκτονικής RAG.

Όταν οι δύο μέθοδοι, LLM και RAG συνδυάζονται, τα συστήματα AI λειτουργούν σε υψηλότερο επίπεδο δυνατοτήτων και αξιοπιστίας. Υπάρχει μια αρμονική συνεργασία μεταξύ των δύο μεθόδων, σε θέματα όπως:

- Η ενσωμάτωση αξιόπιστων πληροφοριών: Το RAG ενισχύει το LLM παρέχοντάς του πρόσβαση σε μια μεγάλη, επιμελημένη βάση γνώσεων, διασφαλίζοντας ότι το σύστημα τεχνητής νοημοσύνης μπορεί να δημιουργήσει απαντήσεις που είναι δημιουργικές και πραγματικές.
- Η επίγνωση περιεχομένου: Το σύστημα ανακτά σχετικές πληροφορίες για το περιεχόμενο και παράγει ένα αποτέλεσμα που ανταποκρίνεται καλύτερα στις απαιτήσεις και τις λεπτομέρειες της εργασίας [8].
- Επεκτασιμότητα και αποτελεσματικότητα: Ο συνδυασμός RAG και LLM έχει τη δυνατότητα χειρισμού πολλών δεδομένων, εκτελεί αποτελεσματικά σύνθετα ερωτήματα και υποστηρίζει τη λήψη αποφάσεων σε πραγματικό χρόνο. Παράγονται πιο ακριβή αποτελέσματα για τον χρήστη, παρέχοντάς τους τις πιο σχετικές, έγκαιρες και ουσιώδεις πληροφορίες [8], [36].

Τα LLMs με αρχιτεκτονική RAG, χρησιμοποιούνται σε διάφορες εφαρμογές, συμπεριλαμβανομένων συστημάτων υποστήριξης πελατών, ακαδημαϊκών συμβούλων, μηχανών αναζήτησης και σε οποιοδήποτε σενάριο όπου η ακρίβεια στην ανάκτηση πληροφοριών με βάση το περιεχόμενο, είναι ζωτικής σημασίας [4]. Ο συνδυασμός των τεχνικών RAG με τις προηγμένες δυνατοτήτες των LLMs, προωθούν μια αρθρωτή, ευέλικτη και προσαρμόσιμη αρχιτεκτονική chatbot. Αυτή η προσέγγιση απλοποιεί την ανάπτυξη, βελτιώνει τη συντηρησιμότητα και τελικά οδηγεί σε πιο ικανά και ενημερωτικά chatbot για την κάλυψη των εξελισσόμενων αναγκών των χρηστών [37].

2.4.9 Προκλήσεις RAG

Παρόλα αυτά, η αρχιτεκτονική RAG διαθέτει πολλαπλά σημεία ελέγχου, τα οποία, αν δεν ρυθμιστούν σωστά, μπορεί να μειώσουν την ακρίβεια και να οδηγήσουν σε άσχετες απαντήσεις από τα chatbots. Επιπλέον, η διαχείριση δικαιωμάτων πρόσβασης σε έγγραφα περιπλέκει τη διαδικασία αναζήτησης και ανάκτησης, απαιτώντας ιδιαίτερη προσοχή για τη διασφάλιση της ασφάλειας και της συνάφειας των δεδομένων. Επίσης, η διαχείριση πολυτροπικού (multimodal) περιεχομένου απαιτεί τη χρήση διαφορετικών retrievers για την επεξεργασία δομημένων, μη δομημένων και ημιδομημένων δεδομένων, όπως παρουσιάσεις, διαγράμματα, βίντεο και ηχητικά αρχεία. Η αντιμετώπιση αυτών των προκλήσεων είναι απαραίτητη για τη διατήρηση της ακρίβειας και της αξιοπιστίας των chatbots [3].

2.5 Επίλογος

Συνοψίζοντας, τα LLMs και η αρχιτεκτονική RAG αποτελούν δύο βασικές τεχνολογίες που αξιοποιούνται για την επεξεργασία και παραγωγή φυσικής γλώσσας. Τα LLMs βασίζονται σε εκτεταμένη συλλογή και προ – επεξεργασία δεδομένων, αξιοποιώντας τεχνικές όπως η βελτιστοποίηση, η μηχανική προτροπών και οι transformers για την επίτευξη υψηλής απόδοσης. Παράλληλα, το RAG συνδυάζει τις δυνατότητες των LLMs με την ανάκτηση πληροφορίας από εξωτερικές βάσεις δεδομένων, χρησιμοποιώντας τεχνικές όπως indexing, embeddings και vector databases, με χαρακτηριστικό παράδειγμα τη χρήση της ChromaDB.

Αν και καινοτόμες και οι δύο προσεγγίσεις συνοδεύονται από προκλήσεις, όπως η ποιότητα των δεδομένων, η αποδοτικότητα της ανάκτησης και η ερμηνευσιμότητα των αποτελεσμάτων. Ωστόσο, μέσα από βελτιστοποιημένες τεχνικές και εξελίξεις στον τομέα, συνεχίζουν να βελτιώνονται, ανοίγοντας νέους ορίζοντες για τη χρήση της τεχνητής νοημοσύνης στη φυσική γλώσσα.

Κεφάλαιο 3ο: Μεθοδολογία – Υλοποίηση

3.1 Εισαγωγή

Στο 3ο κεφάλαιο περιγράφεται η μεθοδολογία που ακολουθήσαμε για την ανάπτυξη του διαλογικού βοηθού και έπειτα περιγράφονται τα στάδια υλοποίησής του, στα οποία αναφέρουμε τα εργαλεία που χρησιμοποιήσαμε για τη δημιουργία του Chatbot.

3.2 Μεθοδολογία

Το chatbot που δημιουργήθηκε σ' αυτήν τη μελέτη, λειτουργεί χρησιμοποιώντας μια συστηματική ροή εργασίας για τη δημιουργία απαντήσεων με βάση τα ερωτήματα των μαθητών. Χρησιμοποιεί την αρχιτεκτονική Retrieval-Augmented-Generation για την ανάκτηση σχετικών πληροφοριών. Η εφαρμογή βασίζεται στη χρήση πέντε ξεχωριστών διανυσματικών βάσεων δεδομένων, καθεμία από τις οποίες εξυπηρετεί έναν συγκεκριμένο σκοπό και συμβάλλει στη βελτίωση της ανάκτησης και παρουσίασης των πληροφοριών. Επιπλέον, έχει υλοποιηθεί ένας μηχανισμός επιλογής κομματιών κειμένου με στόχο την αποτελεσματική ενσωμάτωση των πιο σχετικών πληροφοριών στο τελικό ερώτημα προς το LLM.

Ο σκοπός και τα οφέλη της κάθε βάσης είναι:

- Multimodal Βάση Δεδομένων:

  Σκοπός: Αποθήκευση περιεχομένου που περιλαμβάνει όχι μόνο κείμενο, αλλά και πίνακες και εικόνες.

  Οφέλη: Επιτρέπει την ανάκτηση πλουσιότερων δεδομένων, παρέχοντας μια ολιστική εικόνα του περιεχομένου του βιβλίου, απαραίτητη για ορισμένες εκπαιδευτικές εφαρμογές.

- Βάση Δεδομένων για Προτάσεις:

  Σκοπός: Διαχωρισμός του κειμένου σε αυτόνομες, κατανοητές προτάσεις.

  Οφέλη: Ενισχύει την ακρίβεια στην ανάκτηση πληροφοριών, καθώς οι προτάσεις αποτελούν μικρές μονάδες πληροφορίας που διατηρούν το συνολικό νόημα.

- Βάση Δεδομένων Ερωτοαπαντήσεων:

  Σκοπός: Αυτόματη εξαγωγή ερωτήσεων και των αντίστοιχων απαντήσεων από το κείμενο μέσω ενός προσαρμοσμένου prompt.

  Οφέλη: Δημιουργεί ένα δομημένο σύνολο δεδομένων που μπορεί να υποστηρίξει διαδραστικά quiz και να προσφέρει στοχευμένες απαντήσεις, ενισχύοντας την εμπειρία μάθησης.

- Βάση Δεδομένων Ομαδοποίησης Μεγαλύτερου Κειμένου:

  Σκοπός: Ομαδοποίηση μεγαλύτερων τμημάτων του κειμένου, έτσι ώστε να αποθηκεύονται κομμάτια με πλούσιο περιεχόμενο που εξυπηρετούν την ανάκτηση σύνθετων απαντήσεων.

  Οφέλη: Επιτρέπει την αντιμετώπιση πιο πολύπλοκων ερωτήσεων, καθώς τα ανακτημένα τμήματα περιέχουν επαρκείς πληροφορίες για λεπτομερείς και εμπεριστατωμένες απαντήσεις.

- Βάση Δεδομένων με Εξαγωγή Σημαντικότερων Λέξεων:

  Σκοπός: Διενέργεια semantic chunking σε μεγαλύτερα κομμάτια του κειμένου, εξάγοντας τις 10 πιο σημαντικές λέξεις από κάθε κεφάλαιο και αποθηκεύοντάς τες ως metadata μαζί με το όνομα του κεφαλαίου.

Οφέλη: Η παρουσίαση συμπυκνωμένων πληροφοριών βοηθά στην γρήγορη αναγνώριση του βασικού περιεχομένου κάθε κεφαλαίου, διευκολύνοντας την αναζήτηση σε πιο σύνθετα ερωτήματα.

Για να ενισχυθεί η αποτελεσματικότητα της ανάκτησης πληροφοριών και να διασφαλιστεί ότι το τελικό "input" προς το LLM δεν υπερβαίνει το όριο των tokens, έχει υλοποιηθεί ο ακόλουθος μηχανισμός επιλογής:

- Αρχική Ανάκτηση: Από κάθε μία από τις πέντε βάσεις δεδομένων, για το δεδομένο ερώτημα του χρήστη ανακτώνται τα 10 κομμάτια με την υψηλότερη σημασιολογική ομοιότητα.
- Ενσωμάτωση Κειμένων: Ο μηχανισμός επιλέγει αρχικά το πρώτο κομμάτι από κάθε βάση και τα ενσωματώνει στο κείμενο που θα σταλεί στο μοντέλο. Αν ο συνδυασμός των επιλεγμένων κομματιών δεν υπερβαίνει το προκαθορισμένο όριο των tokens, τότε συνεχίζει επιλέγοντας το δεύτερο κομμάτι από κάθε βάση, και ούτω καθεξής.
- Ρυθμιζόμενη Δυναμική: Ο μηχανισμός μπορεί να ρυθμιστεί ώστε να λαμβάνει υπόψη του μία ή περισσότερες βάσεις δεδομένων, ανάλογα με το επιθυμητό αποτέλεσμα και τις απαιτήσεις της ταχύτητας απόκρισης, καθώς και τους περιορισμούς στα tokens, ιδιαίτερα λόγω των ορίων που επιβάλλονται από το Groq API.
- Αποτέλεσμα: Η τελική επιλογή συνδυάζει τις πιο σχετικές πληροφορίες από κάθε βάση, εξασφαλίζοντας μια πλήρη και ακριβή απάντηση στο ερώτημα του χρήστη, χωρίς να παραβιάζονται οι τεχνικοί περιορισμοί του συστήματος.

Γενικά, σε όλες τι περιπτώσεις για την προ – επεξεργασία και τον διαχωρισμό των δεδομένων, ακολουθούνται τα εξής βήματα:

- Το αρχικό PDF του βιβλίου μετατρέπεται σε κείμενο με τη βοήθεια εργαλείων.
- Το κείμενο οργανώνεται ανά κεφάλαιο και υποβάλλεται σε επεξεργασία για την αφαίρεση ειδικών χαρακτήρων και την ομαλοποίησή του.
- Ο διαχωρισμός σε τμήματα (chunks) γίνεται με τεχνικές οι οποίες εξασφαλίζουν ότι το περιεχόμενο διατηρεί το νόημά του, μέσω κατάλληλων οριοθετήσεων και επικάλυψης.

3.2.1 Βάση Multimodal

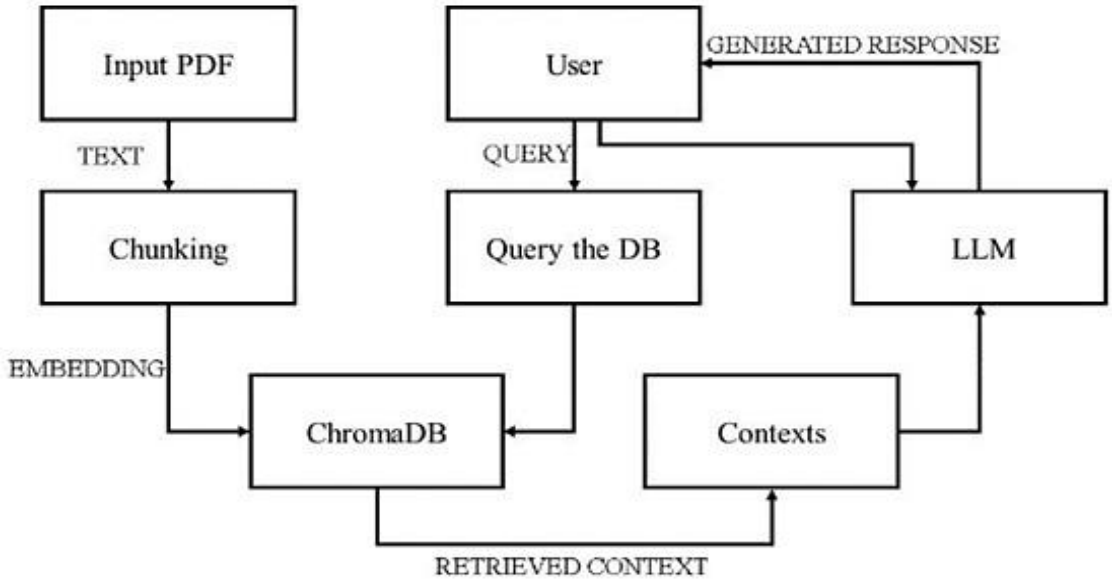
Η ροή εργασίας ξεκινά με το σχολικό βιβλίο του μαθήματος της Γεωγραφίας της Ε' τάξης του δημοτικού σχολείου, το οποίο αποτελεί την πηγή των δεδομένων, περιέχοντας όλες τις ουσιαστικές και σχετικές πληροφορίες που χρειάζεται το chatbot. Το βιβλίο είναι σε μορφή αρχείου PDF και περιλαμβάνει κείμενο, πίνακες και εικόνες. Η ψηφιακή μορφή του βιβλίου της Γεωγραφίας της Ε' Δημοτικού που χρησιμοποιήθηκε, όπως και όλα τα σχολικά βιβλία της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, βρίσκεται στην εκπαιδευτική ιστοσελίδα http://ebooks.edu.gr/ebooks/, η οποία τελεί υπό την αιγίδα του ΥΠΑΙΘΑ. Η πηγές συλλέχθηκαν με σκοπό να εξοπλίσουν τον διαλογικό βοηθό (chatbot) με σημαντικές πληροφορίες που αφορούν την εκπαιδευτική και μαθησιακή διαδικασία των μαθητών. Με τις πληροφορίες αυτές το chatbot μπορεί να αναγνωρίσει και να καταλάβει διάφορες ερωτήσεις και εκπαιδευτικά σενάρια που αφορούν τα μαθήματα. Επιπλέον, στην εργασία περιλαμβάνονται και πηγές δεδομένων σε μορφή TXT και CSV, οι οποίες εξάγονται σε αυτές τις μορφές από την επεξεργασία του PDF και ανατροφοδοτούν το μοντέλο.

Μετά τη φόρτωση του PDF στο πρόγραμμα, διαβάζονται οι 10 πρώτες σελίδες του με σκοπό την εύρεση του πίνακα περιεχομένων του βιβλίου. Έτσι χωρίζουμε το PDF σε PDFs ανά κεφάλαιο με σκοπό να εργαστούμε πιο ευέλικτα και αποτελεσματικά. Μετά τη δημιουργία των επιμέρους PDFs, δημιουργούνται ερωτήσεις με την αντίστοιχη σωστή απάντηση από κάθε κεφάλαιο. Επίσης, δημιουργούνται και λάθος απαντήσεις. Οι ερωταπαντήσεις εξάγονται σε αρχείο μορφής CSV και χρησιμοποιούνται στην επιλογή "Κουίζ" της διεπαφής του chatbot.

Έπειτα, από κάθε αρχείο pdf, που περιέχει το αντίστοιχο κεφάλαιο, αντλούνται το κείμενο, οι πίνακες και οι εικόνες του κεφαλαίου. Το κείμενο ενώνεται σε μια μεταβλητή, έτσι ώστε να επεξεργαστεί ενιαία. Ομοίως και οι πίνακες, οι οποίοι θα προωθηθούν στο LLM, για να εξαχθεί η περιγραφή του καθενός. Οι εικόνες αποθηκεύονται σε φακέλους. Κάθε φάκελος αντιστοιχεί σε ένα κεφάλαιο. Αυτοί οι φάκελοι των εικόνων προωθούνται σε ένα vision μοντέλο LLM, το οποίο κάνει περιγραφή την κάθε εικόνα. Οι περιγραφές μεταφράζονται στα ελληνικά. Τα αρχεία του κειμένου, των περιγραφών των πινάκων και των εικόνων αποθηκεύονται σε μορφή TXT, έτσι ώστε να μπορούν να φορτωθούν και να μην εκτελείται κάθε φορά όλη η διαδικασία από την αρχή.

Πλέον όλα τα δεδομένα είναι σε μορφή κειμένου. Στη συνέχεια τα εξαγόμενα κείμενα χωρίζονται σε μικρότερα κομμάτια (Chunks). Αυτή η διαδικασία του διαχωρισμού είναι απαραίτητη, λόγω του ότι όταν τα πιο σχετικά κομμάτια ανακτηθούν αργότερα, θα περάσουν ως μέρος της προτροπής στο LLM, οπότε αν είναι πολύ μεγάλα τότε η τελική προτροπή (prompt) κατά τη διάρκεια του σταδίου δημιουργίας θα έχει πάρα πολλά κομμάτια δεδομένων (tokens), για να τα επεξεργαστεί το LLM. Κάθε κομμάτι περνά μέσα από έναν αλγόριθμο ενσωμάτωσης διανυσμάτων για τη μετατροπή του απλού κειμένου σε διανυσματικές αναπαραστάσεις υψηλών διαστάσεων (embeddings). Στη συνέχεια, αυτά τα ενσωματωμένα διανυσματικά κομμάτια αποθηκεύονται στη διανυσματική βάση δεδομένων ChromaDB. Αυτό το βήμα πρέπει να πραγματοποιηθεί πριν ο χρήστης κάνει μια ερώτηση, επειδή ορίζει το έγγραφο προέλευσης ως περιεχόμενο. Η μετατροπή των κειμένων σε διανύσματα υψηλών διαστάσεων γίνεται με τη βοήθεια ενός transformer (μετασχηματιστή). Η χρήση διανυσματικής βάσης δεδομένων (VectorDB), διευκολύνει τους υπολογισμούς ομοιότητας μεταξύ διανυσμάτων, καθώς βελτιστοποιεί και αποθηκεύει αυτά τα διανύσματα με αποδοτικό τρόπο. Μετατρέποντας τα κείμενα σε αριθμητικά διανύσματα, τα δεδομένα γίνονται πιο εύκολα αξιοποιήσιμα. Αυτή η μετατροπή επιτρέπει στο μοντέλο να κατανοήσει το σημασιολογικό περιεχόμενο του κειμένου, βελτιώνοντας έτσι την ανάκτηση σχετικών πληροφοριών από τη βάση δεδομένων κατά την απάντηση. Αυτή η διαδικασία γίνεται μόνο μία φορά.

Όταν ένας χρήστης υποβάλλει ένα ερώτημα μέσω της διεπαφής χρήστη, το μοντέλο το μετατρέπει και αυτό σε διανυσματική αναπαράσταση και το συγκρίνει με τα ευρετηριασμένα διανύσματα. Μέσω της διανυσματικής αναζήτησης, εντοπίζονται τα πιο σχετικά τμήματα, χρησιμοποιώντας τη μέτρηση cosine similarity (ομοιότητα συνημιτόνου), εξασφαλίζοντας ότι οι απαντήσεις βασίζονται στο περιεχόμενο του ερωτήματος. Η ερώτηση χρήστη και τα ανακτημένα κομμάτια μετατρέπονται σε απλό κείμενο και στη συνέχεια συνδέονται σε ένα προκαθορισμένο πρότυπο προτροπής συνθέτοντας μια προτροπή, η οποία αποστέλλεται στο LLM. Το πρότυπο προτροπής χρησιμοποιεί μηχανική προτροπών που βασίζεται σε κανόνες, για να διασφαλίσει ότι οι απαντήσεις του LLM είναι στη μορφή, τη δομή, τις λεπτομέρειες, τις καθορισμένες συμπεριφορές και έχουν τη συνέπεια με αυτό που περιμένει ο μαθητής, δηλαδή ο τελικός χρήστης. Αυτή η περιεκτική εισαγωγή επιτρέπει στο LLM να αναλύει και να συνθέτει αποτελεσματικά μια απάντηση που είναι σχετική με το ερώτημα του χρήστη και κατάλληλη με το περιεχόμενο. Μόλις δημιουργηθεί η απάντηση, παρουσιάζεται σε μορφή απλού κειμένου και εμφανίζεται στον χρήστη μέσω μιας φιλικής διεπαφής, που έχει σχεδιαστεί για να διασφαλίζει τη βέλτιστη αναγνωσιμότητα και προσβασιμότητα. Το σχήμα 3.1 [30] αναπαριστά τα βασικά βήματα της μεθοδολογίας.



Σχήμα 3.1: Αναπαράσταση των βασικών τμημάτων της μεθοδολογίας

3.2.2 Διανυσματική Βάση Δεδομένων με Διάσπαση Κειμένου και Δημιουργία Σημασιολογικών Ενσωματώσεων (embeddings)

Η προσέγγιση αυτή οργανώνει και διαχειρίζεται τα δεδομένα για γρήγορη ανάκτηση σχετικών πληροφοριών μέσω σημασιολογικών ομοιοτήτων, διευκολύνοντας εφαρμογές, όπως chatbots και διαδραστικά quiz.

Η διαδικασία ξεκινά με την εξαγωγή του κειμένου από το αρχικό PDF. Στη συνέχεια, το κείμενο επεξεργάζεται για την αφαίρεση ειδικών χαρακτήρων και την ομαλοποίηση του περιεχομένου, ώστε να είναι κατάλληλο για περαιτέρω ανάλυση. Ακολουθεί η διάσπαση του κειμένου σε μικρότερα τμήματα (chunks), διατηρώντας τη συνοχή του νοήματος. Αυτό διασφαλίζει ότι τα δεδομένα που θα αναλυθούν είναι σε κατάλληλο μέγεθος, διευκολύνοντας τη σωστή επεξεργασία τους.

Με τη χρήση ενός προηγμένου LLM, το σύστημα μετατρέπει τα διαχωρισμένα τμήματα σε σαφείς, αυτόνομες προτάσεις. Ένα προσαρμοσμένο prompt καθοδηγεί τη διαδικασία, εξασφαλίζοντας ότι το κείμενο αναδιαμορφώνεται σε απλές, κατανοητές προτάσεις, οι οποίες διατηρούν το αρχικό ύφος και περιλαμβάνουν πλήρεις πληροφορίες. Το αποτέλεσμα αποθηκεύεται ως λίστα προτάσεων, που αποτελεί τη βάση δεδομένων του συστήματος.

Για να καταστεί δυνατή η αναζήτηση προτάσεων με βάση το περιεχόμενο, και όχι απλώς με λεξιλογική σύγκριση, κάθε πρόταση μετατρέπεται σε διανυσματική αναπαράσταση (embedding). Αυτή η μετατροπή επιτρέπει την αριθμητική απεικόνιση της σημασίας του κειμένου, διευκολύνοντας την εύρεση ομοιοτήτων μεταξύ διαφορετικών προτάσεων. Τέλος, οι διανυσματικές αναπαραστάσεις αποθηκεύονται στη βάση δεδομένων Chroma, η οποία λειτουργεί ως μηχανισμός ανάκτησης (retrieval). Έτσι, το σύστημα μπορεί να εντοπίζει γρήγορα και αποτελεσματικά τις πιο σχετικές πληροφορίες, βασισμένο στη σημασιολογική ομοιότητα με τα ερωτήματα των χρηστών.

3.2.3 Διανυσματική Βάση Δεδομένων με Εξαγωγή Ερωτοαπαντήσεων μέσω Προσαρμοσμένου Prompt

Σε αντίθεση με το προηγούμενο σύστημα που εστίαζε στη διάσπαση κειμένου και στη δημιουργία σημασιολογικών ενσωματώσεων, το νέο σύστημα επικεντρώνεται στην αυτόματη παραγωγή ερωτήσεων και απαντήσεων.

Η διαδικασία ξεκινά με την εξαγωγή του κειμένου από το PDF, το οποίο συγκεντρώνεται σε μια ενιαία συμβολοσειρά και στη συνέχεια διασπάται σε τμήματα (chunks). Αυτή η διάσπαση διασφαλίζει ότι το περιεχόμενο καλύπτεται επαρκώς, επιτρέποντας την εξαγωγή λεπτομερών ερωτοαπαντήσεων.

Ο πυρήνας του συστήματος είναι το προσαρμοσμένο prompt, το οποίο καθοδηγεί το LLM να αναλύσει το κείμενο και να δημιουργήσει όσο το δυνατόν περισσότερες ερωτήσεις με τις αντίστοιχες απαντήσεις. Το prompt είναι διαμορφωμένο ώστε να εστιάζει σε σημαντικές πληροφορίες – όπως ονόματα, ημερομηνίες και γεγονότα – αλλά και σε περιεκτικές ερωτήσεις που συνοψίζουν τις βασικές ιδέες του κειμένου.

Αφού παραχθούν οι ερωτοαπαντήσεις, το σύστημα τις μετατρέπει σε διανυσματικές αναπαραστάσεις, αποτυπώνοντας τη σημασιολογική πληροφορία του περιεχομένου. Οι αναπαραστάσεις αυτές αποθηκεύονται στη βάση δεδομένων Chroma. Αυτή η νέα βάση δεδομένων διαφέρει από την προηγούμενη, καθώς επικεντρώνεται στις ερωτήσεις και απαντήσεις αντί για μεμονωμένες προτάσεις. Επιπλέον, η χρήση της επεκτείνεται πέρα από την ανάκτηση πληροφοριών, αφού λειτουργεί και ως βάση δεδομένων για διαδραστικά quiz, διευκολύνοντας την παροχή εκπαιδευτικού υλικού.

Αυτή η προσέγγιση βελτιώνει τη μάθηση, επιτρέποντας στοχευμένες και δομημένες πληροφορίες, ενώ η διανυσματική βάση δεδομένων με ερωτοαπαντήσεις συμβάλλει στην ανάπτυξη έξυπνων εκπαιδευτικών εφαρμογών. Επιπλέον, ενισχύει την αποδοτικότητα των LLMs με χαμηλότερη υπολογιστική ισχύ, καθώς μικρότερα μοντέλα μπορούν να αντλούν συμπυκνωμένες πληροφορίες από τη βάση, μειώνοντας το κόστος και επιταχύνοντας τις αποκρίσεις, καθιστώντας τα κατάλληλα για περιβάλλοντα με περιορισμένους πόρους.

3.2.4 Ομαδοποίηση Εκτενέστερου Κειμένου για Βελτιστοποιημένη Ανάκτηση σύνθετων ερωτήσεων

Σε αυτό το στάδιο, υλοποιείται μια διανυσματική βάση δεδομένων που διαμορφώνεται από μικρότερα κομμάτια που εξάγονται από τα κεφάλαια του βιβλίου. Σε αντίθεση με τις δύο προηγούμενες βάσεις, όπου το κείμενο διασπάστηκε σε πιο σύντομες, απομονωμένες προτάσεις ή ερωτοαπαντήσεις, η παρούσα προσέγγιση επικεντρώνεται στην ομαδοποίηση μεγαλύτερων τμημάτων του κειμένου.

Η διαδικασία ξεκινά με την ανάγνωση του περιεχομένου από αρχεία κειμένου, τα οποία προέρχονται από τα κεφάλαια που έχουν εξαχθεί από το PDF. Ο διαχωρισμός του κειμένου γίνεται με καθορισμένους χαρακτήρες ως οριοθέτες, εξασφαλίζοντας ότι κάθε τμήμα είναι περιεκτικό και νοηματικά ολοκληρωμένο.

Στη συνέχεια, μέσω ενός προσαρμοσμένου prompt, το σύστημα καθοδηγεί το LLM στη διάσπαση του κειμένου σε νοηματικές ενότητες. Το prompt περιλαμβάνει οδηγίες για:

- Τη διαμόρφωση των αποσπασμάτων, ώστε να διατηρούν το πλήρες νόημά τους.
- Την αναφορά του αρχικού τίτλου του κεφαλαίου, καθώς και τη δημιουργία ενός νέου συνοπτικού τίτλου, που περιγράφει περιεκτικά το περιεχόμενο του κάθε αποσπάσματος.
- Τη διατήρηση όλων των σημαντικών λεπτομερειών, αποτρέποντας την απώλεια κρίσιμων πληροφοριών.

Αφού δημιουργηθούν τα αποσπάσματα, υποβάλλονται σε περαιτέρω επεξεργασία ώστε να προκύψουν οι τελικές διανυσματικές αναπαραστάσεις (embeddings).

Η κύρια διαφορά αυτής της προσέγγισης, σε σύγκριση με τις προηγούμενες βάσεις δεδομένων, είναι ότι η ομαδοποίηση μεγαλύτερων τμημάτων κειμένου επιτρέπει την αποτύπωση πιο πλούσιου και ολοκληρωμένου περιεχομένου. Αυτό σημαίνει ότι, όταν το σύστημα καλείται να απαντήσει σε σύνθετες ερωτήσεις ή να παρέχει λεπτομερείς αναλύσεις, τα ανακτημένα αποσπάσματα περιλαμβάνουν επαρκείς πληροφορίες. Ως αποτέλεσμα, το σύστημα μπορεί να παρέχει ακριβέστερες και πιο τεκμηριωμένες απαντήσεις, μειώνοντας την ανάγκη επανεπεξεργασίας του κειμένου σε πραγματικό χρόνο. Παράλληλα, βελτιώνει την απόδοση των LLMs μικρότερης κλίμακας, καθιστώντας τα πιο αποδοτικά και λειτουργικά.

3.2.5 Ομαδοποίηση μεγάλων νοηματικών κειμένων και εξαγωγή 10 σημαντικότερων λέξεων

Στην περίπτωση αυτή, η διανυσματική βάση δεδομένων εφαρμόζει semantic chunking σε εκτενέστερα τμήματα του εγγράφου. Για κάθε κεφάλαιο του βιβλίου, εξάγονται οι 10 πιο σημαντικές λέξεις, οι οποίες αποθηκεύονται στα metadata μαζί με τον τίτλο του κεφαλαίου. Αυτή η μέθοδος επιτρέπει την οργάνωση του περιεχομένου σε πιο συγκεντρωμένες και περιεκτικές ενότητες, διευκολύνοντας την ανάκτηση πληροφοριών σε πολύπλοκες ερωτήσεις και βελτιώνοντας την ακρίβεια της κατανόησης.

3.3 Υλοποίηση

Η συγκεκριμένη εφαρμογή αποτελεί ένα εκπαιδευτικό σύστημα που χρησιμοποιεί σύγχρονες τεχνολογίες cloud computing, τεχνητής νοημοσύνης και containerization. Η αρχιτεκτονική της είναι δομημένη με γνώμονα την ευελιξία, την επεκτασιμότητα και την απόδοση. Η υποδομή βασίζεται σε έναν συνδυασμό τοπικού διακομιστή (on-premises server), πανεπιστημιακού διακομιστή (remote AI processing) και cloud hosting μέσω Cloudflare.

3.3.1 Αρχιτεκτονική Υποδομής

Τοπικός Server (Self-Hosted Infrastructure). Η εφαρμογή αποτελείται από έναν τοπικό server που φιλοξενεί πολλαπλά Docker containers, καθένα από τα οποία επιτελεί μια συγκεκριμένη λειτουργία, προσφέροντας μια αρθρωτή (modular) αρχιτεκτονική, η οποία επιτρέπει την απομόνωση των υπηρεσιών, διευκολύνει τη διαχείριση και βελτιώνει την επεκτασιμότητα. Με αυτόν τον τρόπο, κάθε υπηρεσία εκτελείται ως ξεχωριστό Docker container, επιτρέποντας την ανεξάρτητη ανάπτυξη, αναβάθμιση και συντήρηση των επιμέρους λειτουργιών της εφαρμογής. Επιπλέον, μέσω OpenVPN tunneling, ο server επικοινωνεί με έναν πανεπιστημιακό server, ο οποίος παρέχει πρόσβαση σε ισχυρούς υπολογιστικούς πόρους και μοντέλα τεχνητής νοημοσύνης. Παρακάτω αναλύονται τα κύρια στοιχεία της αρχιτεκτονικής καθώς και τα πλεονεκτήματα και μειονεκτήματα των τεχνολογιών που επιλέχθηκαν.

3.3.1.1 MariaDB Container

Ρόλος: Το συγκεκριμένο container φιλοξενεί μια MariaDB βάση δεδομένων που χρησιμοποιείται για την αποθήκευση δεδομένων όπως:

- Πληροφορίες χρηστών
- Δεδομένα quiz (ερωτήσεις, απαντήσεις, σκορ)
- Στατιστικά χρήσης

Χρησιμοποιήθηκε η MariaDB ως βάση δεδομένων, καθώς προσφέρει σημαντικά πλεονεκτήματα. Αρχικά, είναι open-source και πλήρως συμβατή με MySQL, γεγονός που διευκολύνει τη μετάβαση από άλλα συστήματα και την ευρεία υποστήριξη από την κοινότητα. Επιπλέον, η MariaDB παρέχει υψηλή απόδοση, ιδιαίτερα σε εφαρμογές που χαρακτηρίζονται από έντονα αναγνωστικά φορτία (read-heavy workloads), καθιστώντας την ιδανική για την αποθήκευση και ανάκτηση δεδομένων με υψηλή ταχύτητα. Ένα άλλο σημαντικό χαρακτηριστικό είναι η υποστήριξη αντιγραφής δεδομένων (replication), που επιτρέπει πλεονασματικότητα (redundancy) και συμβάλλει στη βελτίωση της ανθεκτικότητας και της διαθεσιμότητας του συστήματος. Παράλληλα, η ενεργή ανάπτυξη από την κοινότητα διασφαλίζει συνεχείς βελτιώσεις και ενημερώσεις ασφαλείας.

Ωστόσο, η MariaDB παρουσιάζει και ορισμένα μειονεκτήματα. Συγκεκριμένα, δεν είναι τόσο βελτιστοποιημένη για εφαρμογές μεγάλης κλίμακας όσο η PostgreSQL, η οποία θεωρείται ανώτερη όσον αφορά την επεξεργασία μεγάλων όγκων δεδομένων και τη διαχείριση ταυτόχρονων συναλλαγών. Επιπλέον, η MariaDB δεν προσφέρει την ίδια αποδοτικότητα στη διαχείριση πολύπλοκων δεδομένων JSON, κάτι που μπορεί να αποτελέσει περιορισμό σε εφαρμογές που βασίζονται έντονα σε ημιδομημένα δεδομένα.

Κατά τον σχεδιασμό της εφαρμογής, εξετάστηκαν και άλλες εναλλακτικές επιλογές. Η PostgreSQL θα μπορούσε να είναι μια καλή εναλλακτική εάν η εφαρμογή απαιτούσε πιο ισχυρή υποστήριξη JSON και προηγμένα χαρακτηριστικά διαχείρισης δεδομένων. Από την άλλη, η MongoDB, ως NoSQL βάση δεδομένων, θα μπορούσε να αποτελέσει επιλογή εάν χρειαζόταν μεγαλύτερη ευελιξία στη δομή των δεδομένων και υποστήριξη δυναμικών σχημάτων (dynamic schemas), επιτρέποντας πιο ευέλικτη ανάπτυξη. Παρόλα αυτά, επιλέχθηκε η MariaDB επειδή παρέχει μια ισορροπημένη λύση που συνδυάζει αποδοτικότητα, ευκολία ενσωμάτωσης και αξιοπιστία, καλύπτοντας τις ανάγκες της συγκεκριμένης εφαρμογής.

3.3.1.2 Quiz API Container

Ρόλος: Αυτό το container αποτελεί το backend API που εξυπηρετεί τα quiz. Περιλαμβάνει σημεία πρόσβασης (endpoints) για:

- Δημιουργία, αποθήκευση και ανάκτηση ερωτήσεων
- Καταγραφή απαντήσεων και σκορ
- Πιστοποίηση χρηστών (User authentication) και διαχείριση συνεδριών (session management).

Για την ανάπτυξή του επιλέχθηκε η χρήση του Flask, μιας ελαφριάς και ευέλικτης Python web framework, που επιτρέπει τη γρήγορη ανάπτυξη RESTful APIs με απλό και κατανοητό τρόπο.

Η επιλογή του Flask έγινε κυρίως λόγω της απλότητας και της ευελιξίας του. Επιτρέπει την εύκολη διαχείριση της δρομολόγησης αιτημάτων (routing), της ταυτοποίησης χρηστών (authentication) και της διασύνδεσης με τη βάση δεδομένων (database integration). , ενώ η μεγάλη κοινότητα χρηστών του σημαίνει ότι υπάρχει εκτενής τεκμηρίωση και υποστήριξη. Επιπλέον, η απόδοσή του είναι ικανοποιητική για εφαρμογές με μέτριο φόρτο εργασίας, ενώ η συμβατότητά του με πολλές βιβλιοθήκες και εργαλεία το καθιστά κατάλληλο για σύνθετα backend συστήματα.

Ωστόσο, υπάρχουν και ορισμένα μειονεκτήματα. Το Flask, όντας ένα micro-framework, δεν προσφέρει out-of-the-box λειτουργίες όπως συμβαίνει με πιο ολοκληρωμένα frameworks, με αποτέλεσμα να απαιτείται η ενσωμάτωση πρόσθετων βιβλιοθηκών για λειτουργίες όπως authentication, validation και session management. Επιπλέον, σε περιβάλλοντα με πολύ υψηλό φόρτο εργασίας, το Flask ενδέχεται να μην είναι η πιο αποδοτική λύση, καθώς δεν έχει τον ίδιο βαθμό βελτιστοποίησης για concurrency όπως άλλα frameworks.

Εναλλακτικές επιλογές που εξετάστηκαν ήταν το FastAPI και το Django REST Framework. Το FastAPI προσφέρει εξαιρετική απόδοση, ειδικά σε εφαρμογές που απαιτούν υψηλή ταχύτητα και υποστήριξη για ασύγχρονες λειτουργίες, κάτι που θα μπορούσε να βελτιώσει τη διαχείριση αιτημάτων σε ένα σύστημα με πολλούς ταυτόχρονους χρήστες. Από την άλλη, το Django REST Framework παρέχει μια πιο δομημένη και ολοκληρωμένη λύση, με ενσωματωμένες λειτουργίες για authentication, permissions και data serialization, γεγονός που θα μπορούσε να απλοποιήσει την ανάπτυξη σε μεγαλύτερης κλίμακας εφαρμογές.

3.3.1.3 Backend Container - Langchain & Groq API

Ρόλος: Αυτό το backend container χρησιμοποιεί το Langchain, μια προηγμένη βιβλιοθήκη που επιτρέπει την αλληλεπίδραση με μεγάλα γλωσσικά μοντέλα, και συνδέεται με το Groq API, το οποίο παρέχει πρόσβαση σε ισχυρά AI μοντέλα, όπως το Llama 3.3 70B, το Llama 3.3 8B και το DeepSeek.

Η κύρια λειτουργία του συγκεκριμένου backend είναι η διαχείριση όλων των διεργασιών που σχετίζονται με την τεχνητή νοημοσύνη, προσφέροντας:

- Δημιουργία εξατομικευμένων απαντήσεων, προσαρμοσμένων στις ανάγκες του χρήστη.
- Ανάλυση και κατανόηση φυσικής γλώσσας, βελτιώνοντας την αλληλεπίδραση με το σύστημα.
- Προτάσεις και προσαρμογή περιεχομένου με βάση τις απαντήσεις των χρηστών.

Η επιλογή του Langchain βασίστηκε σε ορισμένα σημαντικά πλεονεκτήματα που προσφέρει:

- Υποστήριξη AI pipelines: Επιτρέπει τη δημιουργία ροών εργασίας τεχνητής νοημοσύνης, οι οποίες μπορούν να συνδυάσουν πολλαπλά LLMs για πιο ακριβείς και αποδοτικές απαντήσεις.
- Συμβατότητα με πολλές βάσεις δεδομένων και τεχνικές ανάκτησης γνώσης RAG, επιτρέποντας την ενσωμάτωση εξωτερικών πληροφοριών στις απαντήσεις.
- Εύκολη διασύνδεση με APIs, όπως OpenAI και Groq, προσφέροντας ευελιξία στην επιλογή μοντέλων τεχνητής νοημοσύνης.

Παρόλα αυτά, υπάρχουν και ορισμένα μειονεκτήματα στη χρήση του:

- Αυξημένες απαιτήσεις σε μνήμη και επεξεργαστική ισχύ, καθώς η εκτέλεση μεγάλων μοντέλων απαιτεί ισχυρούς υπολογιστικούς πόρους.
- Ανάγκη συνεχούς παραμετροποίησης (fine-tuning) ώστε να εξασφαλιστεί η βέλτιστη ακρίβεια και απόδοση των μοντέλων, γεγονός που αυξάνει τη συνολική πολυπλοκότητα.

Εναλλακτικές τεχνολογίες που θα μπορούσαν να χρησιμοποιηθούν αντί του Langchain περιλαμβάνουν:

- LlamaIndex, το οποίο προσφέρει πιο εξειδικευμένες δυνατότητες για αναζήτηση και οργάνωση γνώσης, ειδικά σε εφαρμογές που απαιτούν εξελιγμένη ανάκτηση πληροφορίας (Retrieval – Based AI).
- Haystack, μια ανοιχτού κώδικα βιβλιοθήκη για document search και NLP pipelines, που θα μπορούσε να αποτελέσει μια πιο εξειδικευμένη επιλογή για εφαρμογές που επικεντρώνονται στην αναζήτηση και την κατανόηση κειμένου.

Η τελική επιλογή του Langchain έγινε επειδή προσφέρει τη μέγιστη δυνατή ευελιξία, επιτρέποντας τη σύνθεση διαφορετικών AI εργαλείων και τεχνικών για την ανάπτυξη μιας ισχυρής, modular αρχιτεκτονικής.

3.3.1.4 My Teacher Container - LangGraph

Ρόλος: Το My Teacher Container είναι ένα backend container που παρέχει AI – driven καθοδήγηση, προσφέροντας μια εξατομικευμένη μαθησιακή εμπειρία στους χρήστες. Η τεχνολογία που χρησιμοποιείται είναι η LangGraph, η οποία επιτρέπει τη δημιουργία διαδραστικών εμπειριών

συνομιλίας (chatbot-like interactions), δίνοντας τη δυνατότητα στον χρήστη να αλληλεπιδράσει με ένα σύστημα τεχνητής νοημοσύνης που προσφέρει προσωποποιημένες εκπαιδευτικές συμβουλές [42].

Η επιλογή της LangGraph βασίστηκε στα εξής πλεονεκτήματα:

- Ιδανικό για διαδραστικές AI εφαρμογές, καθώς επιτρέπει την ανάπτυξη πολύπλοκων συνομιλιακών ροών που προσαρμόζονται δυναμικά με βάση τις απαντήσεις των χρηστών.
- Δυνατότητα χαρτογράφησης πολύπλοκων διαλόγων, επιτρέποντας τη δημιουργία stateful συνομιλιών, όπου το σύστημα μπορεί να διατηρεί πληροφορίες και να προσαρμόζει τις απαντήσεις του ανάλογα με την προηγούμενη αλληλεπίδραση.
- Βελτιωμένη κατανόηση και προσαρμογή απαντήσεων, με αποτέλεσμα ένα πιο φυσικό και ευχάριστο περιβάλλον για τον χρήστη.

Ωστόσο, η χρήση της LangGraph συνοδεύεται και από ορισμένα μειονεκτήματα:

- Περιορισμένη τεκμηρίωση, γεγονός που μπορεί να δυσκολέψει την ενσωμάτωσή της και την ανάπτυξη σύνθετων διαλογικών ροών.
- Αυξημένη ανάγκη για fine-tuning, ώστε να διαμορφωθεί μια ομαλή και φυσική συνομιλιακή εμπειρία για τον χρήστη, καθώς απαιτεί συνεχή εκπαίδευση και ρύθμιση των απαντήσεων του AI μοντέλου.

Εναλλακτικές τεχνολογίες που θα μπορούσαν να χρησιμοποιηθούν αντί της LangGraph περιλαμβάνουν:

- Rasa, μια ισχυρή open-source πλατφόρμα για conversational AI, η οποία προσφέρει μεγαλύτερη ευελιξία στη δημιουργία διαλογικών chatbot με state management και NLU (Natural Language Understanding) [43].
- Microsoft Bot Framework, που παρέχει ενσωματωμένες δυνατότητες για AI – driven conversational agents, με εκτεταμένη τεκμηρίωση και υποστήριξη από τη Microsoft.

Η επιλογή της LangGraph έγινε επειδή προσφέρει μια αρθρωτή (modular) προσέγγιση στη δημιουργία διαδραστικών, εξατομικευμένων chatbot, εστιάζοντας στην εκπαίδευση και την καθοδήγηση των χρηστών, καθιστώντας την ιδανική λύση για το συγκεκριμένο έργο.

3.3.1.5 React Frontend Container

Ρόλος: Το React Frontend Container αποτελεί το frontend της εφαρμογής, το οποίο είναι υλοποιημένο σε React.js, μια από τις πιο δημοφιλείς βιβλιοθήκες JavaScript για τη δημιουργία διαδραστικών και δυναμικών διεπαφών χρήστη (UI – User Interface).

Η επιλογή του React.js έγινε λόγω των σημαντικών πλεονεκτημάτων που προσφέρει:

- Αρχιτεκτονική βασισμένη σε components, η οποία επιτρέπει την επαναχρησιμοποίηση στοιχείων, βελτιώνοντας έτσι την οργάνωση του κώδικα και διευκολύνοντας τη συντήρηση και την επεκτασιμότητα της εφαρμογής.
- Virtual DOM, ένας μηχανισμός που μειώνει τις περιττές ενημερώσεις στο πραγματικό DOM, βελτιώνοντας αισθητά την απόδοση και την ταχύτητα rendering της εφαρμογής.
- Μεγάλη κοινότητα και πλούσιο οικοσύστημα, προσφέροντας πληθώρα εργαλείων, βιβλιοθηκών και υποστήριξης, που διευκολύνουν την ανάπτυξη και τη συντήρηση του κώδικα.

Ωστόσο, η χρήση του React.js συνοδεύεται και από ορισμένα μειονεκτήματα:

- Αυξημένη πολυπλοκότητα καθώς η εφαρμογή μεγαλώνει, καθώς η διαχείριση της κατάστασης (state management) μπορεί να γίνει δύσκολη χωρίς τη χρήση εξωτερικών βιβλιοθηκών όπως Redux ή Zustand.
- Συχνές ενημερώσεις και αλλαγές στις εξαρτήσεις, που απαιτούν προσεκτική διαχείριση για την αποφυγή ασυμβατότητας μεταξύ των διαφόρων πακέτων.

Εναλλακτικές τεχνολογίες που θα μπορούσαν να χρησιμοποιηθούν:

- Vue.js, το οποίο προσφέρει απλούστερη σύνταξη και είναι πιο φιλικό για μικρότερες εφαρμογές, διατηρώντας παράλληλα υψηλή απόδοση.
- Svelte, που μεταγλωττίζει τον κώδικα σε αποδοτικό JavaScript κατά το build time, εξαλείφοντας την ανάγκη για Virtual DOM και βελτιώνοντας την απόδοση.
- Angular, ένα πιο ολοκληρωμένο framework που προσφέρει ενσωματωμένα εργαλεία για state management, dependency injection και form handling, αλλά μπορεί να είναι πιο περίπλοκο στη μάθηση και την υλοποίηση.

Η επιλογή του React.js έγινε επειδή παρέχει ιδανική ισορροπία μεταξύ απόδοσης, επεκτασιμότητας και ευκολίας ανάπτυξης, καθιστώντας το κατάλληλο για εφαρμογές που απαιτούν διαδραστικό UI και ταχύτητα απόκρισης.

3.3.1.6 Docker Nginx (Reverse Proxy)

Ρόλος: Το Nginx χρησιμοποιείται ως reverse proxy στο περιβάλλον Docker, διευκολύνοντας την πρόσβαση και τη δρομολόγηση των αιτήσεων προς τα διάφορα containers που φιλοξενούν τις υπηρεσίες της εφαρμογής. Με αυτόν τον τρόπο, επιτυγχάνεται αποτελεσματική διαχείριση της κυκλοφορίας, βελτιστοποιείται η απόδοση και ενισχύεται το επίπεδο ασφάλειας.

Η επιλογή του Nginx έγινε λόγω των σημαντικών πλεονεκτημάτων που προσφέρει:

- Ελαφρύς και γρήγορος: Το Nginx χαρακτηρίζεται από χαμηλές απαιτήσεις σε πόρους και υψηλή απόδοση, επιτρέποντας την ταχεία επεξεργασία αιτήσεων και την αποτελεσματική λειτουργία σε περιβάλλοντα υψηλού φορτίου.
- Υποστήριξη load balancing: Η δυνατότητα κατανομής των εισερχόμενων αιτήσεων σε πολλαπλά containers συμβάλλει στη διατήρηση της διαθεσιμότητας και της αξιοπιστίας της εφαρμογής, αποφεύγοντας την υπερφόρτωση μεμονωμένων κόμβων.
- Ασφάλεια μέσω HTTPS και rate limiting: Η ενσωματωμένη υποστήριξη για ασφαλείς συνδέσεις μέσω HTTPS, σε συνδυασμό με μηχανισμούς περιορισμού του ρυθμού των αιτήσεων (rate limiting), προσφέρει ένα πρόσθετο επίπεδο προστασίας απέναντι σε επιθέσεις και κακόβουλες ενέργειες.

Μειονεκτήματα:

- Απαιτούμενη επιπρόσθετη διαμόρφωση για πολύπλοκα routing σενάρια:Σε περιπτώσεις όπου απαιτείται εξειδικευμένη δρομολόγηση, η παραμετροποίηση του Nginx μπορεί να αποδειχθεί περίπλοκη και χρονοβόρα, απαιτώντας επιπλέον προσπάθεια από τους διαχειριστές της υποδομής.

Εναλλακτικές τεχνολογίες που θα μπορούσαν να χρησιμοποιηθούν:

- HAProxy: Μια αξιόπιστη λύση για load balancing, γνωστή για την υψηλή της απόδοση και τη σταθερότητα σε περιβάλλοντα με μεγάλα φορτία.
- Traefik: Ένας σύγχρονος reverse proxy σχεδιασμένος ειδικά για containerized εφαρμογές, ο οποίος προσφέρει αυτόματη διαμόρφωση και δυναμική διαχείριση των routes, μειώνοντας έτσι την ανάγκη για χειροκίνητη παραμετροποίηση.

Η υιοθέτηση του Nginx ως reverse proxy στο περιβάλλον Docker αποδεικνύεται κρίσιμη για την ομαλή λειτουργία και την ασφάλεια της εφαρμογής. Παρά το μειονέκτημα της επιπρόσθετης διαμόρφωσης για πολύπλοκα routing σενάρια, τα πλεονεκτήματα που προσφέρει σε όρους απόδοσης, load balancing και ασφάλειας τον καθιστούν την προτιμώμενη επιλογή για τη διαχείριση της κυκλοφορίας και τη βελτιστοποίηση της υποδομής.

3.3.1.7 Backend Container - Πανεπιστημιακός Server & AI Διαχείριση

Ρόλος: Αυτό το backend container λειτουργεί σε Docker και συνδέεται με τον πανεπιστημιακό server μέσω VPN, επιτρέποντας ασφαλή διαχείριση AI διεργασιών και ανάκτηση πληροφορίας. Παρέχει [44]:

- Ασφαλή VPN σύνδεση με προστασία δεδομένων.
- Υποστήριξη AI μοντέλων για ανάλυση φυσικής γλώσσας.
- Ευέλικτη αρχιτεκτονική, επιτρέποντας εύκολη επέκταση.

Τεχνολογίες:

- Ubuntu 22.04, Flask, Sentence-Transformers για AI επεξεργασία.
- OpenVPN για ασφαλή σύνδεση.
- Python3 & Pip για διαχείριση βιβλιοθηκών.

Πλεονεκτήματα:

- Ασφαλής πρόσβαση στον πανεπιστημιακό server.
- Τοπική εκτέλεση AI μοντέλων, μειώνοντας εξωτερικές εξαρτήσεις.
- Επεκτασιμότητα και ευελιξία.

Προκλήσεις:

- Αυξημένες απαιτήσεις υπολογιστικών πόρων.
- Διαχείριση VPN σύνδεσης.
- Συνεχής ενημέρωση AI μοντέλων.

Εναλλακτικές:

- FastAPI αντί για Flask.
- LlamaIndex ή Haystack για αναζήτηση πληροφορίας.
- WireGuard αντί για OpenVPN.

3.3.2 Διασύνδεση

3.3.2.1 Σύνδεση με Πανεπιστημιακό Server μέσω OpenVPN

Ρόλος: Ένα backend container αναλαμβάνει την επικοινωνία με τον server του πανεπιστημίου μέσω OpenVPN tunneling. Ο ρόλος του είναι να εξασφαλίσει μια ασφαλή, κρυπτογραφημένη σύνδεση μεταξύ του container και του server, επιτρέποντας την αξιόπιστη μεταφορά δεδομένων σε ένα περιβάλλον που απαιτεί αυξημένη ασφάλεια.

Πλεονεκτήματα του OpenVPN:

- Ισχυρή κρυπτογράφηση: Το OpenVPN παρέχει υψηλό επίπεδο κρυπτογράφησης, προστατεύοντας τα δεδομένα εν μέσω μεταφοράς και διασφαλίζοντας την εμπιστευτικότητα και την ακεραιότητα των πληροφοριών.
- Ασφαλής μεταφορά δεδομένων: Η χρήση του OpenVPN εγγυάται την ασφαλή μετάδοση των δεδομένων, μειώνοντας τον κίνδυνο παραβίασης και επιθέσεων κατά την επικοινωνία με τον server του πανεπιστημίου.
- Ευρεία υποστήριξη από λειτουργικά συστήματα: Το OpenVPN είναι διαθέσιμο και υποστηρίζεται από μια πληθώρα λειτουργικών συστημάτων, διευκολύνοντας την ενσωμάτωση και τη διαχείριση σε ποικίλα περιβάλλοντα.

Μειονεκτήματα:

- Υπερβολική πολυπλοκότητα σε μεγάλες υλοποιήσεις: Σε περιπτώσεις εκτεταμένων και πολύπλοκων υποδομών, η παραμετροποίηση και διαχείριση του OpenVPN μπορεί να αποδειχθεί αρκετά απαιτητική, αυξάνοντας την πολυπλοκότητα της υλοποίησης.
- Πιθανή εμφάνιση latency: Αν δεν ρυθμιστεί σωστά, το OpenVPN μπορεί να προκαλέσει καθυστερήσεις (latency) στην επικοινωνία, κάτι που ενδεχομένως να επηρεάσει την απόδοση της εφαρμογής σε περιβάλλοντα υψηλής κυκλοφορίας.

Εναλλακτικές τεχνολογίες:

- WireGuard: Μια σύγχρονη τεχνολογία VPN που προσφέρει απλούστερη διαμόρφωση και υψηλότερη απόδοση, ενώ διατηρεί τα απαιτούμενα επίπεδα ασφάλειας.
- IPSec: Ένα καθιερωμένο πρωτόκολλο VPN που παρέχει αξιόπιστη ασφάλεια, αν και η ρύθμιση του μπορεί να είναι πιο πολύπλοκη ανάλογα με τις απαιτήσεις της υλοποίησης.

Η υλοποίηση του OpenVPN tunneling στο backend container εξυπηρετεί κρίσιμους στόχους ασφαλούς επικοινωνίας με τον server του πανεπιστημίου, παρέχοντας υψηλή ασφάλεια και αξιόπιστη μεταφορά δεδομένων. Παρά την πολυπλοκότητα που μπορεί να επιφέρει σε μεγαλύτερες υλοποιήσεις και την πιθανότητα εμφάνισης latency, τα πλεονεκτήματα του OpenVPN το καθιστούν μια κατάλληλη επιλογή για περιβάλλοντα όπου η ασφάλεια και η προστασία των δεδομένων είναι προτεραιότητα. Εναλλακτικές λύσεις όπως το WireGuard και το IPSec μπορούν να αξιολογηθούν σε μελλοντικές αναβαθμίσεις, με στόχο τη βελτίωση της απόδοσης και της απλοποίησης της διαχείρισης.

3.3.2.2 Σύνδεση με Groq api

Ένα backend container συνδέεται με το Groq API μέσω του Langchain, επιτρέποντας την άμεση πρόσβαση σε μεγάλα γλωσσικά μοντέλα όπως τα Llama 3.3 70B, Llama 3.3 8B και το DeepSeek. Μέσω αυτής της ενσωμάτωσης, διαχειρίζεται αποτελεσματικά AI pipelines για την παραγωγή εξατομικευμένων απαντήσεων και την ανάλυση φυσικής γλώσσας [45].

3.3.2.3 Σύνδεση μέσω Tunneling με Cloudflare

Η εφαρμογή μας εκτίθεται στον εξωτερικό κόσμο μέσω μιας σύνδεσης tunneling που υλοποιείται από την υπηρεσία Cloudflare. Σε αυτό το σενάριο, το Cloudflare λειτουργεί ως το «παράθυρο» της εφαρμογής, δρομολογώντας με ασφάλεια τις εισερχόμενες αιτήσεις από το διαδίκτυο προς τον backend server. Η σύνδεση αυτή βασίζεται στο πρωτόκολλο HTTPS, το οποίο εξασφαλίζει την κρυπτογράφηση και την ακεραιότητα των δεδομένων κατά τη μεταφορά.

Λεπτομέρειες της Σύνδεσης [46]:

- Δωρεάν Υπηρεσία: Το Cloudflare παρέχει ένα δωρεάν πακέτο που καλύπτει τις βασικές ανάγκες όπως DNS management, reverse proxy, caching και προστασία από επιθέσεις DDoS. Αυτή η δωρεάν έκδοση καθιστά την υλοποίηση οικονομικά αποδοτική χωρίς να θυσιάζεται η ασφάλεια ή η απόδοση.
- Ασφάλεια μέσω HTTPS: Όλες οι επικοινωνίες μεταξύ των χρηστών και της εφαρμογής πραγματοποιούνται μέσω του πρωτοκόλλου HTTPS. Αυτή η μέθοδος εξασφαλίζει ότι οι μεταφερόμενες πληροφορίες κρυπτογραφούνται, προστατεύοντας έτσι τα δεδομένα από πιθανές υποκλοπές ή επιθέσεις.
- Tunneling και Reverse Proxy: Μέσω της τεχνολογίας tunneling, το Cloudflare λειτουργεί ως ενδιάμεσος διακομιστής (reverse proxy) που λαμβάνει τις εισερχόμενες αιτήσεις και τις προωθεί στο κατάλληλο backend container. Αυτή η διαδικασία επιτρέπει την απόκρυψη της πραγματικής διεύθυνσης IP του server, παρέχοντας επιπλέον επίπεδα ασφάλειας και διαχείρισης κυκλοφορίας.
- Διεύθυνση Πρόσβασης: Η εφαρμογή είναι δημόσια προσβάσιμη μέσω της διεύθυνσης https://ab.aieducation.icu, επιτρέποντας στους χρήστες εύκολη και ασφαλή πρόσβαση από οπουδήποτε στον κόσμο.

Η ενσωμάτωση του Cloudflare ως υπηρεσίας tunneling και reverse proxy επιτρέπει την ασφαλή και αποτελεσματική έκθεση της εφαρμογής στο διαδίκτυο. Η χρήση δωρεάν υπηρεσιών, η κρυπτογράφηση μέσω HTTPS και οι δυνατότητες διαχείρισης κυκλοφορίας καθιστούν το Cloudflare την ιδανική επιλογή για την προστασία και βελτιστοποίηση της πρόσβασης στην εφαρμογή μας [46].

3.3.2.4 Πανεπιστημιακός Server

Ο πανεπιστημιακός server αποτελεί έναν πυλώνα της υποδομής μας, παρέχοντας ισχυρούς υπολογιστικούς πόρους, μεταξύ των οποίων η επιτάχυνση μέσω GPU, για τη λειτουργία προχωρημένων μοντέλων τεχνητής νοημοσύνης. Σε αυτό το περιβάλλον, υλοποιούνται και αξιοποιούνται εξειδικευμένα μοντέλα με στόχο την εκτέλεση απαιτητικών εργασιών AI, προσφέροντας υψηλή απόδοση και αποτελεσματικότητα.

Χρησιμοποιούμενα AI Μοντέλα:

- Ollama για AI inference: Το Ollama χρησιμοποιείται για την υλοποίηση διαδικασιών inference, επιτρέποντας την παραγωγή αποτελεσμάτων από τα δεδομένα εισόδου.
- Llama 3.3 70B με μείωση αριθμητικής ακρίβειας σε 3 bits: Το συγκεκριμένο μοντέλο έχει υποστεί μείωση αριθμητικής ακρίβειας (quantization) στα 3 bits, επιτρέποντας την αποτελεσματική χρήση των διαθέσιμων πόρων . Αυτή η τεχνική μειώνει το μέγεθος του μοντέλου και τις απαιτήσεις σε μνήμη, καθιστώντας το ιδανικό για εφαρμογές με περιορισμένους υπολογιστικούς πόρους.
- Meltemi model (fully quantized AI model): Στο πλαίσιο της έρευνάς μας, χρησιμοποιήθηκε και το μοντέλο Meltemi, το πρώτο μεγάλο γλωσσικό μοντέλο (LLM) ανοιχτού κώδικα για την ελληνική γλώσσα, το οποίο αναπτύχθηκε από το Ινστιτούτο Επεξεργασίας του Λόγου του Ερευνητικού Κέντρου "Αθηνά". Το Meltemi έχει εκπαιδευτεί σε ένα εκτενές σώμα ελληνικών κειμένων, περιλαμβάνοντας και ελληνικά βιβλία, με στόχο την κατανόηση και παραγωγή φυσικής γλώσσας στα ελληνικά. Παρόλο που χρησιμοποιήθηκε η πλήρως κβαντισμένη έκδοση του μοντέλου, τα αποτελέσματα δεν ήταν συγκρίσιμα με εκείνα του Llama 3.3 και άλλων μοντέλων [47].

3.4 Υλοποίηση Εφαρμογής με Δωρεάν Εργαλεία και Βελτιστοποίηση Πρόσβασης σε AI APIs

Στο παρακάτω κεφάλαιο περιγράφεται η ανάπτυξη μιας εφαρμογής βασισμένης αποκλειστικά σε δωρεάν εργαλεία και υπηρεσίες. Ο κύριος στόχος ήταν η δημιουργία ενός ευέλικτου και οικονομικά αποδοτικού συστήματος που να εκμεταλλεύεται διαθέσιμες δωρεάν υπηρεσίες, διατηρώντας παράλληλα υψηλή απόδοση και αξιοπιστία.

Υποδομή και Ανάπτυξη της Εφαρμογής. Για την υλοποίηση της εφαρμογής, επιλέχθηκε η χρήση ενός φορητού υπολογιστή φοιτητή ως βασικός server, ο οποίος φιλοξενεί την εφαρμογή. Η πρόσβαση στον εξωτερικό κόσμο εξασφαλίστηκε μέσω της υπηρεσίας Cloudflare, η οποία παρέχει ασφαλή και γρήγορη σύνδεση χωρίς την ανάγκη δημόσιας IP διεύθυνσης ή προχωρημένων ρυθμίσεων δικτύου [46].

Παράλληλα, για την εκτέλεση μοντέλων μηχανικής μάθησης, χρησιμοποιήθηκε ο server του πανεπιστημίου, όπου τρέχουν μοντέλα με χαμηλότερο κβαντισμό, επιτρέποντας τη διατήρηση μιας ισορροπίας μεταξύ υπολογιστικής απόδοσης και κατανάλωσης πόρων.

3.4.1 Χρήση του Groq API και Διαχείριση Κλειδιών API.

Το Groq API επιλέχθηκε ως μία από τις κύριες πλατφόρμες παροχής AI υπηρεσιών, δεδομένου ότι προσφέρει δωρεάν χρήση με ορισμένους περιορισμούς. Οι περιορισμοί αυτοί περιλαμβάνουν:

- Περιορισμένο αριθμό αιτημάτων ανά λεπτό και ανά ημέρα.
- Όριο στα tokens που μπορούν να καταναλωθούν σε συγκεκριμένο χρονικό διάστημα.

Για να ξεπεραστούν οι παραπάνω περιορισμοί και να διασφαλιστεί η συνεχής λειτουργία της εφαρμογής, δημιουργήθηκε το αρχείο GroqKeyManager.py. Η κλάση GroqKeyManager αναλαμβάνει τη διαχείριση πολλαπλών API κλειδιών και την αυτόματη εναλλαγή τους όταν κάποιο εξαντληθεί [45].

Στην παρούσα διπλωματική εργασία παρουσιάζεται η ανάπτυξη ενός chatbot που αξιοποιεί την τεχνική Retrieval Augmented Generation (RAG), επιτρέποντας στους μαθητές να αναζητούν και να επεξεργάζονται πληροφορίες απευθείας από τα βιβλία τους, όπως το βιβλίο γεωγραφίας της πέμπτης δημοτίκου και το κείμενο για τη Ζώη του Αριστοτέλη. Η εφαρμογή υλοποιείται με έναν υβριδικό τρόπο, συνδυάζοντας την παραδοσιακή ανάπτυξη σε Python με το κρίσιμο ρόλο του prompt engineering, το οποίο καθοδηγεί το σύγχρονο μοντέλο γλώσσας στην εξαγωγή των επιθυμητών πληροφοριών.

Αρχικά, η εφαρμογή επεξεργάζεται τα PDF αρχεία ώστε να μειώσει το όγκο του κειμένου που θα υποβληθεί στο μοντέλο. Συγκεκριμένα, εξάγονται οι πρώτες δέκα σελίδες από το αρχικό έγγραφο, διασφαλίζοντας έτσι ότι το περιεχόμενο που θα αναλυθεί είναι επαρκές και ταυτόχρονα εντός των ορίων του context window του LLM. Το επεξεργασμένο αρχείο ανεβαίνει έπειτα στο σύστημα της Google, όπου χρησιμοποιούνται οι εκδόσεις Gemini 1.5 Pro και Gemini 1.5 Flash, οι οποίες διαθέτουν μεγαλύτερο παράθυρο επεξεργασίας και επιτρέπουν την ανάλυση μεγαλύτερου όγκου δεδομένων.

Ένας από τους βασικούς πυλώνες της υλοποίησης είναι το prompt engineering, που επιτρέπει τον ακριβή καθορισμό των οδηγιών προς το LLM. Μέσω ενός προσεκτικά διαμορφωμένου prompt, το μοντέλο κατευθύνεται ώστε να διαβάσει το περιεχόμενο του αρχείου και να εξαγάγει τα ονόματα των κεφαλαίων καθώς και τις σελίδες στις οποίες ξεκινούν. Το αποτέλεσμα επιστρέφεται σε δομημένη μορφή JSON, επιτρέποντας τη συνέπειά του σε επόμενα στάδια επεξεργασίας. Στη συνέχεια, τα δεδομένα αυτά αποθηκεύονται σε μορφή CSV, διευκολύνοντας την περαιτέρω χρήση τους, όπως για τη δημιουργία πίνακα περιεχομένων.

Σε συνέχεια της εφαρμογής, υλοποιείται μια διαδικασία διαχωρισμού του αρχικού PDF σε ξεχωριστά αρχεία για κάθε κεφάλαιο. Χρησιμοποιώντας το CSV αρχείο που περιέχει τους τίτλους και τις σελίδες εκκίνησης των κεφαλαίων, υπολογίζονται αυτόματα τα όρια (αρχική και τελική σελίδα) για κάθε τμήμα του βιβλίου. Στη συνέχεια, δημιουργούνται νέα PDF αρχεία, με καθαρισμένα ονόματα για την αποφυγή ειδικών χαρακτήρων και τη διατήρηση της σωστής ακολουθίας, διευκολύνοντας έτσι την πρόσβαση και την περαιτέρω επεξεργασία του περιεχομένου ανά κεφάλαιο.

Σε περίπτωση που το αρχείο δεν διαθέτει πίνακα περιεχομένων, υιοθετείται μια εναλλακτική διαδικασία για την εξαγωγή των κεφαλαίων και τη διαίρεση του αρχικού PDF. Αρχικά, με το script split_papers.py χρησιμοποιείται το μοντέλο Gemini (μέσω του API) για την ανάλυση του κειμένου του PDF. Το μοντέλο δέχεται ένα προσαρμοσμένο prompt που του ζητά να εντοπίσει τους τίτλους των παραγράφων ή ενοτήτων καθώς και τις σελίδες στις οποίες εμφανίζονται. Το αποτέλεσμα της ανάλυσης επιστρέφεται σε μορφή JSON και στη συνέχεια αποθηκεύεται ως αρχείο CSV, όπου καταγράφονται οι τίτλοι και οι αντίστοιχες σελίδες εκκίνησης των κεφαλαίων.

Με τη βοήθεια του script save_papers_from_csv.py, το παραγόμενο CSV χρησιμοποιείται για να διαχωριστεί το αρχικό PDF σε ξεχωριστά αρχεία. Συγκεκριμένα, για κάθε κεφάλαιο υπολογίζεται το διάστημα σελίδων που το περιέχει (έως την έναρξη του επόμενου ή μέχρι το τέλος του αρχείου). Σε περίπτωση που το κείμενο ενός κεφαλαίου υπερβαίνει το όριο των tokens, υιοθετείται μια επιπλέον διαδικασία διαχωρισμού σε υποενότητες, χρησιμοποιώντας τεχνικές που βασίζονται σε tokenization (με τη χρήση του GPT2TokenizerFast). Κάθε κεφάλαιο ή υποενότητα αποθηκεύεται έπειτα ως ξεχωριστό PDF αρχείο με κατάλληλη ονοματολογία, διασφαλίζοντας την ομαλή και οργανωμένη πρόσβαση στο περιεχόμενο.

Η υβριδική αυτή προσέγγιση, όπου ο παραδοσιακός κώδικας Python συνεργάζεται στενά με τις τεχνικές του prompt engineering, αποδεικνύει πόσο σημαντική είναι η ακριβής καθοδήγηση του μοντέλου γλώσσας για την επίτευξη αξιόπιστων και δομημένων αποτελεσμάτων. Η χρήση του Gemini LLM, σε

συνδυασμό με το λεπτομερώς σχεδιασμένο prompt, καταδεικνύει την αποτελεσματικότητα της ενσωμάτωσης παραδοσιακών τεχνικών προγραμματισμού με τις σύγχρονες μεθοδολογίες τεχνητής νοημοσύνης, επιτρέποντας την ανάπτυξη εργαλείων που απαντούν στις ανάγκες των χρηστών με ακρίβεια και ταχύτητα.

Σε αυτό το στάδιο της εφαρμογής δημιουργούνται αυτόματα ερωτήσεις και σωστές απαντήσεις για κάθε κεφάλαιο, οι οποίες θα χρησιμεύσουν ως βάση τόσο για retrieval όσο και για τη δημιουργία ενός quiz που μπορεί να απαντήσει ο μαθητής. Για την υλοποίηση αυτής της λειτουργίας, χρησιμοποιείται το script make_questions.py.

Το συγκεκριμένο script αξιοποιεί το μοντέλο Gemini μέσω του API της Google, στέλνοντας ένα προσαρμοσμένο prompt που καθοδηγεί το LLM να αναλύσει το κείμενο του κεφαλαίου και να παράγει τουλάχιστον 20 ερωτοαπαντήσεις. Οι ερωτήσεις καλύπτουν διάφορες πτυχές του περιεχομένου, όπως ορισμούς, τοποθεσίες, περιγραφές γεωγραφικών χαρακτηριστικών και άλλες βασικές πληροφορίες. Η έξοδος από το μοντέλο παρέχεται σε μορφή JSON, η οποία στη συνέχεια επεξεργάζεται και αποθηκεύεται σε ένα αρχείο CSV με τις στήλες "Question" και "Answer".

Με αυτήν την προσέγγιση, επιτυγχάνεται όχι μόνο η ανάκτηση στοχευμένων πληροφοριών από κάθε κεφάλαιο, αλλά και η δημιουργία διαδραστικού υλικού για αξιολόγηση, ενισχύοντας έτσι τη μαθησιακή διαδικασία με δομημένο και εύχρηστο περιεχόμενο.

Σε αυτό το στάδιο της εφαρμογής, στοχεύουμε στη δημιουργία λανθασμένων ερωτήσεων που θα χρησιμοποιηθούν για το quiz. Το αρχείο make_wrong_answers.py αναλαμβάνει να επεξεργαστεί κάθε ήδη παραχθείσα ερώτηση μαζί με τη σωστή της απάντηση, ώστε να παραχθούν τέσσερις ρεαλιστικές αλλά λανθασμένες απαντήσεις. Με τη χρήση του μοντέλου Gemini και ενός προσαρμοσμένου prompt, το σύστημα λαμβάνει ως είσοδο την ερώτηση και τη σωστή απάντηση, και επιστρέφει μια λίστα με τέσσερις πιθανές λανθασμένες απαντήσεις, που παρουσιάζουν κάποια αληθή στοιχεία αλλά δεν είναι ολόκληρα σωστές.

Για την αντιμετώπιση πιθανών σφαλμάτων ή μη αναμενόμενων αποκρίσεων, έχει ενσωματωθεί μηχανισμός επαναπροσπάθειας (retry mechanism) ο οποίος, σε περίπτωση αποτυχίας της κλήσης στο API ή προβλημάτων αποκωδικοποίησης του JSON, επαναλαμβάνει τη διαδικασία με αλλαγή ρυθμίσεων (API keys) και καθυστερήσεις μεταξύ των προσπαθειών. Τελικά, οι ερωτήσεις μαζί με τη σωστή και τις τέσσερις λανθασμένες απαντήσεις αποθηκεύονται σε ένα CSV αρχείο. Το αρχείο με τις ερωτήσεις και τις σωστές απαντήσεις θα χρησιμοποιηθεί ως βάση για retrieval, δηλαδή για την ανάκτηση πληροφοριών όταν ο μαθητής χρειάζεται να βρει σχετικές απαντήσεις. Παράλληλα, το ίδιο αρχείο, εμπλουτισμένο με τις λανθασμένες απαντήσεις που δημιουργήσαμε, θα χρησιμεύσει για τη δημιουργία του διαδραστικού quiz.

Έτσι, το σύστημα θα μπορεί:

- Να απαντά σε ερωτήσεις των μαθητών μέσω του retrieval, χρησιμοποιώντας τις σωστές απαντήσεις που εξήχθησαν από το κείμενο.
- Να δημιουργεί quiz με ερωτήσεις πολλαπλής επιλογής, όπου ο μαθητής θα διαλέγει ανάμεσα στη σωστή και στις λανθασμένες απαντήσεις.

Αυτή η διπλή χρήση του dataset βοηθά στη βελτίωση της μαθησιακής διαδικασίας, συνδυάζοντας ενεργητική μάθηση μέσω quiz και παθητική μάθηση μέσω αναζήτησης πληροφοριών.

3.4.2 Δημιουργία Διανυσματικών Βάσεων Δεδομένων για Βελτιστοποιημένη Ανάκτηση Κειμένου

3.4.2.1 Βάση Multimodal

Για τη δημιουργία του μοντέλου, χρησιμοποιούνται βιβλιοθήκες της Python. Για την επεξεργασία του περιεχομένου του PDF, δηλαδή την εξαγωγή του κειμένου, των πινάκων και των εικόνων, χρησιμοποιήθηκαν οι βιβλιοθήκες:

- Unstructured με την παράμετρο [all – docs]. Η unstructured είναι μια βιβλιοθήκη σχεδιασμένη για επεξεργασία και ανάλυση μη δομημένων δεδομένων, όπως αρχεία PDF, έγγραφα Word, HTML και άλλα κείμενα. Το [all-docs] διασφαλίζει ότι εγκαθίστανται όλες οι εξαρτήσεις για την επεξεργασία εγγράφων. Επίσης, από την unstructured εισήχθησαν οι λειτουργίες partition_pdf και pytesseract. Η partition_pdf χρησιμοποιείται για την ανάλυση και εξαγωγή περιεχομένου από αρχεία PDF. Αυτή η συνάρτηση εξάγει το κείμενο από τις σελίδες ενός PDF και το οργανώνει σε δομημένα δεδομένα, όπως λίστες ή λεξικά, καθιστώντας εύκολη την επεξεργασία του περιεχομένου. Το unstructured – pytesseract είναι ένα χρήσιμο εργαλείο για OCR (Optical Character Recognition) σε PDF και εικόνες μέσα από το Unstructured framework, επιτρέποντας την ανάλυση και εξαγωγή κειμένου από μη δομημένα έγγραφα.
- PyPDF2. Η PyPDF2 χρησιμοποιείται για ανάγνωση και επεξεργασία αρχείων PDF, με τις εντολές PdfReader και PdfWriter για ανάγνωση και εγγραφή αρχείων PDF.
- PyMuPDF. Η PyMuPDF είναι μια βιβλιοθήκη για ανάγνωση, επεξεργασία και εξαγωγή δεδομένων από αρχεία PDF.
- Poppler – utils. Η Poppler – utils είναι μια συλλογή εργαλείων για την επεξεργασία αρχείων PDF, βασισμένη στη βιβλιοθήκη Poppler. Η poppler – utils παρέχει διάφορα εργαλεία για την επεξεργασία και εξαγωγή περιεχομένου από αρχεία PDF. Η εγκατάσταση της είναι ιδιαίτερα χρήσιμη όταν χρειάζεται να επεξεργαστούμε, να μετατρέψουμε ή να εξάγουμε δεδομένα από αρχεία PDF μέσω τερματικού ή σε εφαρμογές Python.
- Pillow. Η pillow είναι δημοφιλής βιβλιοθήκη επεξεργασίας εικόνων, διάδοχος της PIL (Python Imaging Library), που χρησιμοποιείται για άνοιγμα, επεξεργασία και αποθήκευση εικόνων διαφόρων τύπων (PNG, JPEG, BMP κ.λπ.).
- Base64. Η Base64 επιτρέπει την κωδικοποίηση και αποκωδικοποίηση δεδομένων σε Base64, χρήσιμη για τη μετατροπή εικόνων και αρχείων σε συμβολοσειρές (strings).

Για την εύρεση του πίνακα περιεχομένων στις δέκα πρώτες σελίδες του βιβλίου της γεωγραφίας και την εξαγωγή των τίτλων των κεφαλαίων και τον αριθμό των σελίδων του κάθε κεφαλαίου, με σκοπό τη δημιουργία των ερωταπαντήσεων, χρησιμοποιείται η βιβλιοθήκη PyPDF2 και το μοντέλο Gemini της Google, μέσω API key. Η PyPDF2 χρησιμοποιεί και τις δύο λειτουργίες της, την PdfReader, για να διαβάσει το PDF και την PdfWriter, για να εξάγει σε ένα PDF αρχείο τις δέκα πρώτες σελίδες. Έπειτα, διαμορφώνεται ένα prompt, σχήμα 3.2, για το μοντέλο Gemini "gemini-1.5-pro". Το μοντέλο καλείται με τη βιβλιοθήκη google.generativeai. Το prompt ζητά από το Gemini να διαβάσει το PDF των 10 σελίδων, να εξάγει τους τίτλους των κεφαλαίων και τις σελίδες έναρξης, να ελέγξει την ορθότητα των αποτελεσμάτων και να τα επιστρέψει σε μορφή JSON. Γι' αυτόν τον λόγο καλείται και η βιβλιοθήκη json. Το prompt παρέχει επίσης και ένα παράδειγμα JSON, για να καθοδηγήσει το Gemini. Αφού επιστραφεί η απάντηση, αποθηκεύεται και σε μορφή CSV, με τη βοήθεια της βιβλιοθήκης csv.

Στη συνέχεια, οι βιβλιοθήκες PyPDF2, JSON, CSV και google.generativeai, συνεχίζουν να αξιοποιούνται. Το αρχείο CSV περιέχει τους τίτλους των κεφαλαίων και τον αριθμό της πρώτης τους σελίδας αντίστοιχα. Με αυτά τα στοιχεία διασπάται το συνολικό αρχείο σε ξεχωριστά αρχεία PDFs, ένα για κάθε κεφάλαιο. Τα αρχεία αυτά αποστέλλονται πάλι στο μοντέλο "gemini-1.5-pro", για τη

δημιουργία ερωταπαντήσεων για κάθε κεφάλαιο, οι οποίες αποθηκεύονται σε μορφές JSON και CSV. Το prompt αυτής της περίπτωσης φαίνεται στο σχήμα 3.3.

Όπως αναφέρθηκε παραπάνω, με τη βιβλιοθήκη unstructured και τη λειτουργία partition_pdf, εξάχθηκαν το κείμενο, οι πίνακες και οι εικόνες από τα επιμέρους αρχεία PDF των κεφαλαίων, σε δομημένη μορφή για την αξιοποίησή τους από το σύστημα. Όλα καταχωρήθηκαν σε λίστες. Η λίστα των πινάκων προωθήθηκε στο μοντέλο "llama-3.3-70b-versatile" για την εξαγωγή περιγραφών από κάθε πίνακα. Οι περιγραφές αποθηκεύτηκαν σε αρχείο τύπου TXT. Δοκιμάστηκαν αρκετά μοντέλα, αλλά το "llama-3.3-70b-versatile" έδωσε τις καλύτερες περιγραφές. Το μοντέλο χρησιμοποιήθηκε μέσω του Groq, δημιουργώντας API key. Το σχήμα 3.4 δείχνει το prompt για τις περιγραφές των πινάκων, ενώ το σχήμα 3.5 δείχνει την κλήση του μοντέλου μέσω Groq.

Για τη χρησιμοποίηση του Groq απαιτείται η εγκατάσταση των πακέτων Langchain_community και Langchain_groq. Το Langchain_community είναι ένα υποσύνολο του Langchain που περιλαμβάνει εργαλεία και ενοποιήσεις που προέρχονται από την κοινότητα, επιτρέποντας τη χρήση εξωτερικών API και μοντέλων γλώσσας. Το Langchain_groq επιτρέπει την ενσωμάτωση των Groq AI μοντέλων στο Langchain. Έτσι, οι προγραμματιστές μπορούν να χρησιμοποιήσουν Groq AI μοντέλα μέσα σε Langchain εφαρμογές, επιτρέποντας τη βελτιωμένη επεξεργασία φυσικής γλώσσας (NLP). Τα παραπάνω, προϋποθέτουν την εγκατάσταση της βιβλιοθήκης Langchain, η οποία είναι ένα framework για την ανάπτυξη εφαρμογών που ενσωματώνουν μοντέλα γλώσσας. Ακόμα, τα πακέτα που χρησιμοποιήθηκαν από τη βιβλιοθήκη Langchain για δοκιμή και την υλοποίηση του συνολικού multimodal μοντέλου, είναι τα:

1. Importing Prompt Handling & Parsing (Εισαγωγή διαχείρισης προτροπών & ανάλυσης εξόδου)
   - ChatPromptTemplate: Ένα πρότυπο για τη διαμόρφωση προτροπών που χρησιμοποιούνται σε εφαρμογές συνομιλίας (chat-based). Βοηθάει στη δυναμική διαμόρφωση εισόδων του χρήστη.
   - StrOutputParser: Ένας αναλυτής που εξάγει απλό κείμενο από την έξοδο ενός LLM.
   - PromptTemplate: Ένα γενικό πρότυπο για τη δημιουργία προτροπών. Χρησιμοποιείται για τη δημιουργία δομημένων προτροπών με μεταβλητές.
   - BaseModel: Εισάγεται από την Pydantic v1, που χρησιμοποιείται στο Langchain για επικύρωση. Βοηθά στον ορισμό δομημένων σχημάτων για την εξαγωγή συγκεκριμένων δεδομένων.
2. Creating an Extraction Chain (Δημιουργία αλυσίδας εξαγωγής δεδομένων).
   - LLMChain. Η κλάση LLMChain εισάγεται από τη βιβλιοθήκη Langchain.chains και είναι ένα βασικό εργαλείο για τη δημιουργία αλυσιδών (chains) που χρησιμοποιούν μεγάλα γλωσσικά μοντέλα (LLMs). Επιτρέπει τη σύνδεση ενός LLM με μια προτροπή (prompt) και την εκτέλεση της, απλοποιώντας την αλληλεπίδραση με το μοντέλο και καθιστώντας την πιο οργανωμένη. Με απλά λόγια, παίρνει μια προτροπή, την στέλνει στο LLM και επιστρέφει το αποτέλεσμα.
   - Create_extraction_chain_pydantic. Μια συνάρτηση που εξάγει δομημένες πληροφορίες από μη δομημένο κείμενο χρησιμοποιώντας ένα μοντέλο Pydantic. Είναι χρήσιμο όταν χρειάζεστε οργανωμένες απαντήσεις αντί για ελεύθερο κείμενο.
   - Langchainhub. Ένα αποθετήριο που περιέχει προκαθορισμένα συστατικά, π.χ. προτροπές, αλυσίδες κτλ. που μπορούν να χρησιμοποιηθούν εύκολα σε εφαρμογές LangChain και χρησιμοποιείται για τη δημιουργία της σύνδεσης prompt και llm.
3. Working with Vector Stores (Εργασία με βάσεις δεδομένων διανυσμάτων).
   - Chroma. Η Chroma είναι βάση δεδομένων διανυσμάτων που αποθηκεύει ενσωματώσεις κειμένου και επιτρέπει την αναζήτηση ομοιότητας. Εισάγεται από τη βιβλιοθήκη langchain_community.vectorstores.

- SentenceTransformerEmbeddings: Ένα μοντέλο που μετατρέπει το κείμενο σε διανυσματικές αναπαραστάσεις (embeddings) χρησιμοποιώντας Sentence Transformers, καθιστώντας το αναζητήσιμο σε μια βάση διανυσμάτων.

4. Text Splitting and Chunking (Διαχωρισμός & κατάτμηση κειμένου).
   - SemanticChunker: Ένας έξυπνος διαχωριστής κειμένου που προσπαθεί να χωρίσει το κείμενο με βάση το σημασιολογικό νόημα και όχι απλώς τον αριθμό χαρακτήρων.
   - RecursiveCharacterTextSplitter: Ένας πιο βασικός διαχωριστής που διαχωρίζει το κείμενο σε μικρότερα τμήματα βάσει ενός καθορισμένου ορίου χαρακτήρων.
5. Storage and Document Handling (Αποθήκευση και διαχείριση εγγράφων).
   - Document: Μια τυπική μορφή που χρησιμοποιείται στο LangChain για τη διαχείριση εγγράφων κειμένου.
6. Multi-Vector Retrieval (Ανάκτηση πολλαπλών διανυσμάτων)
   - RunnableMap. Δημιουργεί έναν χάρτη διεργασιών όπου κάθε κλειδί αντιστοιχεί σε μια υπολογιστική μονάδα (runnable), διευκολύνοντας τη ροή δεδομένων.
   - RunnablePassthrough. Είναι ένα runnable που δεν αλλάζει τα δεδομένα που περνούν μέσα από αυτό, χρησιμεύοντας ως απλή γέφυρα στη ροή εργασιών.
   - StrOutputParser. Μετατρέπει την έξοδο του LLM σε απλό κείμενο (string), αφαιρώντας περιττές δομές.
   - MultiVectorRetriever: Ένας μηχανισμός ανάκτησης που αποθηκεύει πολλαπλές ενσωματώσεις (embeddings) ανά έγγραφο, για πιο ακριβή αναζήτηση. Αυτό είναι χρήσιμο όταν διαφορετικά τμήματα ενός εγγράφου πρέπει να ευρετηριαστούν ξεχωριστά.

```
def upload_to_gemini(path):
    prompt = """
    SYSTEM
    1. θέλω να διαβάσεις τα περιεχόμενα του βιβλίου {path}, συνήθως βρίσκονται στις πρώτες σελίδες,
    2. Να μου τα επιστρέψεις τους τίτλους των κεφάλαιον και την σελίδα στην οποία βρίσκονται.
    3. όταν τελειώσεις θέλω να ελέγξεις τα αποτελέσματα (πρίν να στείλεις το json αρχείο) για να δείς ότι είναι σωστά
    3. Επιστρέψτε το αποτέλεσμα σε μορφή JSON.
    EXAMPLE:
    Α΄ Ενότητα: Οι χάρτες. Ένα εργαλείο για τη μελέτη του κόσμου
    Ο χάρτης 10
    Τα είδη χαρτών 13
    Η ταυτότητα του χάρτη: Τίτλος και Υπόμνημα 16
    Η ταυτότητα του χάρτη: Κλίμακα 19
    Προσανατολισμός 22
    Β΄ Ενότητα: Το φυσικό περιβάλλον της Ελλάδας
    Η μορφή και το σχήμα της Ελλάδας 26
    Η θέση της Ελλάδας 29
    Οι ακτές της Ελλάδας 32
    Οι θάλασσες της Ελλάδας 35
    {
  "chapters": [
    {
      "title": "Α΄ Ενότητα: Οι χάρτες. Ένα εργαλείο για τη μελέτη του κόσμου",
      "starting_page": 9

    },
    {
      "title": "Ο χάρτης",
      "starting_page": 10
    },
    {
      "title": "Τα είδη χαρτών",
      "starting_page": 13
    },
    {
      "title": "Η ταυτότητα του χάρτη: Τίτλος και Υπόμνημα",
      "starting_page": 16

    }
      ]
    }
    το παραπάνω είναι ένα παράδειγμα εάν έχει και άλλες ενότητες και κεφάλαια θέλω να τα συμπεριλάβεις
    """
```

Σχήμα 3.2: Prompt για την εξαγωγή των 10 πρώτων σελίδων σε ένα νέο αρχείο PDF

```
prompt = """
SYSTEM
1. Ανάλυσε το κείμενο και εντόπισε όσο το δυνατόν περισσότερες ερωτήσεις με απαντήσεις που βασίζονται αποκλειστικά στο κείμενο,θέλω να καταγράψεις κάθε πληροφ
2. Δημιούργησε ερωτήσεις που καλύπτουν όλες τις πληροφορίες, για παράδειγμα  ορισμούς, τοποθεσίες, περιγραφές γεωγραφικών χαρακτηριστικών, πληροφορίες για τη
3. Όπου υπάρχουν πολλές πληροφορίες, προσπάθησε να δημιουργήσεις ερωτήσεις που συνοψίζουν τις κύριες ιδέες, όπως: "Τι γνωρίζεις για την τοπογραφία της Ελλάδας
4. Παρουσίασε τις ερωτήσεις και τις απαντήσεις σε μορφή JSON, με τον εξής τρόπο:
    "Τι είναι ένα δέλτα; Ένα δέλτα είναι μια περιοχή όπου ένας ποταμός εκβάλλει στη θάλασσα και δημιουργεί νησίδες από φερτές ύλες.",
    "Ποια είναι τα κύρια χαρακτηριστικά του μεσογειακού κλίματος; Το μεσογειακό κλίμα χαρακτηρίζεται από ήπιους, υγρούς χειμώνες και ζεστά, ξηρά καλοκαίρια."
7.Εάν υπάρχουν ερωτήσεις στο κείμενο θέλω να τις καταγράψεις και να προσπαθήσεις να τις απαντήσεις
8.όπου υπάρχει ομαδική δραστηριότητα θέλω την καταγράψεις , και να προτείνεις ιδέες για την υλοποίησή της
5. Η απάντηση πρέπει να είναι στα ελληνικά.
6. Επιστρέψτε το αποτέλεσμα σε μορφή JSON.
EXAMPLE:
"Τι είναι ένα δέλτα; Ένα δέλτα είναι μια περιοχή όπου ένας ποταμός εκβάλλει στη θάλασσα και δημιουργεί νησίδες από φερτές ύλες.",
"Ποια είναι τα κύρια χαρακτηριστικά του μεσογειακού κλίματος; Το μεσογειακό κλίμα χαρακτηρίζεται από ήπιους, υγρούς χειμώνες και ζεστά, ξηρά καλοκαίρια."
{
"chapters": [
{
  "Question": "Τι είναι ένα δέλτα;",
  "Answer": "Ένα δέλτα είναι μια περιοχή όπου ένας ποταμός εκβάλλει στη θάλασσα και δημιουργεί νησίδες από φερτές ύλες."
},
{
  "Question": "Ποια είναι τα κύρια χαρακτηριστικά του μεσογειακού κλίματος;",
  "Answer": "Το μεσογειακό κλίμα χαρακτηρίζεται από ήπιους, υγρούς χειμώνες και ζεστά, ξηρά καλοκαίρια."
}
]
}
"""
```

Σχήμα 3.3: Prompt για τη δημιουργία ερωταπαντήσεων από κάθε κεφάλαιο

```
prompt_text = """You are an assistant tasked with summarizing tables for retrieval. \
   These summaries will be embedded and used to retrieve the raw table elements. \
   Give a concise summary in greek, of the table that is well optimized for retrieval. Do not make a translation in english. Table {element} """

'''  """Είσαι ένας βοηθός επιφορτισμένος με τη δημιουργία συνοπτικών περιγραφών πινάκων (tables) για ανάκτηση. \
   Αυτές οι περιλήψεις θα ενσωματωθούν και θα χρησιμοποιηθούν για την ανάκτηση του αρχικού πίνακα (table). \
   Δώστε μια συνοπτική περιγραφή του πίνακα, βελτιστοποιημένη για ανάκτηση, στα ελληνικά.
   Αν δεν μπορείς να κάνεις την περιγραφή, απλά πες ότι δεν μπορείς να πειγράψεις τον πίνακα. Μην προσπαθήσεις να επινοήσεις μια απάντηση."""  '''

prompt = PromptTemplate.from_template(prompt_text)
```

Σχήμα 3.4: Prompt για τις περιγραφές των πινάκων

```
llm = ChatGroq(
    # model="mixtral-8x7b-32768", # Δίνει και 3η κατηγορία, τις αστικές πόλεις
    model="llama-3.3-70b-versatile", # Δίνει σωστά τις πόλεις
    # model="llama-3.2-11b-vision-preview", # Δίνει την Πάτρα ορεινή
    # model="llama-3.2-90b-vision-preview", # Δίνει σωστά τις πόλεις
    # model="llama3-groq-70b-8192-tool-use-preview", # Συντακτικά δίνει το καλύτερο αποτέλεσμα, αλλά καταχωρεί την Χαλκίδα στις ορεινές πόλεις
    # model="llama3-70b-8192", # 4 ορεινές (Θεσσαλονίκη, Ηράκλειο, Βόλος, Ιωάννινα)
    temperature=0,
    max_tokens=2048,
    timeout=None,
    max_retries=2,
)
```

Σχήμα 3.5: Κλήση του μοντέλου "llama-3.3-70b-versatile" μέσω Groq

Η λίστα με τις εικόνες χρησιμοποιήθηκε για την εξαγωγή περιγραφών των εικόνων, όμως προέκυψαν προβλήματα υπερχείλισης μνήμης. Έτσι οι εικόνες αποθηκεύτηκαν σε φακέλους, ανά κεφάλαιο με τη χρήση των εργαλείων PyMuPDF, Pillow και Base64. Έπειτα κλήθηκαν διάφορα vision μοντέλα από το Groq, για να περιγράψουν τις εικόνες, όπως τα llama-3.2-70b-preview και llama-3.2-90b-preview, με πολύ καλές περιγραφές. Οι δοκιμές έγινες σε επιμέρους κεφάλαιο του βιβλίου. Δυστυχώς, αν και οι δοκιμές στο επιμέρους κεφάλαιο ήταν ικανοποιητικές, το Groq σταμάτησε την παροχή χρήσης αυτών των μοντέλων για την παραγωγή περιγραφών. Έτσι αναγκαστήκαμε να στραφούμε σε άλλες λύσεις. Καταλήξαμε στο Gemini. Αρχικά χρησιμοποιήσαμε το “gemini-1.5-pro”, το οποίο έδινε πολύ καλές περιγραφές, αλλά είχε περιορισμένη χρήση tokens και request per day. Μετά από έρευνα, επιλέξαμε το “gemini-1.5-flash”, το οποίο έδωσε ικανοποιητικές περιγραφές, αρκετά απλές, αλλά παρείχε μεγαλύτερο αριθμό tokens και request per day. Τροφοδοτήσαμε όλες τις εικόνες στο μοντέλο. Για την

αποφυγή προβλημάτων με τα όρια του API key, δημιουργήσαμε περισσότερα, αλλά δε χρειάστηκαν. Ορίσαμε μια χρονοκαθυστέρηση από περιγραφή σε περιγραφή και δεν αντιμετωπίσαμε πρόβλημα. Βέβαια η διαδικασία των περιγραφών διήρκησε αρκετή ώρα. Το prompt για τις περιγραφές των εικόνων παρουσιάζεται στο σχήμα 3.6.

```
prompt = """
You are an assistant tasked with summarizing images for retrieval. \
These summaries will be embedded and used to retrieve the raw image. \
Give a concise summary of the image that is well optimized for retrieval."""
```

Σχήμα 3.6: Prompt για τις περιγραφές των εικόνων

Οι περιγραφές των εικόνων γίνονται στα αγγλικά, αποθηκεύονται σε αρχείο TXT, μεταφράζονται στα ελληνικά και αποθηκεύονται σε ξεχωριστό αρχείο ίδιου τύπου με το προηγούμενο. Για κάθε ένα αρχείο εικόνας, αποθηκεύεται το όνομα, η περιγραφή και το path της στο Google Drive. Η αποθήκευση του path συμβάλλει στη σύνδεση αρχείου εικόνας και url, έτσι ώστε να μπορεί να προβληθεί κατά την ανάκτηση. Η σύνδεση αυτή έγινε με τη βοήθεια Google API Key.

Τώρα όλα τα δεδομένα, κείμενο – περιγραφές πινάκων – περιγραφές εικόνων, είναι σε μορφή κειμένου και εισάγονται στη διαδικασία του χωρισμού σε Chunks, τη δημιουργία Embeddings, την αποθήκευση στη διανυσματική βάση ChromaDB και την ανάκτησή τους για την τροφοδοσία του LLM, με σκοπό την παραγωγή της απάντησης. Τα σχήματα 3.7 – 3.9 δείχνουν τμήματα του κειμένου, των περιγραφών των πινάκων και των περιγραφών των εικόνων αντίστοιχα.

```
NarrativeText

['«Με μια πυξίδα στρογγυλή κι ένα φθαρμένο χάρτη κίνησα μέσα στο πρωί μέχρι της Γης την άκρη. Γνώρισα χώρες και λαούς κι ήμουν στους ξένους ξένος, μα όταν
ξαναγύρισα ήμουν μ' αυτούς δεμένος».',
 'Μαρία Ταστσόγλου',
 'Ας υποθέσουμε ότι οι παραπάνω εικόνες είναι από τις χριστουγεννιάτικες διακοπές σου, όταν μαζί με τα παιδιά που γνώρισες από την Ιταλία αποφασίσατε να κάνετε
μια εκδρομή στο βουνό. Με ποιον τρόπο προσπαθήσατε να διαμορφώσετε το πρόγραμμα της πεζοπορίας σας, ενώ δε μι- λούσατε την ίδια γλώσσα;',
 'Το «εργαλείο» που θα χρησιμοποιήσετε, για να επικοινωνήσετε και να σχεδιάσετε την εκδρομή σας, είναι ο χάρτης. Κάθε χάρτης απεικονίζει τη μορφή που έχει ένα
μέρος της επιφάνειας της Γης, όσο μικρό ή μεγάλο κι αν είναι αυτό. Αναπαριστά την πραγματικότητα που υπάρχει γύρω μας, δηλαδή τις φυσικές ομορφιές και τα
ανθρώπινα δημιουργήματα. Η απεικόνιση γίνεται με διάφορα σύμβολα που όλοι οι άνθρωποι μπορούν να καταλάβουν.',
 'Είσαι μαθητής του Δημοτικού σχολείου της Βυτίνας, τοπία της οποίας δείχνουν οι παραπάνω εικόνες. Ποια από αυτές θα στείλεις στους νέους Ιταλούς φίλους σου,
ώστε να τους «μιλήσεις» για τον τόπο σου και τη γύρω περιοχή; Δικαιολόγησε την απάντησή σου.',
 'Όλες οι εικόνες απεικονίζουν ομορφιές της Βυτίνας, όμως μόνο μία από αυτές μπορεί να σε βοηθήσει να «επικοινωνήσεις» με τους φίλους σου. Είναι αυτή που
περιλαμβάνει πληροφορίες για τις φυσικές ομορφιές και τα ανθρώπινα έργα που υπάρχουν στην περιοχή.',
 'Παρατηρώντας τις παραπάνω εικόνες ας συζητήσουμε για τις αλλαγές που προκάλεσαν οι άνθρωποι στο τοπίο.',
 'Τα τοπία που υπάρχουν γύρω μας, καθώς περνούν τα χρόνια, αλλάζουν συνεχώς μορφή. Ένα μικρό βουνό μπορεί να εξαφανιστεί και στη θέση του να γίνει ένας μεγάλος
δρόμος, τα νερά της βροχής ίσως δημιουργήσουν έναν μικρό χείμαρρο, ένας ισχυρός σεισμός πιθανόν να ανοίξει ρήγμα ή να καταστρέψει ένα ανθρώπινο έργο, ένα
ηφαίστειο με τη λάβα του ίσως δημιουργήσει ένα μικρό νησί κ.ά. Όλες αυτές οι αλλαγές διαμορφώνουν την επιφάνεια της Γης και μπορούν να απεικονιστούν σε έναν
χάρτη. Επομένως ο χάρτης είναι το «εργαλείο» που μας βοηθά να μελε- τήσουμε τη μορφή ενός τοπίου.',
```

Σχήμα 3.7: Τμήμα κειμένου καταχωρημένο στη μεταβλητή NarrativeText

```
 'Κεφάλαιο 31ο: Κοινωνικοί & Οικονομικοί Χαρακτηρισμοί - Ανάγκη για καλύτερη ποιότητα ζωής, ανασφάλεια μικρών χωριών, έλλειψη πρόσβασης σε κέντρα υγείας,
ανεπτυγμένα αστικά κέντρα, βιομηχανίες & εταιρείες.',
 'Πίνακας 6: Μεγάλες πόλεις Ελλάδας - Παραθαλάσσιες: Αθήνα, Θεσσαλονίκη, Ηράκλειο, Βόλος, Πειραιάς, Καβάλα, Πάτρα, Χαλκίδα. Ορεινές: Ιωάννινα.',
 'Πίνακας 7: Πελοπόννησος, νησιά, τοπία και υγρότοποι Ελλάδας.',
 'Πίνακας 8: Περίληψη',
 'Περιέχει πληροφορίες για διάφορα θέματα, όπως:',
 '- Δελτία καιρού και ενετικά λιμάνια',
 '- Οροπέδια και ποταμοί',
 '- Λίμνες και υδροθάλασσες',
 '- Υδροδότηση και αρδευση',
 '- Τεχνητές λίμνες',
 '- Ζώα και πουλιά',
 '- Φυτά και θάμνοι',
 '- Περιβάλλον και εκπαίδευση',
 '- Γεωγραφικές περιοχές και τοποθεσίες',
 '- Βιβλιογραφικές αναφορές και ιστοσελίδες.',
```

Σχήμα 3.8: Τμήμα των περιγραφών των πινάκων

```
 ('page_1_img_2.jpeg',
  'Hillside πόλη, ασβεστωμένα κτίρια, βραχώδες έδαφος, βάρκες στο λιμάνι.',
  '/content/drive/MyDrive/Διπλωματική/Books/Geography_E_Class/Images_per_chapter/κεφάλαιο 10 Μεγάλα νησιωτικά συμπλέγματα και νησιά της Ελλάδας/
page_1_img_2.jpeg'),
 ('page_1_img_3.jpeg',
  'Μεγάλο, ανοιχτόχρωμο μεσογειακό στυλ κτίριο με πολλαπλές ιστορίες, μπαλκόνια, και ένα θολωτό τμήμα, που βρίσκεται σε μια προκυμαία περιπάτου με φοίνικες και
μικρότερα κτίρια γύρω από το ξενοδοχείο.',
  '/content/drive/MyDrive/Διπλωματική/Books/Geography_E_Class/Images_per_chapter/κεφάλαιο 10 Μεγάλα νησιωτικά συμπλέγματα και νησιά της Ελλάδας/
page_1_img_3.jpeg'),
 ('page_1_img_4.jpeg',
  'Ελληνικά νησιώτικα λιμάνια με ανεμόμυλο, ασβεστωμένα κτίρια και ήρεμη θάλασσα.',
  '/content/drive/MyDrive/Διπλωματική/Books/Geography_E_Class/Images_per_chapter/κεφάλαιο 10 Μεγάλα νησιωτικά συμπλέγματα και νησιά της Ελλάδας/
page_1_img_4.jpeg'),
 ('page_2_img_1.jpeg',
  'Χάρτης Ελληνικού Αιγαίου με περιφερειακά όρια και νησιά.',
  '/content/drive/MyDrive/Διπλωματική/Books/Geography_E_Class/Images_per_chapter/κεφάλαιο 10 Μεγάλα νησιωτικά συμπλέγματα και νησιά της Ελλάδας/
page_2_img_1.jpeg'),
```

Σχήμα 3.9: Τμήμα των περιγραφών των εικόνων

Για τη δημιουργία των Chunks, Embeddings και την αποθήκευση στη διανυσματική βάση ChromaDB ακολουθούνται τα εξής:

- Εισαγωγή της Chroma από τη βιβλιοθήκη langchain_community.vectorstores.
- Δημιουργία του μοντέλου ενσωμάτωσης (embeddings), χρησιμοποιώντας το SentenceTransformerEmbeddings με το μοντέλο "intfloat/multilingual-e5-base", το οποίο υποστηρίζει πολλές γλώσσες.
- Διαχωρισμός κειμένου (text_splitter) με το SemanticChunker, το οποίο χρησιμοποιεί σημασιολογική κατάτμηση βασισμένη στη στατιστική τυπική απόκλιση.
- Δημιουργία διανυσματικής βάσης δεδομένων Chroma από όλα τα έγγραφα (all_embeddings), αποθηκεύοντας τα ενσωματωμένα διανύσματα.
- Αποθήκευση της βάσης δεδομένων με την συνάρτηση persist(), ώστε να μπορεί να χρησιμοποιηθεί αργότερα, χωρίς να χρειάζεται να επαναϋπολογιστούν τα embeddings.

Τέλος, δημιουργείται το prompt μέσω της κλάσης PromptTemplate, σχήμα 3.10, το οποίο δίνει τις οδηγίες προς το μοντέλο. Αν δεν γνωρίζει την απάντηση, να το δηλώσει, ενώ αν η απάντηση περιέχει εικόνες, να δώσει πρώτα τη σύνοψη και μετά τα μονοπάτια των εικόνων. Μετά ακολουθεί η σύνδεση με το μοντέλο "llama-3.2-90b-vision-preview" μέσω Groq, σχήμα 3.11. Ορίζεται χαμηλή θερμοκρασία (0) για ακριβείς και σταθερές απαντήσεις και χρησιμοποιεί όριο tokens έως 2048. Όπως φαίνεται από το σχήμα δοκιμάστηκαν αρκετά μοντέλα. Πάρα πολύ καλές απαντήσεις είχαμε και από τα “llama-3.1-70b-versatile”, το οποίο καταργήθηκε από το Groq και από τον αντικαταστάτη του, “llama-3.3-70b-versatile”.

```
prompt_text = """Use the following piece of informations to answer the question. \
    If you don't know the answer just say you do not know. \
    If there is an image or more in the answer, first give the summary and after their paths. \
    Ερώτηση: {question}
    Απάντηση: """

prompt = PromptTemplate.from_template(prompt_text)
```

Σχήμα 3.10: Prompt για τη δημιουργία της τελικής απάντησης προς τον χρήστη

```
llm = ChatGroq(
    # model="mixtral-8x7b-32768", # Δίνει την καλύτερη απάντηση σε ερώτηση που αφορά πίνακα, αλλά στα αγγλικά. Δίνει και επιπλεόν πληροφορίες
    # model="llama-3.1-70b-versatile", Καταργήθηκε
    # model="llama-3.3-70b-versatile",
    model="llama-3.2-90b-vision-preview",
    # model="llama-3.2-11b-vision-preview",
    # model="llama3-70b-8192",
    # model="llama3-groq-70b-8192-tool-use-preview",
    temperature=0,
    max_tokens=2048,
    timeout=None,
    max_retries=2,
)
```

Σχήμα 3.11: Prompt για τη δημιουργία της τελικής απάντησης προς τον χρήστη

Κατά την ανάκτηση των σχετικών εγγράφων, ορίζεται ένας retriever που φέρνει τα 6 πιο σχετικά έγγραφα από το Chroma, με βάση την ερώτηση του χρήστη. Δημιουργείται η αλυσίδα RAG με τη χρήση του RunnableMap, για να περάσει το περιεχόμενο και την ερώτηση στο LLM. Το αποτέλεσμα περνάει από το StrOutputParser(), για να μετατραπεί σε απλό κείμενο. Η αλυσίδα εκτελείται με την εντολή invoke και επιτυγχάνεται το τελικό αποτέλεσμα, το LLM να επιστέψει την απάντηση στον χρήστη.

Κατά την υλοποίηση του μοντέλου χρησιμοποιήθηκαν και άλλες βιβλιοθήκες και κλάσεις, για διάφορες ή βοηθητικές εργασίες. Αυτές είναι:

- NLTK (Natural Language Toolkit). Χρησιμοποιείται για την επεξεργασία φυσικής γλώσσας (NLP) σε εφαρμογές όπως chatbots, ανάλυση συναισθήματος και εξαγωγή πληροφοριών από κείμενο. Το πακέτο punkt χρησιμοποιείται για tokenization, δηλαδή διαχωρισμό κειμένου σε λέξεις ή προτάσεις.
- BaseModel. Είναι μια κλάση που χρησιμοποιείται για ορισμό και επικύρωση δεδομένων με βάση προκαθορισμένα σχήματα (schemas) και για τον ορισμό δομών εισόδου και εξόδου σε μοντέλα και αλυσίδες διεργασιών. Παρέχει αυτόματο έλεγχο τύπων, μετατροπή δεδομένων και εύκολη διαχείριση δομών JSON.
- Sentence-transformers. Αποτελεί μια βιβλιοθήκη που επιτρέπει τη δημιουργία διανυσματικών αναπαραστάσεων προτάσεων, χρήσιμη για αναζήτηση σημασιολογικής ομοιότητας και επεξεργασία φυσικής γλώσσας (NLP).
- InMemoryStore: Ένα προσωρινό, αποθηκευτικό σύστημα στη μνήμη (cache), χρήσιμο για μικρά δεδομένα.

3.4.2.2 Διανυσματική Βάση Δεδομένων με Διάσπαση Κειμένου και Δημιουργία Σημασιολογικών Ενσωματώσεων (embeddings)

Η συγκεκριμένη προσέγγιση στοχεύει στην οργάνωση και διαχείριση των δεδομένων με τέτοιο τρόπο ώστε να διευκολύνει την ταχεία ανάκτηση σχετικών αποσπασμάτων βάσει σημασιολογικών ομοιοτήτων, υποστηρίζοντας εφαρμογές όπως τα chatbots και τα διαδραστικά quiz. Η διαδικασία ξεκινά με την εξαγωγή του κειμένου από το αρχικό PDF χρησιμοποιώντας εργαλεία όπως το pdfplumber. Το εξαγόμενο κείμενο επεξεργάζεται για την αφαίρεση ειδικών χαρακτήρων και την ομαλοποίηση του περιεχομένου, ώστε να είναι έτοιμο για περαιτέρω επεξεργασία. Στη συνέχεια, χρησιμοποιείται ο RecursiveCharacterTextSplitter για να διασπάσει το κείμενο σε μικρότερα τμήματα (chunks) περίπου 1300 χαρακτήρων, με επικάλυψη 200 χαρακτήρων. Αυτή η διαίρεση εξασφαλίζει ότι τα δεδομένα που θα υποβληθούν σε ανάλυση είναι σε κατάλληλο μέγεθος και διατηρούν το συνεκτικό νόημα τους.

Με τη βοήθεια ενός προηγμένου LLM, συγκεκριμένα του μοντέλου llama-3.1-70b-versatile μέσω του ChatGroq, το σύστημα μετατρέπει τα διαχωρισμένα chunks σε σαφείς, αυτόνομες προτάσεις. Το προσαρμοσμένο prompt δίνει οδηγίες για τη διάσπαση του κειμένου σε απλές, κατανοητές προτάσεις, που διατηρούν το αυθεντικό ύφος και περιέχουν πλήρεις πληροφορίες. Το αποτέλεσμα αυτής της διαδικασίας αποθηκεύεται σε μια λίστα προτάσεων, οι οποίες αποτελούν τα θεμέλια της βάσης δεδομένων μας.

Για να είναι δυνατή η αναζήτηση των προτάσεων με βάση το περιεχόμενο και όχι απλώς με λεξιλογική αντιστοιχία, κάθε πρόταση μετατρέπεται σε διανυσματική αναπαράσταση (embedding) χρησιμοποιώντας το μοντέλο SentenceTransformerEmbeddings (με το μοντέλο "intfloat/multilingual-e5-base"). Αυτή η μετατροπή επιτρέπει την αριθμητική αναπαράσταση της σημασίας του κειμένου, καθιστώντας δυνατή την εύρεση ομοιοτήτων μεταξύ διαφορετικών προτάσεων.

Τέλος, οι διανυσματικές αναπαραστάσεις αποθηκεύονται στη βάση δεδομένων Chroma, η οποία φιλοξενείται στην εφαρμογή. Η συγκεκριμένη βάση δεδομένων λειτουργεί ως εργαλείο retrieval, επιτρέποντας την ανάκτηση προτάσεων με βάση την σημασιολογική τους ομοιότητα με τα ερωτήματα

των χρηστών. Με αυτόν τον τρόπο, το σύστημα μπορεί να εντοπίζει γρήγορα και αποτελεσματικά τις πιο σχετικές πληροφορίες από το αρχικό κείμενο.

3.4.2.3 Δεύτερη υλοποίηση : Διανυσματική Βάση Δεδομένων με Εξαγωγή Ερωτοαπαντήσεων μέσω Προσαρμοσμένου Prompt

Στο επόμενο στάδιο της διπλωματικής, επεκτείνουμε την προσέγγιση της δημιουργίας διανυσματικών βάσεων δεδομένων, εστιάζοντας στην εξαγωγή ερωτοαπαντήσεων από το κείμενο. Σε αντίθεση με την προηγούμενη βάση, η οποία επικεντρώθηκε στη διάσπαση του κειμένου σε προτάσεις και τη δημιουργία σημασιολογικών ενσωματώσεων, το παρόν σύστημα στοχεύει στην αυτόματη παραγωγή ερωτήσεων με τις αντίστοιχες σωστές απαντήσεις.

Η διαδικασία ξεκινά με την εξαγωγή του κειμένου από το PDF χρησιμοποιώντας τη βιβλιοθήκη pdfplumber. Το κείμενο συλλέγεται σε μια ενιαία συμβολοσειρά και στη συνέχεια διασπάται σε τμήματα (chunks) με τη χρήση του RecursiveCharacterTextSplitter. Σε αυτή τη φάση, γίνεται χρήση προσαρμοσμένων παραμέτρων (όπως μεγαλύτερο overlap, σε σύγκριση με την προηγούμενη περίπτωση), προκειμένου να διασφαλιστεί η επαρκής κάλυψη του περιεχομένου για την εξαγωγή λεπτομερών ερωτοαπαντήσεων.

Το βασικό στοιχείο του συστήματος είναι το προσαρμοσμένο prompt, το οποίο καθοδηγεί το μοντέλο LLM (llama-3.1-70b-versatile μέσω του ChatGroq) να αναλύσει το εισαγόμενο κείμενο και να παράγει όσο το δυνατόν περισσότερες ερωτήσεις με απαντήσεις. Το prompt διαμορφώνεται έτσι ώστε να εστιάζει σε συγκεκριμένες πληροφορίες – όπως ονόματα, ημερομηνίες και γεγονότα – καθώς και σε συνοπτικές ερωτήσεις που αποτυπώνουν τις κύριες ιδέες του κειμένου. Η έξοδος του μοντέλου δίνεται σε μορφή JSON, εξασφαλίζοντας ότι το παραχθέν υλικό είναι δομημένο και έτοιμο για περαιτέρω επεξεργασία.

Μόλις παραχθούν οι ερωτοαπαντήσεις, το σύστημα προχωρά στη μετατροπή τους σε διανυσματικές αναπαραστάσεις χρησιμοποιώντας το μοντέλο SentenceTransformerEmbeddings (με το "intfloat/multilingual-e5-base"). Οι ενσωματώσεις αυτές, οι οποίες αποτυπώνουν τη σημασιολογική πληροφορία του περιεχομένου, αποθηκεύονται σε μια βάση δεδομένων Chroma. Η νέα αυτή βάση δεδομένων διαφέρει από την προηγούμενη όχι μόνο ως προς το περιεχόμενο – εστιάζοντας στις ερωτοαπαντήσεις αντί για απλές προτάσεις – αλλά και ως προς την εφαρμογή της, αφού θα χρησιμεύσει ως βάση για την παροχή διαδραστικού υλικού στο quiz και για την αποτελεσματική ανάκτηση πληροφοριών.

3.4.2.4 Ομαδοποίηση Εκτενέστερου Κειμένου για Βελτιστοποιημένη Ανάκτηση σύνθετων ερωτήσεων

Σε αυτό το στάδιο, υλοποιείται μια διανυσματική βάση δεδομένων που διαμορφώνεται από μικρότερα κομμάτια που εξάγονται από τα κεφάλαια του βιβλίου. Σε αντίθεση με τις δύο προηγούμενες βάσεις, όπου το κείμενο διασπάστηκε σε πιο σύντομες, απομονωμένες προτάσεις ή ερωτοαπαντήσεις, η παρούσα προσέγγιση επικεντρώνεται στην ομαδοποίηση μεγαλύτερων τμημάτων του κειμένου.

Η διαδικασία ξεκινά με την ανάγνωση του περιεχομένου από αρχεία κειμένου, τα οποία προέρχονται από τα κεφάλαια που έχουν εξαχθεί από το PDF. Χρησιμοποιείται ο RecursiveCharacterTextSplitter με καθορισμένους χαρακτήρες ως διαχωριστές (όπως οι αλλαγές παραγράφου ή οι τελείες) και αυξημένο overlap (500 χαρακτήρες), προκειμένου να διασφαλιστεί ότι κάθε κομμάτι θα είναι όσο το δυνατόν πιο περιεκτικό και νοηματικά ολοκληρωμένο.

Έπειτα, μέσω ενός προσαρμοσμένου prompt, το σύστημα καθοδηγεί το μοντέλο LLM (στην περίπτωση αυτή το llama-3.2-90b-vision-preview) να διασπάσει το κείμενο σε νοηματικά κομμάτια. Το prompt δίνει οδηγίες για:

- Την επεξεργασία του κειμένου ώστε να παραχθούν πολλαπλά κομμάτια που να διατηρούν πλήρες νόημα.
- Την παράθεση του αρχικού τίτλου του κεφαλαίου και τη δημιουργία ενός νέου, περιεκτικού τίτλου που περιγράφει συνοπτικά το περιεχόμενο του συγκεκριμένου κομματιού.
- Τη διατήρηση όλων των σχετικών λεπτομερειών ώστε να μην παραλείπεται καμία σημαντική πληροφορία.

Τα παραχθέντα κομμάτια, που περιέχουν τόσο τον αρχικό όσο και τον νέο τίτλο μαζί με το πλήρες περιεχόμενο, επεξεργάζονται περαιτέρω ώστε να δημιουργηθούν οι τελικές "ενσωματώσεις" (embeddings). Σε αυτή τη φάση, χρησιμοποιείται ένα μοντέλο ενσωματώσεων (SentenceTransformerEmbeddings) για να μετατραπούν τα κείμενα σε διανύσματα που αποτυπώνουν τη σημασιολογική τους πληροφορία.

3.4.2.5 Ομαδοποίηση μεγάλων νοηματικών κειμένων και εξαγωγή 10 σημαντικότερων λέξεων

Σε αυτό το κεφάλαιο παρουσιάζεται η υλοποίηση μιας διανυσματικής βάσης δεδομένων που πραγματοποιεί semantic chunking σε μεγαλύτερα κομμάτια του εγγράφου, με μέγεθος 2000 tokens και overlap 500. Από κάθε κεφάλαιο του βιβλίου εξάγονται οι 10 πιο σημαντικές λέξεις, οι οποίες αποθηκεύονται στα metadata μαζί με το όνομα του κεφαλαίου. Αυτή η ομαδοποίηση επιτρέπει την καταγραφή πιο συγκεντρωμένου και περιεκτικού περιεχομένου, διευκολύνοντας την ανάκτηση πληροφοριών σε πιο πολύπλοκες ερωτήσεις και ενισχύοντας την ακρίβεια της κατανόησης.

3.5 Ανάκτηση κειμένου και απόκριση μέσω LLM: Υλοποιήσεις σε Groq και Πανεπιστημιακό Διακομιστή

Η ανάκτηση απαντήσεων από μοντέλα LLM (Large Language Models) βασίζεται σε δύο βασικά στάδια: τον κατάλληλο σχεδιασμό του prompt, ώστε να διαμορφωθεί το σωστό context για το μοντέλο και την υλοποίηση του συστήματος που θα διαχειρίζεται τα ερωτήματα, θα ανακτά τις σχετικές πληροφορίες και θα επιστρέφει την απάντηση στον χρήστη.

3.5.1 Υλοποίηση σε Groq

3.5.1.1 Σχεδιασμός του Prompt

Το prompt είναι κρίσιμο για την ποιότητα της απάντησης που παράγει το μοντέλο. Στην παρούσα υλοποίηση, χρησιμοποιείται το ακόλουθο template:

“Χρησιμοποίησε τα ακόλουθα κομμάτια πληροφοριών (context) για να απαντήσεις στην ερώτηση στο τέλος. Αν δεν γνωρίζεις την απάντηση, απλά πες ότι δεν γνωρίζεις, μην προσπαθήσεις να επινοήσεις μια απάντηση. Η απάντηση να είναι στα ελληνικά.

Πληροφορίες:{context}

Ερώτηση: {question}

Η απάντηση να είναι στα ελληνικά:”

Αυτό το prompt εξασφαλίζει ότι:

- Το μοντέλο χρησιμοποιεί μόνο τις σχετικές πληροφορίες (context) που ανακτώνται.
- Σε περίπτωση έλλειψης δεδομένων, το μοντέλο αποφεύγει την επινόηση ψευδών απαντήσεων.
- Η απάντηση παρέχεται αποκλειστικά στα ελληνικά.

Το prompt ενσωματώνεται στο LangChain μέσω της κλάσης ChatPromptTemplate.

3.5.1.2 Υλοποίηση Συστήματος

Η υλοποίηση βασίζεται στην πλατφόρμα LangChain και το μοντέλο Groq. Το σύστημα περιλαμβάνει τις εξής λειτουργίες:

- Διαχείριση Embeddings: Χρήση του SentenceTransformerEmbeddings για τη μετατροπή κειμένων σε διανύσματα.
- Vectorstores: Αποθήκευση και ανάκτηση πληροφοριών μέσω του ChromaDB.
- Ανάκτηση Σχετικών Εγγράφων: Χρήση δύο vectorstores που ανακτούν σχετικές πληροφορίες.
- Διαχείριση Tokens: Υπολογισμός tokens για την αποφυγή υπέρβασης των ορίων του μοντέλου.
- Streaming Mode: Διαχείριση της απόκρισης σε τμηματική μορφή (chunks).
- Διαχείριση API Keys: Αυτόματη αλλαγή API κλειδιών όταν ανιχνεύονται σφάλματα 429 (Rate Limit).

Κατά την υποβολή μιας ερώτησης, το σύστημα:

- Ανακτά σχετικά έγγραφα από δύο ή παραπάνω vectorstores.
- Ενσωματώνει τα έγγραφα στο prompt.
- Στέλνει το ερώτημα στο Groq API.
- Επιστρέφει την απάντηση μέσω streaming.

3.5.1.3 Διαχείριση μέσω Flask API

Για την επικοινωνία με εξωτερικά συστήματα (π.χ., frontend εφαρμογές), το backend αναπτύχθηκε σε Flask. Η API υλοποίηση περιλαμβάνει:

- Τη διαχείριση διαφορετικών θεματικών ενοτήτων (π.χ., Γεωγραφία, Αριστοτέλης).
- Τη μετατροπή της ασύγχρονης ροής απαντήσεων σε σύγχρονη (μέσω event loop).
- Την παροχή των απαντήσεων στον χρήστη σε πραγματικό χρόνο.

Η συνδυασμένη χρήση του σωστού prompt, της αποτελεσματικής ανάκτησης εγγράφων και της υποστήριξης streaming mode, επιτρέπει τη δημιουργία ενός αποδοτικού συστήματος ανάκτησης απαντήσεων μέσω LLM. Η υποδομή αυτή μπορεί να επεκταθεί με επιπλέον vectorstores, fine – tuned μοντέλα ή εξειδικευμένη προ – επεξεργασία των δεδομένων.

3.5.2 Υλοποίηση σε Πανεπιστημιακό διακομιστή

Σε αυτήν την υποενότητα παρουσιάζουμε μια διαφορετική υλοποίηση του συστήματος ανάκτησης απαντήσεων, η οποία βασίζεται στην πλατφόρμα Ollama, εγκατεστημένη στον πανεπιστημιακό server. Αν και η βασική αρχιτεκτονική (αναζήτηση σε vectorstores, διαχωρισμός του κειμένου και χρήση προσαρμοσμένου prompt) παραμένει όμοια με την προσέγγιση του κεφαλαίου 4, η διαφορά έγκειται στην πλατφόρμα LLM που χρησιμοποιείται.

Τα κύρια χαρακτηριστικά της υλοποίησης είναι:

- Παρόμοιο Prompt: Το ίδιο προσαρμοσμένο prompt δίνει οδηγίες για την απάντηση με βάση τα σχετικά context κομμάτια. Ωστόσο, η ενσωμάτωση του Ollama επιτρέπει την αποστολή αιτημάτων μέσω του τοπικού server.
- Εναλλακτική Διαχείριση Context: Όπως και στην προηγούμενη υλοποίηση, τα έγγραφα ανακτώνται από δύο η περισσότερες vectorstores και συνδυάζονται εναλλάξ για να παραμείνουν εντός του ορίου των tokens (π.χ., 1500 tokens στην υλοποίηση αυτή).
- Ενσωματωμένη Λειτουργικότητα του Ollama: Το Ollama LLM, που λειτουργεί στον πανεπιστημιακό server, αποτελεί μια σύγχρονη πλατφόρμα για την εκτέλεση μεγάλων γλωσσικών μοντέλων σε τοπικό περιβάλλον, προσφέροντας υψηλή απόδοση και μειωμένη καθυστέρηση σε σύγκριση με απομακρυσμένα API. Με το Ollama, τα μοντέλα μπορούν να λειτουργούν απευθείας σε τοπικούς διακομιστές, επιτρέποντας στα αιτήματα να εξυπηρετούνται με μεγαλύτερη ταχύτητα και αξιοπιστία, γεγονός που είναι ιδιαίτερα χρήσιμο σε περιβάλλοντα όπου η σύνδεση στο διαδίκτυο είναι περιορισμένη ή η ζήτηση για επεξεργασία είναι υψηλή. Επιπλέον, η υποστήριξη λειτουργιών όπως το keep_alive επιτρέπει τη διατήρηση της σύνδεσης για μεγαλύτερα χρονικά διαστήματα, διευκολύνοντας τη διαχείριση streaming αιτημάτων σε πραγματικό χρόνο. Πάντα υπό την προϋπόθεση ότι διακομιστής και πελάτης θα βρίσκονται στο ίδιο τοπικό δίκτυο.
- Εφαρμογή μέσω Flask API: Η υλοποίηση αυτή παρέχει ένα API (μέσω Flask) που επιτρέπει την αποστολή ερωτήσεων και την επιστροφή απαντήσεων σε πραγματικό χρόνο, καθιστώντας την προσέγγιση αυτή εξαιρετικά κατάλληλη για εφαρμογές με frontend σε React.

Η νέα αυτή υλοποίηση με το Ollama παρουσιάζει μια διαφορετική προσέγγιση από την προηγούμενη, καθώς δεν βασίζεται πλέον στο Groq API αλλά αξιοποιεί τις δυνατότητες ενός LLM που είναι τοπικά εγκατεστημένο στον server του πανεπιστημίου.

3.5.3 Μηχανισμός ελέγχου

Σε αυτό το κεφάλαιο θα γίνει αναφορά σε έναν κρίσιμο για ολόκληρη την εφαρμογή μηχανισμό ο οποίος υλοποιείται στην μέθοδο get_answer_stream, η οποία είναι υπεύθυνη για την ανάκτηση απαντήσεων μέσω LLM με ιδιαίτερη έμφαση στην επιλογή από πολλαπλές βιβλιοθήκες (vectorstores) και στον ακριβή έλεγχο του αριθμού των tokens που θα σταλούν στο μοντέλο.

Η μέθοδος αυτή έχει τα εξής βήματα:

- Ανάκτηση Εγγράφων από Πολλαπλές Βιβλιοθήκες: Το σύστημα διαθέτει δύο η περισσότερους retrievers, ο καθένας από διαφορετικό vectorstore, που αντιπροσωπεύουν ξεχωριστές βιβλιοθήκες αποθηκευμένων πληροφοριών. Με αυτόν τον τρόπο, δίνεται η δυνατότητα επιλογής και ανάκτησης αποσπασμάτων από περισσότερες από μία πηγές, ενισχύοντας το context και την ποικιλία των δεδομένων.
- Διαχείριση του Context μέσω Token Limit: Για να αποφευχθεί η υπέρβαση των ορίων του μοντέλου, η μέθοδος format_docs_alternate συγκεντρώνει τα έγγραφα από τους δύο vectorstores εναλλάξ, υπολογίζοντας προσεκτικά τον αριθμό των tokens κάθε εγγράφου (μέσω της μεθόδου count_tokens). Η διαδικασία αυτή συνεχίζεται μέχρι ο συνολικός αριθμός tokens να φτάσει το καθορισμένο όριο (π.χ. 6000 tokens). Έτσι, το σύστημα εξασφαλίζει ότι αποστέλλεται στο LLM ένα πλήρες, αλλά και συγκεντρωμένο context, το οποίο περιέχει μόνο τις πιο σημαντικές πληροφορίες.
- Streaming Απαντήσεων και Διαχείριση Σφαλμάτων: Αφού δημιουργηθεί το context, το σύστημα δημιουργεί ένα λεξικό μηνυμάτων που περιέχει την ερώτηση και το συγκεντρωμένο context. Στη συνέχεια, μέσω της αλυσίδας (chain) που έχει οριστεί, το μοντέλο Groq αποστέλλει την απάντηση σε streaming mode, επιτρέποντας την διαδοχική επιστροφή τμημάτων της απάντησης. Σε περίπτωση που παρουσιαστούν σφάλματα όπως οι rate limits (429, 498), το σύστημα ενεργοποιεί αυτόματα μηχανισμό αλλαγής API key, διασφαλίζοντας τη συνέχεια της λειτουργίας.

Η συγκεκριμένη μέθοδος, με την επιλογή από πολλαπλές βιβλιοθήκες και την προσεκτική διαχείριση του αριθμού των tokens, επιτρέπει την ανάκτηση πολύπλοκων και λεπτομερών απαντήσεων από το μοντέλο. Αυτό είναι ιδιαίτερα σημαντικό όταν η ερώτηση απαιτεί την κατανόηση και τη σύνοψη ενός ευρέος φάσματος πληροφοριών, εξασφαλίζοντας ότι το μοντέλο ανταποκρίνεται με ακρίβεια και πληρότητα.

3.5.4 Υλοποίηση πράκτορα “Δάσκαλος” (Teacher agent)

Η παρούσα διπλωματική εργασία παρουσιάζει την υλοποίηση ενός "δασκάλου πράκτορα" (teacher agent), ο οποίος έχει σχεδιαστεί, ώστε να διεξάγει διαδραστικά κουίζ με ερωτήσεις που αφορούν τη γεωγραφία της πέμπτης τάξης του δημοτικού σχολείου, καθώς και το βιβλίο με θέμα τη ζωή και το έργο του Αριστοτέλη. Ο πράκτορας υλοποιείται σε περιβάλλον Flask και χρησιμοποιεί τεχνολογίες του LangChain σε συνδυασμό με το Groq LLM (Language Model) για την παραγωγή και αξιολόγηση των ερωτήσεων και απαντήσεων.

Οι βασικές λειτουργίες και η ροή εργασίας είναι:

- Εισαγωγή Στοιχείων Χρήστη και Επιλογή Κεφαλαίου. Ο πράκτορας ξεκινά με την απόκτηση των στοιχείων του χρήστη, δηλαδή το όνομα και τον αριθμό του κεφαλαίου που θα εξεταστεί. Για την εξαγωγή αυτών των πληροφοριών χρησιμοποιείται ένα LLM (μέσω του Groq) που επεξεργάζεται την απάντηση του χρήστη και επιστρέφει το όνομα και τον αριθμό του κεφαλαίου σε προκαθορισμένη μορφή. Εάν ο χρήστης δώσει λανθασμένες πληροφορίες (π.χ. κεφάλαιο εκτός ορίων), τότε ο πράκτορας ζητάει διόρθωση μέχρι να παρασχεθούν έγκυρα δεδομένα.
- Παρουσίαση Ερωτήσεων Κουίζ. Με βάση τα εισαγόμενα στοιχεία, ο πράκτορας φορτώνει ένα σύνολο ερωτήσεων (τα οποία έχουν προδιαγραφεί για το συγκεκριμένο κεφάλαιο). Κάθε ερώτηση περιλαμβάνει το περιεχόμενο της ερώτησης και τη σωστή απάντηση. Για την παραγωγή της τελικής διατύπωσης της ερώτησης και απάντησης χρησιμοποιείται μια απλή λειτουργία που διαχωρίζει το περιεχόμενο του κειμένου.
- Αξιολόγηση Απαντήσεων. Μετά την υποβολή της απάντησης από τον χρήστη, ο πράκτορας αξιολογεί την απάντηση αυτή χρησιμοποιώντας ένα LLM, το οποίο συγκρίνει την απάντηση του χρήστη με τη σωστή απάντηση και παράγει βαθμολογία (από 0 έως 10) καθώς και σύντομη επεξήγηση της αξιολόγησης.
- Διαχείριση Ροής με StateGraph. Η συνολική ροή του συστήματος υλοποιείται μέσω ενός state graph (StateGraph), όπου ο πράκτορας μεταβαίνει μεταξύ διαφόρων καταστάσεων (π.χ. awaiting_info, quiz, evaluate, completed). Κάθε κόμβος του γραφήματος αντιστοιχεί σε μία λειτουργία (όπως η εισαγωγή στοιχείων, η παρουσίαση ερωτήσεων ή η αξιολόγηση απαντήσεων), ενώ οι συνδέσεις μεταξύ των κόμβων καθορίζουν την προχωρημένη ροή του διαλόγου.

Ο "δασκάλος πράκτορας" που υλοποιήθηκε με το παραπάνω σύστημα παρουσιάζει μια λύση για την υποστήριξη της διαδικασίας αξιολόγησης και διδασκαλίας σε περιβάλλοντα εκπαίδευσης. Με τη χρήση της τεχνολογίας LLM (μέσω του Groq LLM σε συνδυασμό με το LangChain) και την υλοποίηση της ροής εργασίας με state graphs, το σύστημα επιτρέπει μια διαδραστική εμπειρία για τον χρήστη, παρέχοντας όχι μόνο τις ερωτήσεις αλλά και μια αυτόματη αξιολόγηση των απαντήσεων.

3.5.5 Οικονομικά και Τεχνικές Προκλήσεις στην Υλοποίηση της Εφαρμογής

Η ανάπτυξη της εφαρμογής αυτή χαρακτηρίστηκε από ελάχιστα οικονομικά έξοδα, με το μόνο άμεσο κόστος να αφορά την αγορά του domain, που ανήλθε περίπου στα 1,45 $ και τις ώρες εργασίας που δαπανήθηκαν στην ανάπτυξη και υλοποίηση του συστήματος. Μία από τις κύριες τεχνικές προκλήσεις που αντιμετωπίστηκαν ήταν η σύνδεση με τον server του πανεπιστημίου. Συγκεκριμένα, απαιτήθηκε η δημιουργία ενός Docker container που να εντάσσεται στο υποδίκτυο τόσο της εφαρμογής όσο και του

πανεπιστημίου. Επιπλέον, υπήρξε ιδιαίτερη δυσκολία στη ρύθμιση των servers ώστε να αποστέλλουν το κείμενο ανά tokens που λαμβάνουν – αντί για ολόκληρο το κείμενο – προκειμένου να μην δημιουργείται μεγάλο κενό μεταξύ της ερώτησης και της απάντησης.

Από αυτές τις προκλήσεις προκύπτουν πολύτιμα συμπεράσματα για τη βελτίωση της διαδικασίας ανάπτυξης συστημάτων ανάκτησης απαντήσεων μέσω LLM. Συγκεκριμένα, η λεπτομερής ρύθμιση της υποδομής και η διαχείριση των tokens αποτελούν κρίσιμα στοιχεία για την επίτευξη μιας ομαλής και αξιόπιστης λειτουργίας. Οι εμπειρίες αυτές αναδεικνύουν τη σημασία της προσαρμοσμένης τεχνικής προσέγγισης σε περιβάλλοντα με αυξημένες απαιτήσεις, συμβάλλοντας έτσι στη συνεχή βελτίωση της απόδοσης και της αξιοπιστίας των εφαρμογών βασισμένων σε LLM.

3.6 Επίλογος

Το chatbot αυτής της μελέτης χρησιμοποιεί την αρχιτεκτονική Retrieval-Augmented Generation (RAG) για την ανάκτηση και δημιουργία απαντήσεων, βασιζόμενο σε πέντε διανυσματικές βάσεις δεδομένων:

- Multimodal Βάση Δεδομένων – αποθηκεύει κείμενο, πίνακες και εικόνες για πιο πλούσια ανάκτηση δεδομένων.
- Βάση Δεδομένων για Προτάσεις – χωρίζει το κείμενο σε αυτόνομες προτάσεις για ακριβέστερη ανάκτηση.
- Βάση Δεδομένων Ερωτοαπαντήσεων – δημιουργεί σύνολα ερωτήσεων-απαντήσεων για διαδραστικά quiz.
- Βάση Δεδομένων Ομαδοποίησης Μεγαλύτερου Κειμένου – αποθηκεύει εκτεταμένα αποσπάσματα για σύνθετες ερωτήσεις.
- Βάση Δεδομένων Σημαντικών Λέξεων – εξάγει τις 10 πιο σημαντικές λέξεις από κάθε κεφάλαιο για ταχύτερη αναζήτηση.

Η επιλογή πληροφοριών προς το LLM γίνεται μέσω ενός μηχανισμού που ανακτά τα πιο σχετικά κομμάτια από κάθε βάση, τηρώντας τα όρια tokens και ρυθμίζοντας τη δυναμική συμμετοχή των βάσεων για βέλτιστη απάντηση.

Η εφαρμογή βασίζεται σε τοπικό server με modular αρχιτεκτονική μέσω Docker containers, στο Groq API, σε πανεπιστημιακή υποδομή για AI επεξεργασία και σε Cloudflare για ασφαλή συνδεσιμότητα.

Κεφάλαιο 4ο: Οδηγός Χρήσης και Λειτουργίας του Συστήματος Ερωταποκρίσεων με Ollama και LangChain

Η εφαρμογή αυτή έχει σχεδιαστεί για να επιτρέπει στους χρήστες να υποβάλλουν ερωτήσεις και να λαμβάνουν απαντήσεις από ένα σύστημα LLM που έχει υλοποιηθεί με τεχνολογία LangChain και χρήση Ollama και Groq api.

Η εφαρμογή βρίσκεται στη διεύθυνσή https://ab.aieducation.icu

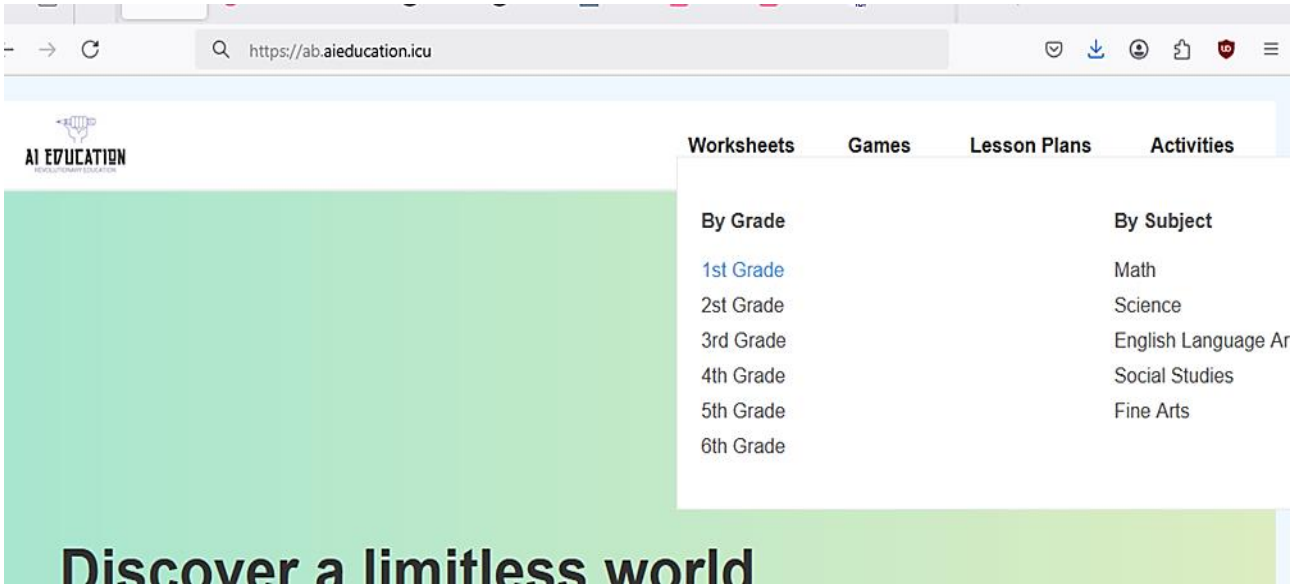
Η αρχική σελίδα της εφαρμογής σχεδιάστηκε με στόχο να προσφέρει μια φιλική και ελκυστική εμπειρία χρήστη, με έναν καθαρό και μοντέρνο σχεδιασμό. Έγινε προσπάθεια να διαμορφωθεί ένα περιβάλλον που να είναι εύκολα προσβάσιμο από οποιαδήποτε συσκευή, είτε πρόκειται για υπολογιστή, tablet ή κινητό τηλέφωνο.

Στο επάνω μέρος της σελίδας, υπάρχει μια μπάρα πλοήγησης που επιτρέπει γρήγορη πρόσβαση σε διαφορετικές ενότητες, όπως φύλλα εργασίας, παιχνίδια, εκπαιδευτικά σχέδια μαθημάτων και δραστηριότητες. Στο κεντρικό τμήμα, προβάλλεται ένα δυναμικό μήνυμα που καλεί τους χρήστες να ανακαλύψουν νέους τρόπους μάθησης.

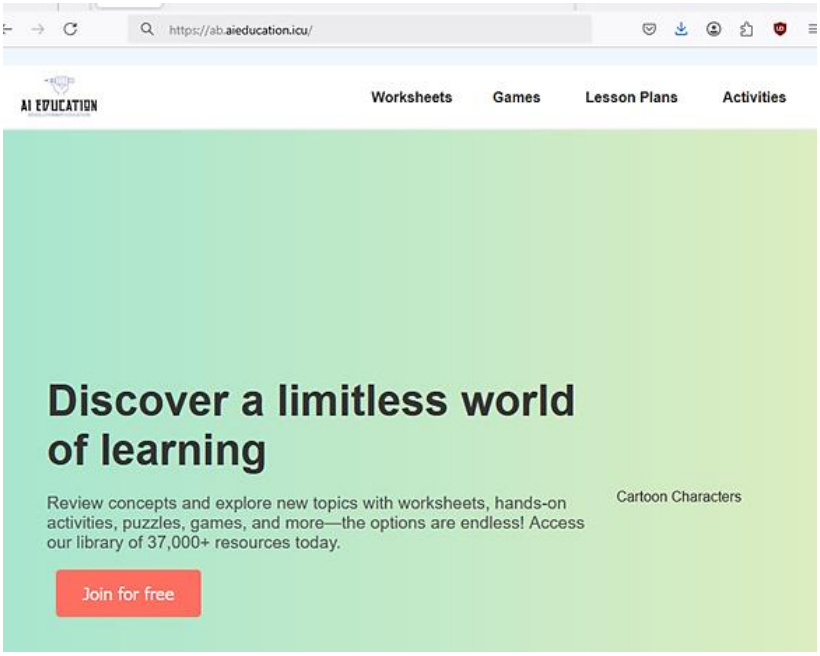
Επιπλέον, υπάρχει ένα εμφανές κουμπί εγγραφής, ώστε οι επισκέπτες να μπορούν εύκολα να ξεκινήσουν την περιήγησή τους και να αποκτήσουν πρόσβαση στο διαθέσιμο εκπαιδευτικό υλικό.

Οι χρήστες έχουν τη δυνατότητα να πλοηγηθούν στο εκπαιδευτικό περιεχόμενο με δύο τρόπους:

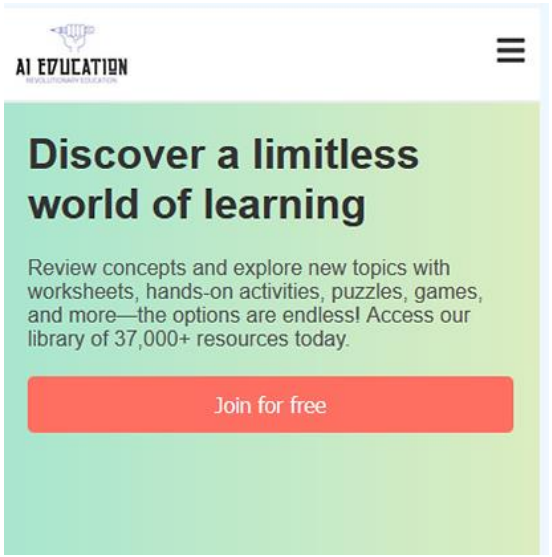
- Επιλέγοντας την τάξη τους από τη λίστα που εμφανίζεται κάτω από την κατηγορία "By Grade", σχήμα 4.1.
- Είτε μπορούν να πατήσουν το κουμπί "Join for free", σχήματα 4.2 – 4.3.



Σχήμα 4.1: Πλοήγηση στην εφαρμογή μέσω της λίστας της κατηγορίας "By Grade"



Σχήμα 4.2: Πλοήγηση στην εφαρμογή από υπολογιστή, μέσω του κουμπιού "Join for free"

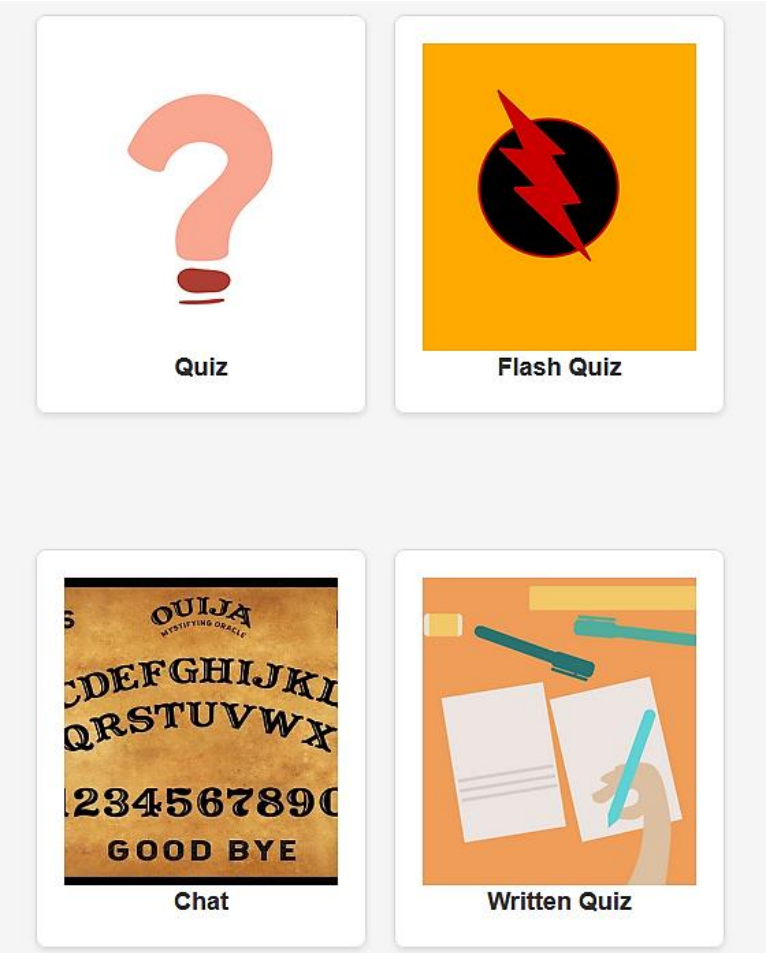


Σχήμα 4.3: Πλοήγηση στην εφαρμογή από κινητό τηλέφωνο, μέσω του κουμπιού "Join for free"

Στην επόμενη σελίδα, οι μαθητές μπορούν να επιλέξουν το μάθημα που τους ενδιαφέρει από τη λίστα των διαθέσιμων γνωστικών αντικειμένων, σχήμα 4.4, τα οποία είναι η Γεωγραφία της Ε' τάξης του δημοτικού σχολείου και η ζωή του Αριστοτέλη. Αφού επιλέξουν το μάθημα, έχουν τη δυνατότητα να διαλέξουν την δραστηριότητα που θέλουν να πραγματοποιήσουν, σχήμα 4.5.

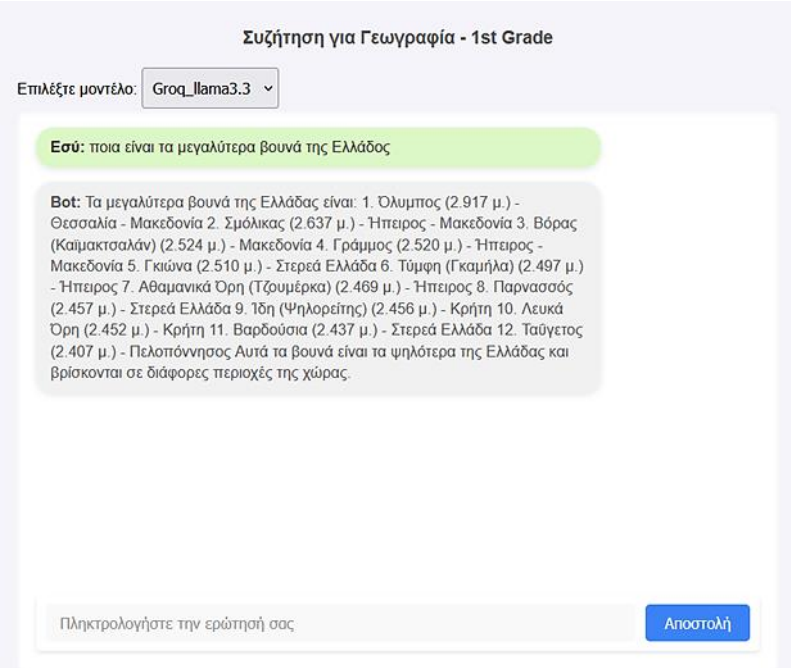
Εάν οι μαθητές επιλέξουν το "Quiz", θα μεταφερθούν σε μια νέα σελίδα όπου μπορούν να διαλέξουν το κεφάλαιο στο οποίο θέλουν να εξεταστούν. Το κουίζ θα ξεκινήσει αυτόματα, δίνοντάς τους τη δυνατότητα να απαντήσουν σε ερωτήσεις σχετικές με το επιλεγμένο μάθημα. Η διαδικασία αυτή βοηθά τους μαθητές να αξιολογήσουν τις γνώσεις τους και να εντοπίσουν τυχόν σημεία που χρειάζονται περισσότερη μελέτη.

Σχήμα 4.4: Λίστα των διαθέσιμων γνωστικών αντικειμένων

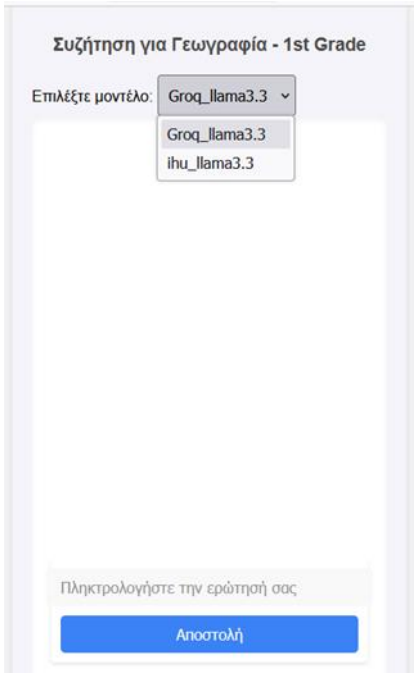


Σχήμα 4.5: Επιλογή δραστηριότητας

Εάν οι μαθητές επιλέξουν το chat σχήμα 4.6, θα μεταφερθούν στον αντίστοιχο χώρο, όπου θα μπορούν να επιλέξουν τόσο το μοντέλο με το οποίο θα συνομιλήσουν όσο και την υπηρεσία που θα χρησιμοποιηθεί, είτε μέσω του Groq είτε μέσω του server του πανεπιστημίου, σχήμα 4.7. Το μοντέλο στην υπηρεσία του Groq είναι το “llama-3.3-70B-versatile”, ενώ στην υπηρεσία του πανεπιστημίου είναι το “llama3.3:70B-instruct-q3_K_M”. Τέλος, επιλέγοντας το “Written Quiz” μεταβαίνουμε σε μια δραστηριότητα που υλοποιείται με τον πράκτορα “Δάσκαλο”, σχήμα 4.8.

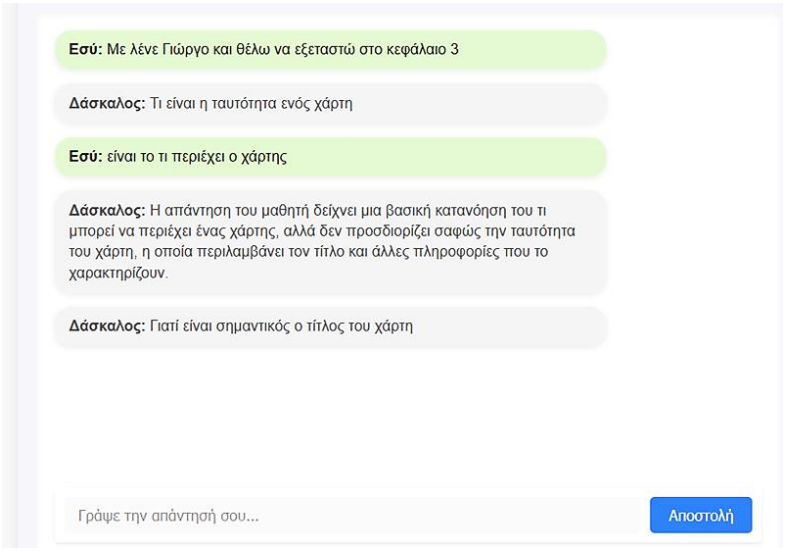


Σχήμα 4.6: Chat



Σχήμα 4.7: Επιλογή μοντέλου και υπηρεσίας

Με παρόμοιο τρόπο, μπορούν επίσης να εμβαθύνουν στο γνωστικό αντικείμενο του Αριστοτέλη.



Σχήμα 4.8: Written Quiz – Πράκτορας “Δάσκαλος”

Κεφάλαιο 5ο: Αξιολόγηση εφαρμογής

5.1 Ερωτηματολόγιο αξιολόγησης ευχρηστίας

Το προς αξιολόγηση σύστημα είναι ένα εκπαιδευτικό chatbot για το μάθημα της Γεωγραφίας της Ε' Τάξης του Δημοτικού Σχολείου και για τη ζωή του Αριστοτέλη.

Η αξιολόγηση πραγματοποιήθηκε με τη χρήση του διαδικτυακού λογισμικού διαχείρισης ερευνών και επεξεργασίας ερωτηματολογίων Google Form και την κλίμακα αξιολόγησης ευχρηστίας SUS (System Usability Scale). Το ερωτηματολόγιο βρίσκεται στην ηλεκτρονική διεύθυνση: https://docs.google.com/forms/d/1YkfOXsER8HIG9QLK-bwRSjSycSLnrtcWb7N3raZTgcg/edit. Η αρχική του σελίδα παρέχει πληροφορίες για το σκοπό της αξιολόγησης, τον δικτυακό τόπο και τις ερωτήσεις που ακολουθούν, οι οποίες αποτελούνται από 5 δημογραφικές και 10 ερωτήσεις που αφορούν τη διαδικασία SUS.

Τα δημογραφικά στοιχεία που ζητήθηκαν είναι:

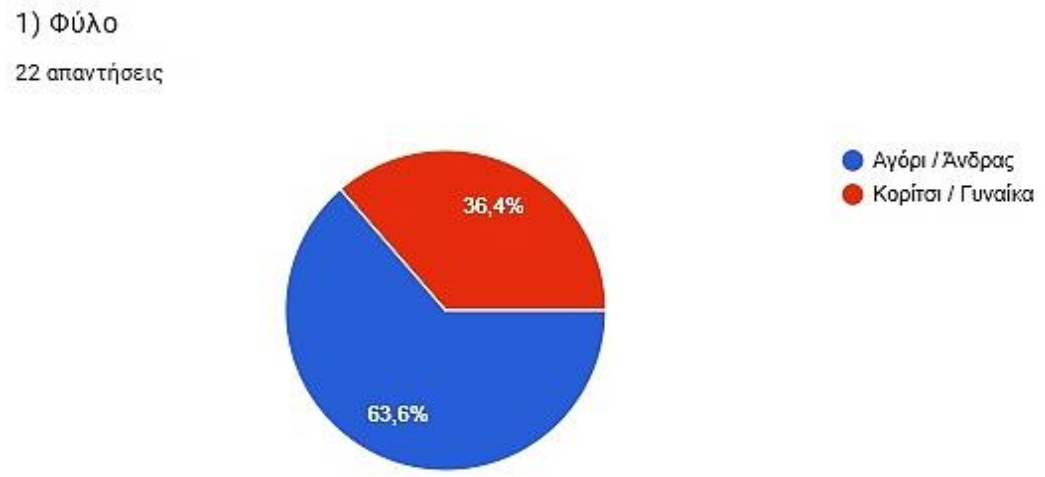
- Φύλο. Επιλογές: Αγόρι / Άνδρας, Κορίτσι / Γυναίκα, για τον εντοπισμό διαφορών λόγω φύλου.
- Ηλικία. Επιλογές: 6-9, 10-12, 13-18, 19-30, 31 και πάνω, για τον εντοπισμό διαφορών λόγω ηλικίας.
- Ιδιότητα. Επιλογές: Μαθητής/τρια, Απόφοιτος/η Δευτεροβάθμιας Εκπαίδευσης, Φοιτητής/τρια, Απόφοιτος/η Πανεπιστημίου, Κάτοχος Μεταπτυχιακού Διπλώματος, Κάτοχος Διδακτορικού Διπλώματος, για τον εντοπισμό διαφορών λόγω επιπέδου σπουδών.
- Συχνότητα χρήσης του διαδικτύου. Επιλογές: Σπάνια, Συχνά, Πολύ συχνά, Καθημερινά, για τον εντοπισμό διαφορών λόγω εμπειρίας χρήσης του διαδικτύου.
- Είναι η 1η φορά που επισκέπτεστε την ιστοσελίδα του chatbot; Επιλογές: Ναι / Όχι, για να διαπιστωθεί αν είναι πιο δύσκολη στη χρήση για όσους την επισκέπτονται για πρώτη φορά ή όχι.

Οι ερωτήσεις που χρησιμοποιήθηκαν για την αξιολόγηση της εφαρμογής με τη μετρική SUS είναι σε πενταβάθμια κλίμακα, ξεκινώντας από την επιλογή "Διαφωνώ απολύτως" και καταλήγοντας στην επιλογή "Συμφωνώ απολύτως" και είναι οι εξής:

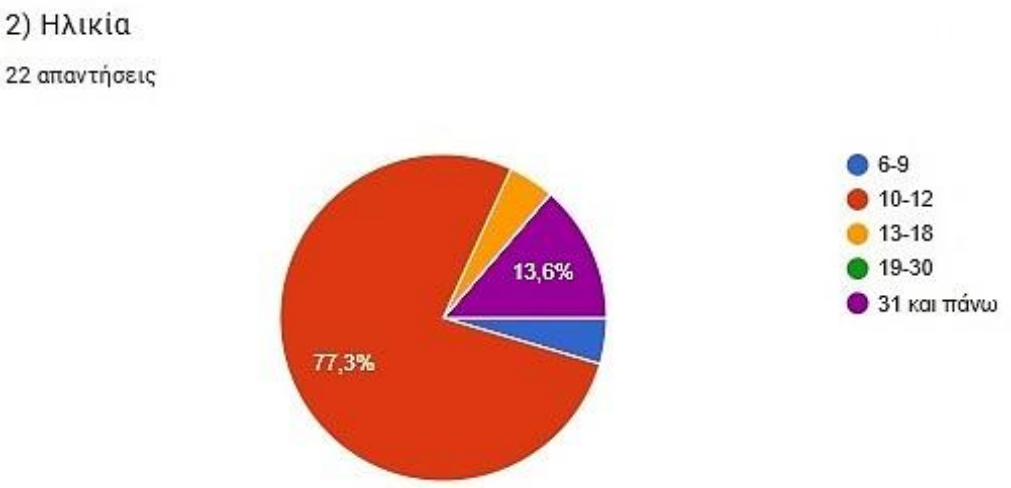
- Ερ. 6: Θα χρησιμοποιούσα συχνά αυτό το chatbot για τη Γεωγραφία.
- Ερ. 7: Βρήκα το chatbot αρκετά πολύπλοκο.
- Ερ. 8: Θεωρώ ότι το chatbot ήταν εύκολο στη χρήση.
- Ερ. 9: Νομίζω ότι θα χρειαστώ τη βοήθεια κάποιου ειδικού για να χρησιμοποιήσω το chatbot.
- Ερ. 10: Οι λειτουργίες του chatbot ήταν καλά ενσωματωμένες.
- Ερ. 11: Βρήκα πολλές ασυνέπειες στο chatbot.
- Ερ. 12: Πιστεύω ότι οι περισσότεροι μαθητές/εκπαιδευτικοί θα μπορούσαν να μάθουν γρήγορα πώς να το χρησιμοποιούν.
- Ερ. 13: Το chatbot ήταν δύσχρηστο.
- Ερ. 14: Είχα αυτοπεποίθηση χρησιμοποιώντας το chatbot.
- Ερ. 15: Χρειάζεται να γνωρίζω πολλά πριν μπορέσω να χρησιμοποιήσω το chatbot.

5.1.1 Γραφική αναπαράσταση των δημογραφικών στοιχείων και των ερωτήσεων SUS

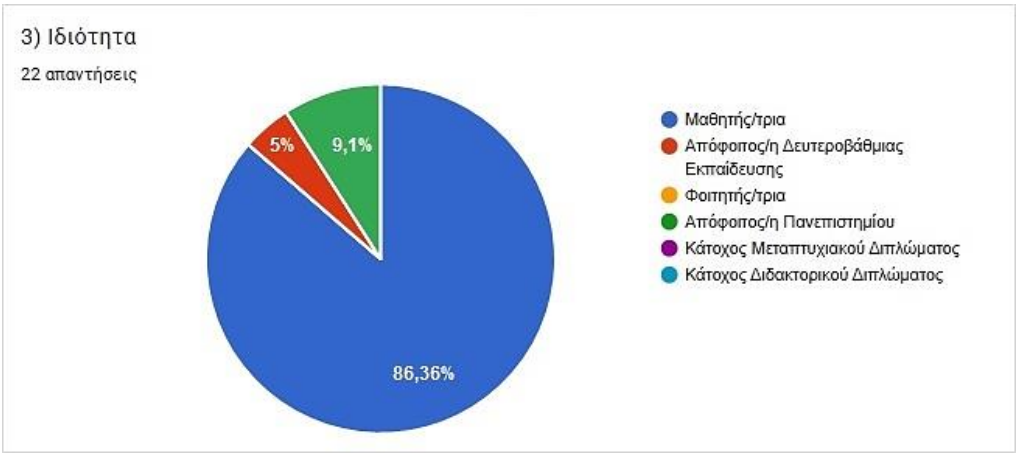
Οι απαντήσεις των δημογραφικών ερωτήσεων του ερωτηματολογίου παρουσιάζονται στα παρακάτω γραφήματα.



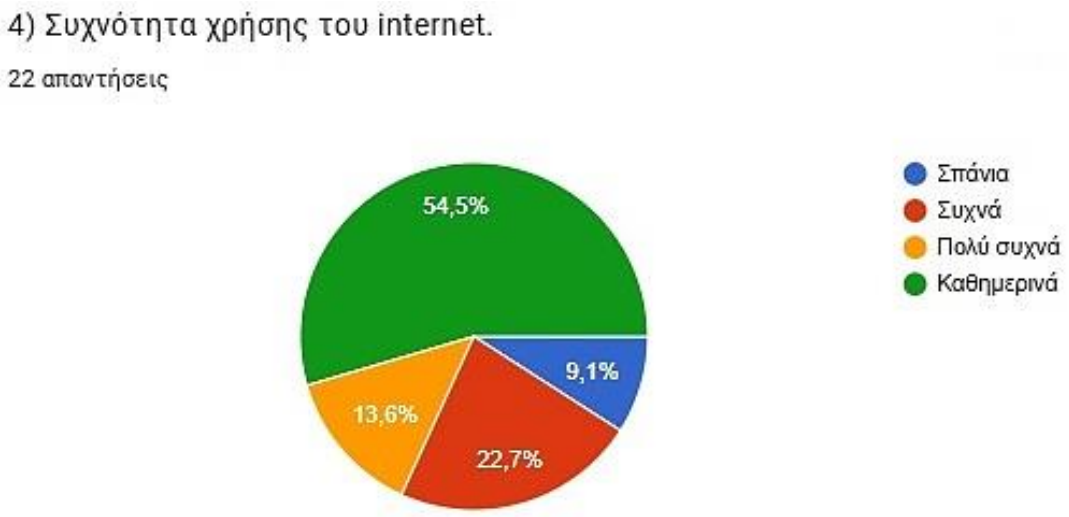
Σχήμα 5.1: Φύλο



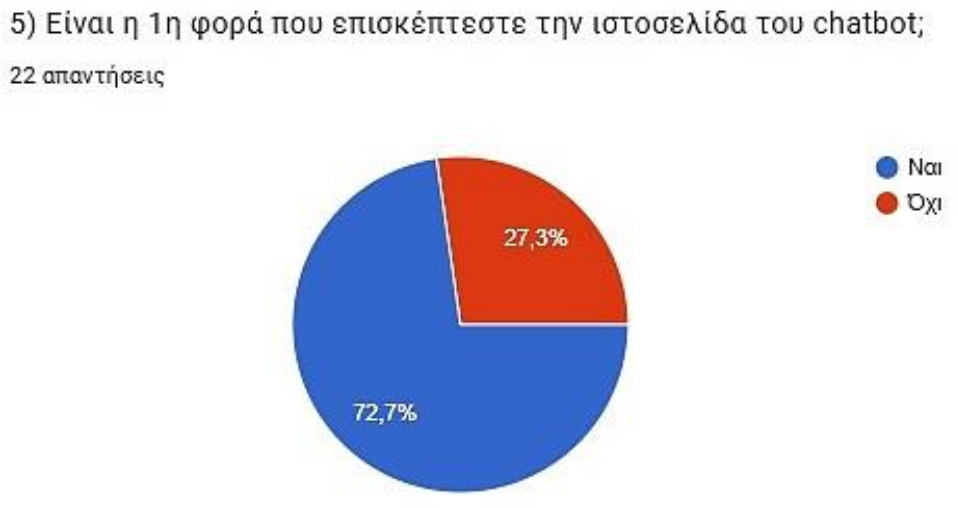
Σχήμα 5.2: Ηλικία



Σχήμα 5.3: Ιδιότητα



Σχήμα 5.4: Συχνότητα χρήσης του διαδικτύου



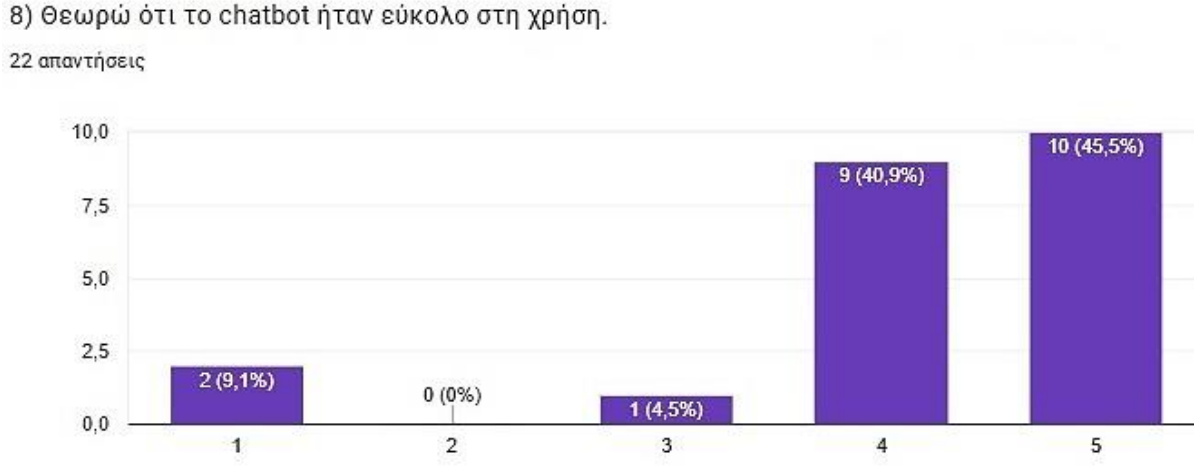
Σχήμα 5.5: Επίσκεψη ιστοσελίδας chatbot για 1η φορά

Παρατηρώντας τα γραφήματα, συμπεραίνουμε ότι η πλειοψηφία των χρηστών είναι μαθητές/τριες (86,36%), ηλικίας 10 – 12 (77,3%), που χρησιμοποιούν συχνά έως καθημερινά το διαδίκτυο (90,8%) και επισκέφθηκαν την ιστοσελίδα για 1η φορά (72,7%).

Τα αποτελέσματα των απαντήσεων στις ερωτήσεις που συμβάλλουν στην αξιολόγηση μέσω SUS παρουσιάζονται στα σχήματα 5.6 έως 5.15.

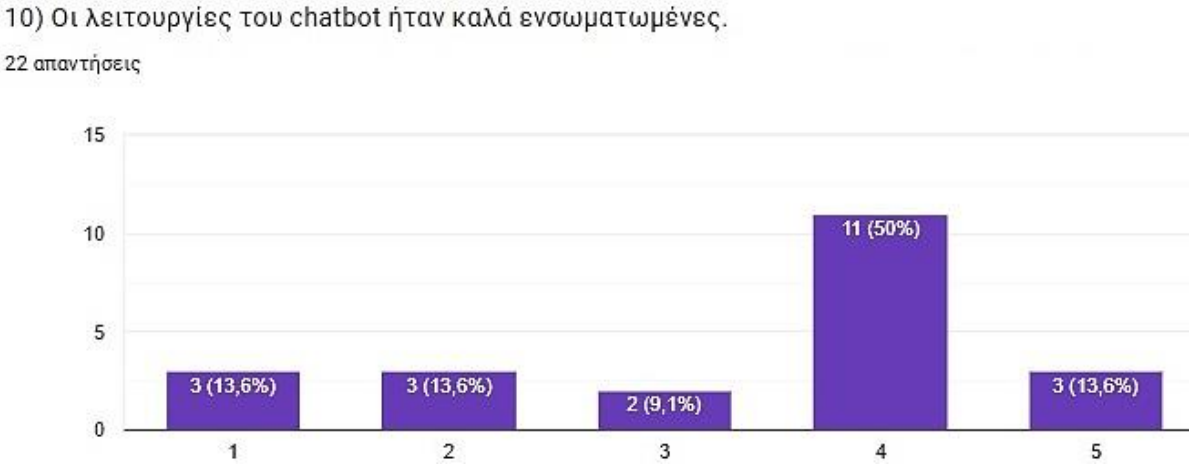
Σχήμα 5.6: Συχνότητα χρήσης chatbot

Σχήμα 5.7: Πολυπλοκότητα chatbot

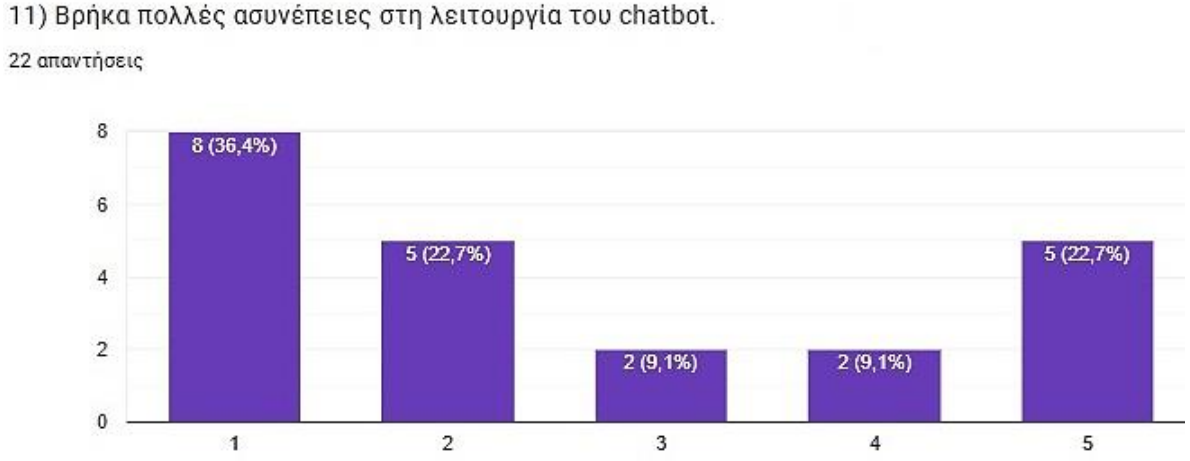


Σχήμα 5.8: Ευκολία χρήσης

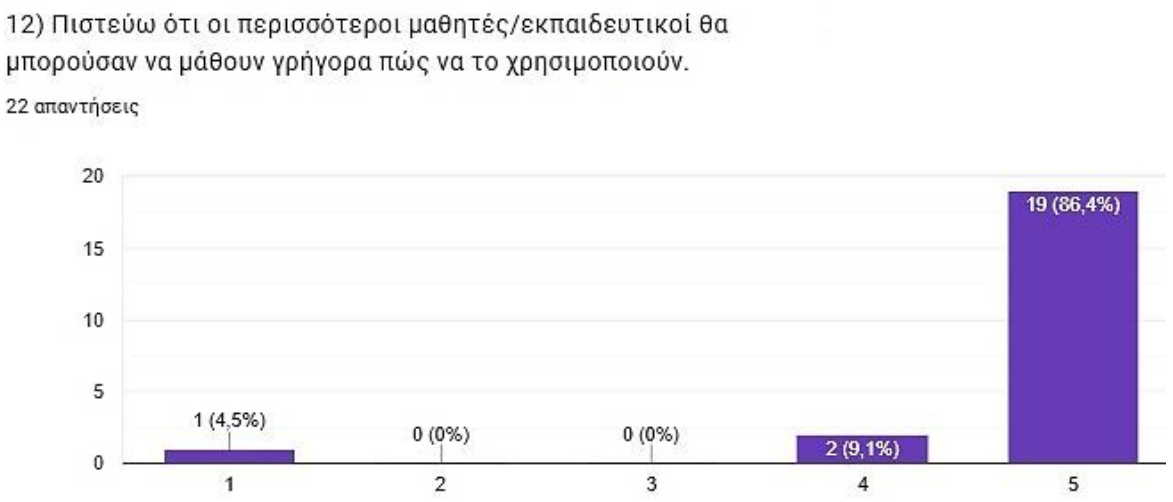
Σχήμα 5.9: Αναγκαιότητα βοήθειας για τη χρήση του chatbot



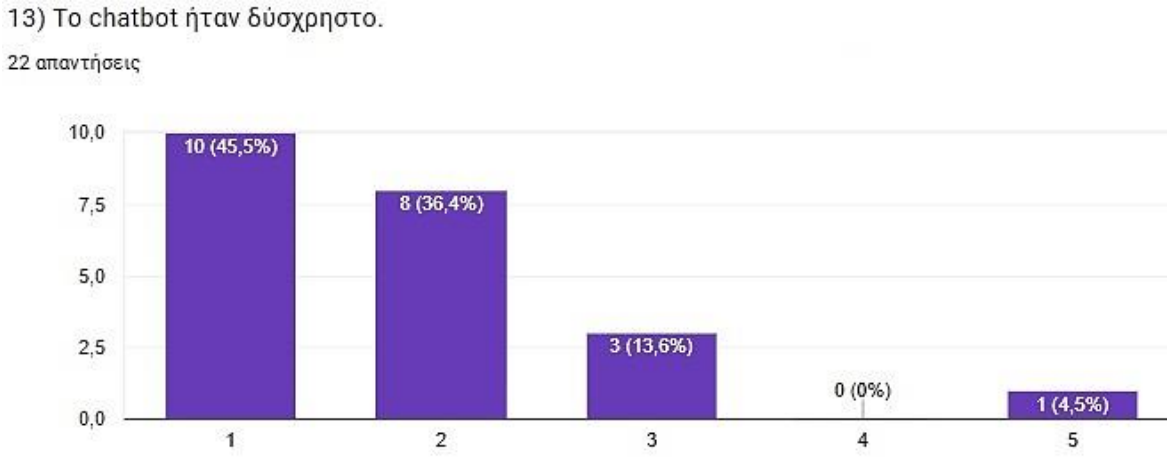
Σχήμα 5.10: Ενσωμάτωση λειτουργιών chatbot



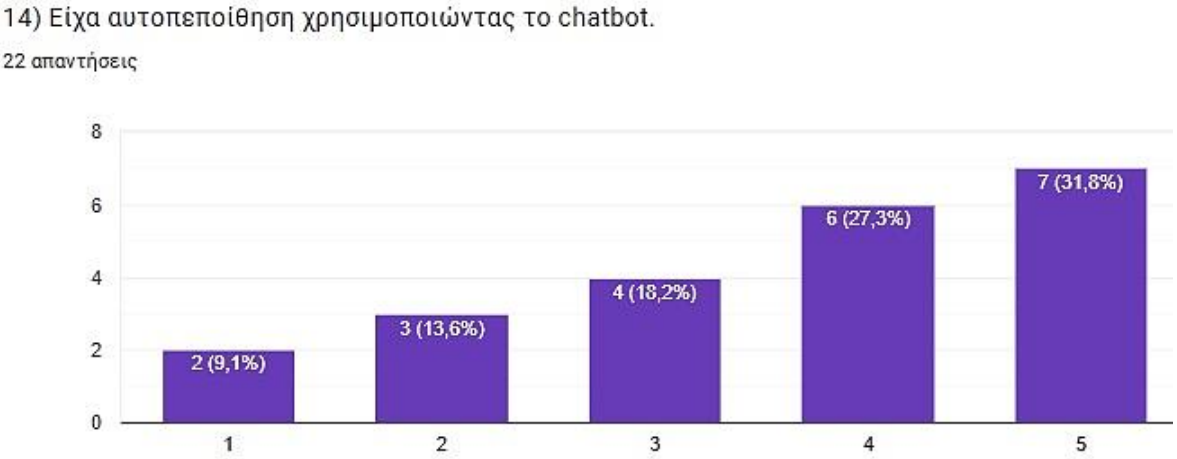
Σχήμα 5.11: Συνέπεια του chatbot



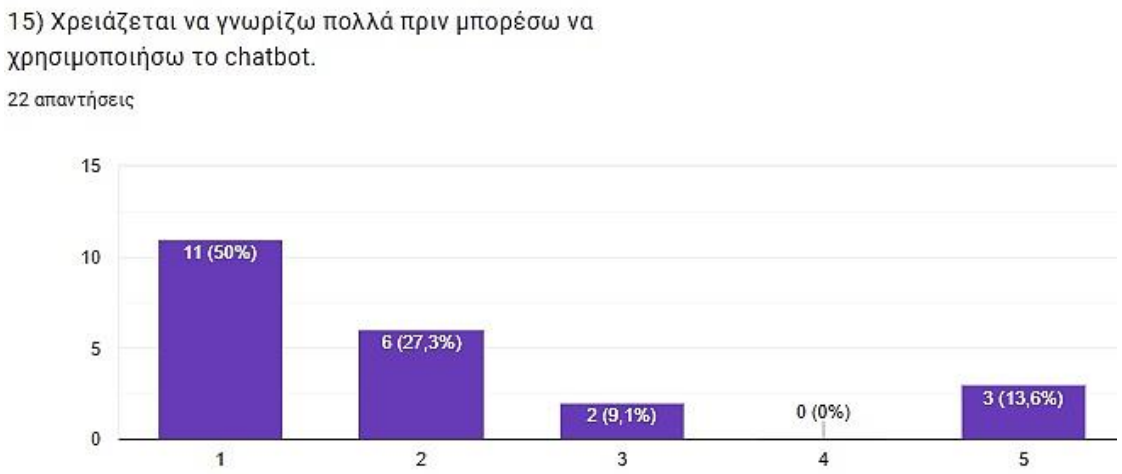
Σχήμα 5.12: Ταχύτητα χρήσης του chatbot



Σχήμα 5.13: Δυσκολία χρήσης του chatbot



Σχήμα 5.14: Αυτοπεποίθηση με τη χρήση του chatbot



Σχήμα 5.15: Επίπεδο γνώσεων πριν τη χρήση του chatbot

5.1.2 System Usability Scale (SUS)

Το SUS είναι μια μέθοδος αξιολόγησης της χρηστικότητας ενός συστήματος, που αναπτύχθηκε από τον John Brooke το 1986. Αποτελείται από 10 ερωτήσεις, τις οποίες οι χρήστες απαντούν χρησιμοποιώντας κλίμακα 5 σημείων, από "Διαφωνώ απολύτως" έως "Συμφωνώ απολύτως".

Το SUS παρέχει μια συνολική βαθμολογία χρηστικότητας (από 0 έως 100), η οποία χρησιμοποιείται για την εκτίμηση της εμπειρίας χρήστη σε συστήματα λογισμικού, ιστοσελίδες, εφαρμογές και άλλες διεπαφές.

Με αυτόν τον τρόπο, δύναται να αξιολογηθεί πόσο εύχρηστο είναι το chatbot για τους μαθητές και τους εκπαιδευτικούς. Μια βαθμολογία πάνω από 68 θεωρείται καλή και πάνω από 80 εξαιρετική, ενώ χαμηλότερες βαθμολογίες, όπως κάτω από 50, υποδεικνύουν προβλήματα στη χρηστικότητα και ότι πρέπει να γίνουν βελτιώσεις [48].

5.1.3 Υπολογισμός SUS score

Μετά τη συλλογή των απαντήσεων από τους χρήστες, υπολογίζεται η βαθμολογία της κάθε ερώτησης.

Για τις θετικές ερωτήσεις: 6, 8, 10, 12, 14 → Αφαίρεσε 1 από τη βαθμολογία. Για παράδειγμα αν κάποιος δώσει 4, τότε γίνεται: 4 - 1 = 3.

Για τις αρνητικές ερωτήσεις: 7, 9, 11, 13, 15 → Αφαίρεσε τη βαθμολογία από το 5. Για παράδειγμα αν κάποιος δώσει 4, τότε γίνεται: 5 - 4 = 1.

Ο υπολογισμός του συνολικού σκορ γίνεται ως εξής, πίνακας 5.1:

- Αθροίζονται οι νέες βαθμολογίες όλων των ερωτήσεων.
- Πολλαπλασιάζεται το σύνολο με το 2.5.
- Το αποτέλεσμα είναι η τελική βαθμολογία SUS από 0 έως 100.
- Αν έχουν απαντήσει πολλοί χρήστες, το συνολικό SUS είναι ο μέσος όρος.

Αν από κάθε SUS Score αφαιρεθεί η τυπική απόκλιση (standard deviation), αυτό θα δώσει μια εκτίμηση του κατώτερου ορίου της απόδοσης του chatbot, δηλαδή μια πιο συντηρητική εκτίμηση της χρηστικότητάς του. Η τυπική απόκλιση (σ) δείχνει πόσο διαφέρουν οι απαντήσεις μεταξύ τους. Έτσι, η διαφορά “SUS Score – σ” δείχνει ένα πιο "αυστηρό" SUS Score, υποθέτοντας ένα χειρότερο σενάριο όπου οι απαντήσεις είναι πιο αρνητικές. Αυτό είναι χρήσιμο αν θέλουμε να εξετάσουμε το χειρότερο

πιθανό σενάριο και να ελέγξουμε αν το chatbot παραμένει χρήσιμο ακόμα και αν οι απαντήσεις ήταν πιο αρνητικές από το μέσο όρο [48].

Πίνακας 5.1: SUS scores

	Ερ. 6	Ερ. 7	Ερ. 8	Ερ. 9	Ερ. 10	Ερ. 11	Ερ. 12	Ερ. 13	Ερ. 14	Ερ. 15	**SUS Score**
Απάντηση 1	4	2	3	3	3	2	4	2	3	3	**72,5**
Απάντηση 2	3	3	4	4	2	0	4	3	3	3	**72,5**
Απάντηση 3	3	4	3	4	3	3	4	3	1	3	**77,5**
Απάντηση 4	3	2	4	2	2	1	4	3	3	0	**60**
Απάντηση 5	0	4	0	4	0	4	0	0	0	4	**40**
Απάντηση 6	4	4	4	4	3	0	4	4	4	4	**87,5**
Απάντηση 7	3	3	3	3	3	3	3	3	4	0	**70**
Απάντηση 8	3	4	3	4	3	4	4	4	2	4	**87,5**
Απάντηση 9	1	4	3	4	4	4	4	4	3	4	**87,5**
Απάντηση 10	3	3	3	4	3	3	4	3	4	3	**82,5**
Απάντηση 11	3	3	4	4	3	4	4	4	2	4	**87,5**
Απάντηση 12	1	2	2	2	1	0	4	3	1	2	**45**
Απάντηση 13	4	4	0	2	0	1	4	2	0	0	**42,5**
Απάντηση 14	2	1	3	4	0	0	3	3	1	4	**52,5**
Απάντηση 15	2	4	4	4	3	3	4	4	4	3	**87,5**
Απάντηση 16	2	3	4	4	1	0	4	2	3	3	**65**
Απάντηση 17	4	2	3	4	3	2	4	3	2	2	**72,5**
Απάντηση 18	2	4	3	4	1	4	4	4	2	4	**80**
Απάντηση 19	4	4	4	4	4	4	4	4	4	4	**100**
Απάντηση 20	3	4	4	4	3	4	4	4	4	4	**95**
Απάντηση 21	4	4	4	4	4	4	4	4	4	4	**100**
Απάντηση 22	3	4	4	4	3	3	4	4	3	4	**90**

Τα SUS Scores των απαντήσεων κυμαίνονται από 40% έως 100%. Το SUS Score χρηστικότητας του chatbot είναι 75,23% που υποδεικνύει ένα πάρα πολύ καλό επίπεδο χρηστικότητας του chatbot, αποδεικνύοντας ότι είναι φιλικό προς τον χρήστη, ο οποίος πολύ πιθανόν να το χρησιμοποιήσει πάλι ή να το προτείνει σε άλλα άτομα. Αν και το μεγαλύτερο μέρος αυτών που απάντησαν ήταν μαθητές/τριες (86,36%), ηλικίας 10 – 12 ετών (77,3%), εν τούτοις η συχνή έως καθημερινή ασχολία τους με το διαδίκτυο, συνολικό ποσοστό 90,8%, τους κάνει ικανούς/ες να μπορούν να διαχειριστούν εύκολα το chatbot.

Υπολογίζοντας το πιο “αυστηρό” SUS Score, για να εξετάσουμε μια πιθανή χείριστη περίπτωση χρησιμότητας του chatbot, η τυπική απόκλιση υπολογίστηκες σε 17,58%, με τελικό SUS Score 57,65%. Το σκορ αυτό αντιπροσωπεύει ένα μέτριο επίπεδο χρηστικότητας, δηλώνοντας ότι το chatbot έχει προβληματάκια που πρέπει να λυθούν, ώστε να βελτιωθεί και να γίνει πιο εύχρηστο, απλό και κατανοητό στους χρήστες.

Κεφάλαιο 6ο: Συμπεράσματα – Μελλοντική εργασία

Η παρούσα εργασία παρουσιάζει την ανάπτυξη ενός προηγμένου συστήματος chatbot, σχεδιασμένου για την υποστήριξη μαθητών, μέσω της αξιοποίησης Μεγάλων Γλωσσικών Μοντέλων και της αρχιτεκτονικής Retrieval-Augmented Generation, για τη διαχείριση μεγάλων και πολύπλοκων εγγράφων. Η αξιοποίηση των LLMs σε συνδυασμό με τις μεθόδους αναζήτησης βασισμένες σε διανύσματα, ενσωματώνοντας την προσέγγιση RAG, επιτρέπει την ανάκτηση πληροφοριών από αξιόπιστες βάσεις γνώσεων, με τα δεδομένα να αποθηκεύονται ως διανυσματικές αναπαραστάσεις μέσω ChromaDB. Το chatbot αναζητά σχετικά έγγραφα από τη διανυσματική βάση δεδομένων, ενώ παράλληλα αναλύει τις ερωτήσεις των χρηστών, παρέχοντας ακριβείς και τεκμηριωμένες απαντήσεις.

Η αρχιτεκτονική RAG βελτιώνει την ικανότητα απόκρισης του chatbot, καθώς υπερβαίνει τους περιορισμούς των παραδοσιακών chatbots που βασίζονται σε στατικά μοντέλα ή προκαθορισμένους κανόνες. Το RAG επιτρέπει στο chatbot να ανακτά δυναμικά σχετικό και ενημερωμένο περιεχόμενο από τη γνωσιακή βάση, παρέχοντας πιο λεπτομερείς και αξιόπιστες πληροφορίες, γεγονός που ενισχύει την εμπιστοσύνη των χρηστών.

Η πλατφόρμα που αναπτύχθηκε είναι ευέλικτη και αρθρωτή, αξιοποιώντας σύγχρονες τεχνολογίες containerization και AI για την παροχή εκπαιδευτικών υπηρεσιών. Η συνδυασμένη χρήση τεχνητής νοημοσύνης, ασφάλειας και επεκτασιμότητας δημιουργεί μια υποδομή ικανή να προσαρμόζεται σε διαφορετικές ανάγκες και απαιτήσεις.

Επιπλέον, η ενσωμάτωση πρακτόρων (agents) στο chatbot προσφέρει νέα επίπεδα λειτουργικότητας και αποτελεσματικότητας. Οι πράκτορες λειτουργούν ως ευφυείς μεσάζοντες, ενορχηστρώνοντας την ενσωμάτωση δεδομένων από εξωτερικές πηγές όπως αρχεία PDF, CSV, βίντεο YouTube και ιστοσελίδες. Μέσω αυτής της δομής, το chatbot μπορεί να προσαρμόζεται δυναμικά στο περιεχόμενο των ερωτήσεων των χρηστών, βελτιώνοντας την ακρίβεια και την ποιότητα των απαντήσεών του.

Η ανάπτυξη του συστήματος αντιμετώπισε προκλήσεις, καθώς η δημιουργία ενός επιτυχημένου εκπαιδευτικού chatbot απαιτεί προσεκτική σχεδίαση των διαδικασιών RAG, προσαρμογή των LLMs και βελτιστοποίηση των προτροπών, ώστε να διασφαλίζεται η συνάφεια και η ακρίβεια της εκπαιδευτικής πληροφορίας. Παράλληλα, είναι απαραίτητη η τήρηση των δικαιωμάτων πρόσβασης σε έγγραφα, η παροχή συνοπτικών και κατανοητών απαντήσεων, η ενσωμάτωση σχετικών αναφορών και η προστασία των προσωπικών δεδομένων. Αυτές οι απαιτήσεις καθιστούν αναγκαία τη συστηματική αξιολόγηση και τις συνεχείς επαναλήψεις στη διαδικασία ανάπτυξης.

Μελλοντικές βελτιώσεις θα μπορούσαν να περιλαμβάνουν:

- Βελτιστοποίηση του χρόνου απόκρισης μέσω caching τεχνικών για τη μείωση του inference latency.
- Δυναμική διαχείριση των containers μέσω Kubernetes, επιτρέποντας αυτόματη κλιμάκωση των υπηρεσιών.
- Επέκταση της βάσης γνώσης με εξειδικευμένα datasets για τη βελτίωση της ακρίβειας των απαντήσεων.
- Ενσωμάτωση δεδομένων σε πραγματικό χρόνο, ανάπτυξη δυνατοτήτων σύνδεσης και μνήμης, καθώς και λεπτομερή έλεγχο της αλυσίδας ανάκτησης.

Συνολικά, το chatbot αξιοποιεί τις δυνατότητες του RAG και των LLMs για να προσφέρει εξατομικευμένη καθοδήγηση στους μαθητές, ενισχύοντας τη διαδικασία μάθησης. Με τη συνεχή εξέλιξη των AI τεχνολογιών, το σύστημα αυτό μπορεί να μεταμορφώσει τον τρόπο με τον οποίο οι μαθητές διαχειρίζονται την εκπαιδευτική τους εμπειρία, παρέχοντας αξιόπιστη, εξατομικευμένη και συνεχώς βελτιωμένη υποστήριξη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Vani Bhat et al., “Retrieval Augmented Generation (RAG) Based Restaurant Chatbot with AI Testability”, 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService), IEEE, Shanghai, China, Jul. 2024, pp. 1-4.

[2] Beyza Belen İrican et al., “QBot Domain – Specific Chatbots with Retrieval – Augmented Generation and Vector Embedding for Complex Documentation Queries”, Innovations in Intelligent Systems and Applications Conference (ASYU), Ankara, Turkiye, Oct. 2024, pp. 1-2.

[3] Akkiraju, Rama et al., “FACTS About Building Retrieval Augmented Generation-based Chatbots”, arXiv preprint arXiv:2407.07858, 2024, pp.1-2.

[4] Anh Nguyen Thi Dieu, Hien T. Nguyen, and Chien Ta Duy Cong, “The enhanced context for AI-generated learning advisors with Advanced RAG”, 18th International Conference on Advanced Computing and Analytics (ACOMPA), IEEE, Ben Cat, Vietnam, Nov. 2024, pp. 94-98.

[5] Kannan Hemachandran et al., “Artificial Intelligence: A Universal Virtual Tool to Augment Tutoring in Higher Education”, *Computational Intelligence and Neuroscience*, vol. 22, pp. 2-3, Jan 2022.

[6] Bashaer Alsafari, Eric Atwell, Aisha Walker, and Martin Callaghan, “Towards effective teaching assistants: From intent-based chatbots to LLM-powered teaching assistants”, Natural Language Processing Journal, vol. 8, 2024, pp 1-4.

[7] Pokrivčáková Silvia, “Preparing teachers for the application of AI-powered technologies in foreign language education” *Journal of Language and Cultural Education*, vol. 7, no. 2, 2019, pp. 135–138

[8] Sundar, Koushik, et al. “Revolutionizing assessment: AI-powered evaluation with RAG and LLM technologies”, Proceedings of the 2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), IEEE, 2024, pp. 43–46.

[9] R. Leer and S. Ivanov, “Rethinking the future of learning: the possibilities and limitations of technology in education in the21 st century,” The International Journal of Oral Implantology, vol. 5, no. 4, 2013.

[10] A. A. Pise, H. Vadapalli, and I. Sanders, “Relational reasoning using neural networks: a survey,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 29, 2021.

[11] Neyem A. et al., “Towards an AI Knowledge Assistant for Context-aware Learning Experiences in Software Capstone Project Development”, IEEE, Transactions on Learning Technologies, May 2024.

[12] Dieu, Nguyen Thi, and Thao Thi Nguyen. “The enhanced context for AI-generated learning advisors with Advanced RAG.” Proceedings of the 2024 International Conference on Advanced Computing and Applications (ACOMP), IEEE, 2024, pp. 94-97.

[13] Martinez-Araneda et al., “Designing a Chatbot to support problem-solving in a programming course”, INTED 2024 Proceedings, pp. 966-975.

[14] Neeraj Singh Amarnath & Rajganesh Nagarajan, “An Intelligent Retrieval Augmented Generation Chatbot for Contextually – Aware Conversations to Guide High School Students”, 4th International Conference on Sustainable Expert Systems (ICSES), Kaski, Nepal, Oct. 2024, pp. 1393-1394, 1395.

[15] Bratić D., Šapina M., Jurečić D., Žiljak Gršić J., “Centralized Database Access: Transformer Framework and LLM/Chatbot Integration-Based Hybrid Model”, Appl. Syst. Innov. 2024.

[16] Miladi F. et al., “Leveraging GPT-4 for Accuracy in Education: A Comparative Study on Retrieval – Augmented Generation in MOOCs”, In International Conference on Artificial Intelligence in Education, Switzerland, Jul. 2024, pp. 427-434.

[17] Alexander Tobias Neumann et al., An LLM – Driven Chatbot in Higher Education for Databases and Information Systems, IEEE Transactions on Education (Early Access), Oct 2024.

[18] Li J., Yuan Y., Zhang Z., “Enhancing LLM Factual Accuracy with RAG to Counter Hallucinations: A Case Study on Domain – Specific Queries in Private Knowledge – Bases”, 2024, arXiv preprint arXiv:2403.10446.

[19] Udayan D. et al., “Conversational Chatbot for College Management Using LSTM”, In Proceedings of the International Conference on Innovative Computing & Communication (ICICC), Feb. 2022.

[20] Thway M. et al., “Harnessing GenAI for Higher Education: A Study of a Retrieval Augmented Generation Chatbot's Impact on Human Learning”, Jun. 2024, arXiv preprint arXiv:2406.07796.

[21] Odede J., Frommholz I., “JayBot--Aiding University Students and Admission with an LLM-based Chatbot”, In Proceedings of the 2024 Conference on Human Information Interaction and Retrieval, Mar. 2024, pp. 391-395.

[22] Nakhod, "Using retrieval-augmented generation to elevate low-code developer skills," *Artificial Intelligence*, vol. 1, no 3, pp. 126 – 130, 2023.

[23] Y. Ricke, A. Agarwal, Q. Ma, P. Denny, “Ai-ta: Towards an intelligent question-answer teaching assistant using open-source llms”, 2023, arXiv preprint arXiv:2311.02775.

[24] Tianshi Zheng et al., “Faculty Perspectives on the Potential of RAG in Computer Science Higher Education”, Aug. 2024, arXiv preprint arXiv:2408.01462.

[25] Tirthankar Halder et al., “Enhancing Email Safety: Harnessing ML, DL, and LLM Models for Spam Detection”, International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON), IEEE, Rajshahi, Bangladesh, Sep. 2024, pp. 976-979.

[26] Khaled Matar, Yousef Mohammad, “Improving the Reliability of Educational AI Chatbots Using Retrieval-Augmented Generation”, Engineering Degree Project, Computer Science, Spring, 2024, p. 6.

[27] Mohamed R. Shoaib, Heba M. Emara, Jun Zhao, “A Survey on the Applications of Frontier AI Foundation Models and Large Language Models to Intelligent Transportation Systems”, Computation and Language, Jan. 2024, arXiv:2401.06831, pp. 1-2.

[28] Dang Hoang Anh, Dinh-Truong Do, Vu Tran, Nguyen Le Minh, “The Impact of Large Language Modeling on Natural Language Processing in Legal Texts: A Comprehensive Survey”, 15th International Conference on Knowledge and Systems Engineering (KSE), IEEE, Hanoi, Vietnam, Oct. 2023, pp. 1-5.

[29] Zhao Z. et al., “A Survey of Large Language Models”, Computation and Language, Mar. 2023, arXiv preprint arXiv:2303.18774, p. 9, 17-19, 85, 87-89.

[30] Sonia Vakayil, D. Sujitha Juliet, Anitha J., Sunil Vakayil, “RAG-Based LLM Chatbot Using Llama-2”, 7th International Conference on Devices, Circuits and Systems (ICDCS), IEEE, Coimbatore, India, Apr. 2024, pp. 195-197.

[31] Shervin Minaee et al., “Large Language Models A Survey”, Artificial Intelligence, Feb 2024, arXiv:2402.06196v2, p. 2-7, 15-17, 20.

[32] Minghao Shao, Abdul Basit, Ramesh Karri, Muhammad Shafique, “Survey of different Large Language Model Architectures: Trends, Benchmarks, and Challenges”, Machine Learning, Dec 2024, arXiv:2412.03220v1, pp. 188664-188667, 188688-188696.

[33] A. Radford et al., “Language models are unsupervised multitask learners” OpenAI blog, 2019, pp. 4, 9.

[34] T. Saier, J. Krause, and M. Farber, “Unarxive 2022: All arxiv publications pre-processed for nlp, including structured full-text and citation network”, Computation and Language, Mar. 2023, arXiv preprint arXiv:2303.14957v1, pp. 2-3.

[35] Russell S., Norvig P., *Artificial Intelligence: A Modern Approach (4th ed.)*, California: Pearson, 2020, pp. 36-38.

[36] Venkata Gummadi et al., “Enhancing Communication and Data Transmission Security in RAG using Large Language Models”, 4th International Conference on Sustainable Expert Systems (ICSES), IEEE, Kaski, Nepal, Dec. 2024, pp. 612-614.

[37] Y. Bhanu Sree et al., “Retrieval-Augmented Generation based Large Language Model Chatbot for Improving Diagnosis for Physical and Mental Health”, 6th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), IEEE, Pattaya, Thailand, Nov. 2024, pp. 1-4.

[38] Gihan Gamage eta al., “Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems”, International Conference on Industrial Technology (ICIT), IEEE, Bristol, United Kingdom, Mar. 2024, pp. 1-2.

[39] Yi Zhang et al., “Long-Term Memory for Large Language Models Through Topic-Based Vector Database”, International Conference on Asian Language Processing (IALP), IEEE, Singapore, Nov. 2023, pp. 259-260.

[40] Introduction – Chroma Docs, “Chroma”. [Online]. Available: https://docs.trychroma.com/docs/overview/introduction.

[41] Datacamp: Learn data and AI skills, “Chroma DB Tutorial: A Step-By-Step Guide”. [Online]. Available: https://www.datacamp.com/tutorial/chromadb-tutorial-step-by-step-guide.

[42] Wang J., Duan Z., “Agent AI with Lang Graph: A Modular Framework for Enhancing Machine Translation Using Large Language Models”, J Curr Trends Comp Sci Res, 3(6), 2024, pp. 01-08.

[43] Bocklisch T., Faulkner J., Pawlowski N., Nichol A., “Rasa: Open Source Language Understanding and Dialogue Management”, 2017, ArXiv, abs/1712.05181.

[44] Python Official Documentation (n.d.), “Environment Variables in Python”. [Online]. Available: https://docs.python.org/3/library/os.html.

[45] Groq API Documentation (n.d.), Retrieved from https://groq.com/docs/.

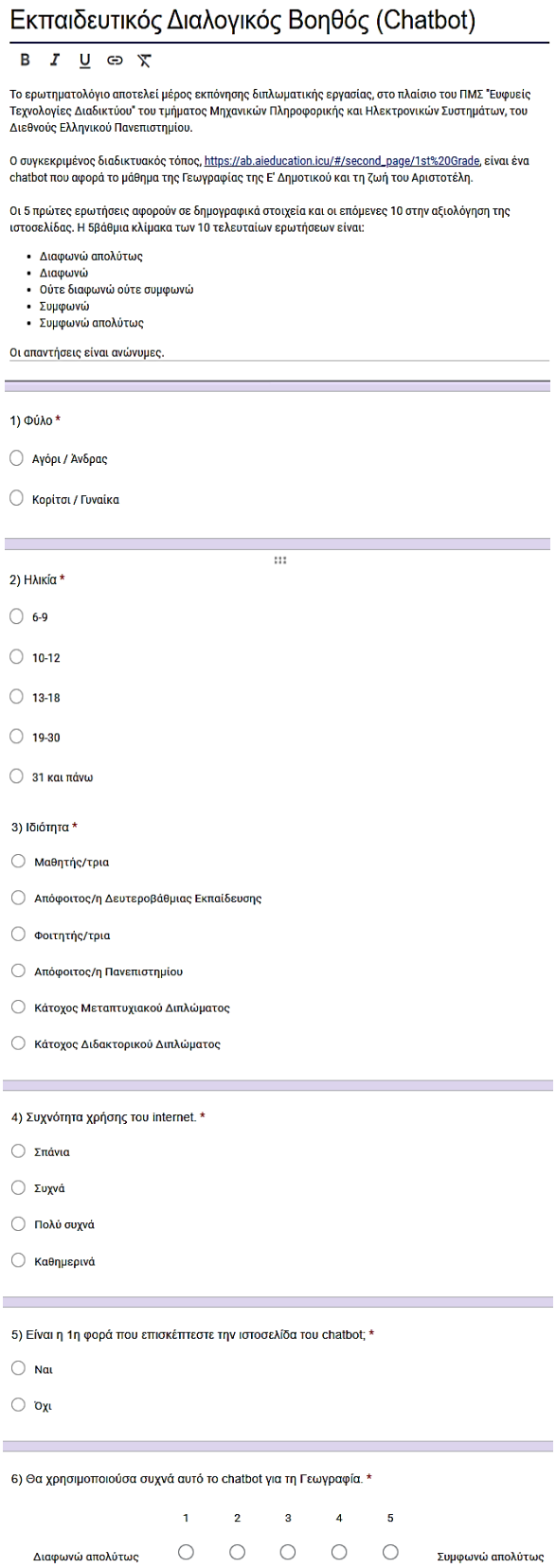
[46] Cloudflare (n.d.), “Secure and accelerate your website”, [Online]. Available: https://www.cloudflare.com/.

[47] Voukoutis Leon et al., “Meltemi: The first open Large Language Model for Greek”, 2024, 10.48550/arXiv.2407.20743.

[48] Interaction Design Foundation, “System Usability Scale for Data-Driven UX”. [Online]. Available: https://www.interaction-design.org/literature/article/system-usability-scale?srsltid=AfmBOoqMKjVv9uP5Wi1TciYaIRvJob4XTbUlZpolUFfPxeUMqsV_Rcjl.

ΠΑΡΑΡΤΗΜΑ Α : ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ

Παρακάτω παρουσιάζονται εικόνες από το ερωτηματολόγιο που χρησιμοποιήθηκε για την αξιολόγηση της χρηστικότητας του chatbot και ο πίνακας με τις απαντήσεις.



Εκπαιδευτικός Διαλογικός Βοηθός (Chatbot)

Το ερωτηματολόγιο αποτελεί μέρος εκπόνησης διπλωματικής εργασίας, στο πλαίσιο του ΠΜΣ "Ευφυείς Τεχνολογίες Διαδικτύου" του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, του Διεθνούς Ελληνικού Πανεπιστημίου.

Ο συγκεκριμένος διαδικτυακός τόπος, https://ab.aieducation.icu/#/second_page/1st%20Grade, είναι ένα chatbot που αφορά το μάθημα της Γεωγραφίας της Ε' Δημοτικού και τη ζωή του Αριστοτέλη.

Οι 5 πρώτες ερωτήσεις αφορούν σε δημογραφικά στοιχεία και οι επόμενες 10 στην αξιολόγηση της ιστοσελίδας. Η 5βάθμια κλίμακα των 10 τελευταίων ερωτήσεων είναι:

- Διαφωνώ απολύτως
- Διαφωνώ
- Ούτε διαφωνώ ούτε συμφωνώ
- Συμφωνώ
- Συμφωνώ απολύτως

Οι απαντήσεις είναι ανώνυμες.

1) Φύλο *

Αγόρι / Άνδρας

Κορίτσι / Γυναίκα

2) Ηλικία *

6-9

10-12

13-18

19-30

31 και πάνω

3) Ιδιότητα *

Μαθητής/τρια

Απόφοιτος/η Δευτεροβάθμιας Εκπαίδευσης

Φοιτητής/τρια

Απόφοιτος/η Πανεπιστημίου

Κάτοχος Μεταπτυχιακού Διπλώματος

Κάτοχος Διδακτορικού Διπλώματος

4) Συχνότητα χρήσης του internet. *

Σπάνια

Συχνά

Πολύ συχνά

Καθημερινά

5) Είναι η 1η φορά που επισκέπτεστε την ιστοσελίδα του chatbot; *

Ναι

Όχι

6) Θα χρησιμοποιούσα συχνά αυτό το chatbot για τη Γεωγραφία. *

	1	2	3	4	5	
Διαφωνώ απολύτως	○	○	○	○	○	Συμφωνώ απολύτως

7) Βρήκα το chatbot αρκετά πολύπλοκο. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

8) Θεωρώ ότι το chatbot ήταν εύκολο στη χρήση. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

9) Νομίζω ότι θα χρειαστώ τη βοήθεια κάποιου ειδικού για να χρησιμοποιήσω το chatbot. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

10) Οι λειτουργίες του chatbot ήταν καλά ενσωματωμένες. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

11) Βρήκα πολλές ασυνέπειες στη λειτουργία του chatbot. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

12) Πιστεύω ότι οι περισσότεροι μαθητές/εκπαιδευτικοί θα μπορούσαν να μάθουν γρήγορα πώς να το χρησιμοποιούν. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

13) Το chatbot ήταν δύσχρηστο. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

14) Είχα αυτοπεποίθηση χρησιμοποιώντας το chatbot. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

15) Χρειάζεται να γνωρίζω πολλά πριν μπορέσω να χρησιμοποιήσω το chatbot. *

	1	2	3	4	5	
Διαφωνώ απολύτως	◯	◯	◯	◯	◯	Συμφωνώ απολύτως

Παρακαλώ πατήστε το κουμπί "Υποβολή". Ευχαριστώ για τον χρόνο που διαθέσατε.

Α/Α	1) Φύλο	2) Ηλικία	3) Ιδιότητα	4) Συχνότητα χρήσης του internet.	5) Είναι η 1η φορά που επισκέπτεστε την ιστοσελίδα του chatbot;
1	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Συχνά	Ναι
2	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Καθημερινά	Όχι
3	Κορίτσι / Γυναίκα	10-12	Μαθητής/τρια	Καθημερινά	Ναι
4	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Πολύ συχνά	Ναι
5	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Καθημερινά	Ναι
6	Κορίτσι / Γυναίκα	10-12	Μαθητής/τρια	Καθημερινά	Ναι
7	Κορίτσι / Γυναίκα	10-12	Μαθητής/τρια	Καθημερινά	Ναι
8	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Πολύ συχνά	Ναι
9	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Συχνά	Όχι
10	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Πολύ συχνά	Όχι
11	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Συχνά	Ναι
12	Κορίτσι / Γυναίκα	10-12	Μαθητής/τρια	Καθημερινά	Ναι
13	Αγόρι / Άνδρας	6-9	Μαθητής/τρια	Σπάνια	Όχι
14	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Σπάνια	Όχι
15	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Συχνά	Ναι
16	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Καθημερινά	Ναι
17	Κορίτσι / Γυναίκα	10-12	Μαθητής/τρια	Συχνά	Ναι
18	Αγόρι / Άνδρας	10-12	Μαθητής/τρια	Καθημερινά	Ναι
19	Κορίτσι / Γυναίκα	31 και πάνω	Απόφοιτος/η Πανεπιστημίου	Καθημερινά	Ναι
20	Αγόρι / Άνδρας	31 και πάνω	Απόφοιτος/η Δευτεροβάθμιας Εκπαίδευσης	Καθημερινά	Ναι
21	Κορίτσι / Γυναίκα	13-18	Μαθητής/τρια	Καθημερινά	Όχι
22	Κορίτσι / Γυναίκα	31 και πάνω	Απόφοιτος/η Πανεπιστημίου	Καθημερινά	Ναι

Ο κώδικας βρίσκεται στην ιστοσελίδα: https://github.com/haris2718/thesis_ihu.