

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΤΗΣ ΑΓΩΓΗΣ
ΠΑΙΔΑΓΩΓΙΚΟ ΔΗΜΟΤΙΚΗΣ ΕΚΠΑΙΔΕΥΣΗΣ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΑΣΙΛΕΙΟΣ ΤΑΣΙΟΣ

96

ΜΑΡΤΙΟΣ 2026

ΣΧΕΔΙΑΣΗ ΑΝΟΙΧΤΟΥ ΔΙΑΛΟΓΙΚΟΥ ΠΡΑΚΤΟΡΑ ΓΙΑ ΤΗΝ
ΕΚΠΑΙΔΕΥΣΗ

Σχεδίαση ανοικτού διαλογικού πράκτορα για την εκπαίδευση

Βασίλειος Τάσιος

pel200096@uoi.gr

Παιδαγωγικό Τμήμα Δημοτικής Εκπαίδευσης, Πανεπιστήμιο Ιωαννίνων

Περίληψη: Η ραγδαία ανάπτυξη των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) έχει δημιουργήσει νέες δυνατότητες για την υποστήριξη της εκπαιδευτικής διαδικασίας, εγείροντας παράλληλα ζητήματα ακρίβειας, ιδιωτικότητας και παιδαγωγικής αξιοπιστίας. Η παρούσα διπλωματική εργασία διερευνά τη χρήση ενός τοπικά εκτελούμενου συστήματος LLM βασισμένου στην αρχιτεκτονική Ανάκτησης-Ενισχυμένης Παραγωγής (Retrieval-Augmented Generation - RAG), με στόχο την υποστήριξη της διδασκαλίας και της μελέτης αλγορίθμων και δομών δεδομένων.

Στο πλαίσιο της εργασίας υλοποιήθηκε ένα σύστημα LLM-RAG, το οποίο λειτουργεί αποκλειστικά σε τοπική υποδομή και αντλεί πληροφορία μόνο από προκαθορισμένο εκπαιδευτικό σώμα (corpus). Το σύστημα αξιοποιεί μηχανισμούς σημασιολογικής ανάκτησης και γλωσσικής παραγωγής, με αυστηρό περιορισμό της απάντησης στο διαθέσιμο υλικό, αποφεύγοντας την παραγωγή μη τεκμηριωμένων ή παραπλανητικών πληροφοριών.

Η αξιολόγηση του συστήματος πραγματοποιήθηκε μέσω πειραματικής διαδικασίας βασισμένης σε τρία ερευνητικά ερωτήματα που αφορούν τη λειτουργική απόδοση, την παιδαγωγική αξία και τους περιορισμούς της προτεινόμενης προσέγγισης. Τα αποτελέσματα έδειξαν ότι το σύστημα παρέχει συνεκτικές, αναπαραγωγίμες και παιδαγωγικά χρήσιμες απαντήσεις, ενώ επιδεικνύει αυξημένη ανθεκτικότητα σε φαινόμενα ψευδαισθήσης (hallucination). Παράλληλα, αναδείχθηκαν περιορισμοί που σχετίζονται με τους υπολογιστικούς πόρους και την εξάρτηση από την ποιότητα του εκπαιδευτικού υλικού.

Συμπερασματικά, η εργασία καταδεικνύει ότι τα τοπικά συστήματα LLM-RAG μπορούν να αποτελέσουν μια αξιόπιστη και παιδαγωγικά βιώσιμη λύση για την ενσωμάτωση της τεχνητής νοημοσύνης στην εκπαίδευση, προσφέροντας αυξημένο έλεγχο, διαφάνεια και προστασία της ιδιωτικότητας.

Λέξεις κλειδιά: Μεγάλα γλωσσικά μοντέλα, Retrieval-Augmented Generation, εικονικοί βοηθοί, εκπαιδευτική τεχνητή νοημοσύνη

Abstract

The rapid development of Large Language Models (LLMs) has introduced new opportunities for supporting educational processes, while simultaneously raising concerns regarding accuracy, privacy, and pedagogical reliability. This thesis investigates the use of a locally executed LLM system based on the Retrieval-Augmented Generation (RAG) architecture, aiming to support the teaching and learning of algorithms and data structures.

Within the scope of this study, a local LLM-RAG system was implemented. The system operates entirely on local infrastructure and retrieves information exclusively from a predefined educational corpus. By combining semantic retrieval mechanisms with controlled language generation, the system restricts its responses to verified source material, effectively reducing the risk of hallucinations and ungrounded content.

The system was evaluated through an experimental process guided by three research questions addressing functional performance, pedagogical value, and implementation limitations. The results indicate that the proposed system produces coherent, reproducible, and pedagogically relevant responses, while demonstrating robustness against hallucination phenomena. At the same time, limitations related to computational resources and dependence on the quality of the embedded educational content were identified.

Overall, this thesis demonstrates that locally deployed LLM-RAG systems can serve as a reliable and pedagogically sound approach for integrating artificial intelligence into educational environments, offering enhanced transparency, control, and data privacy.

Πίνακας Περιεχομένων

Ανάπτυξη ελεύθερου και ανοιχτού εικονικού βοηθού	2
Εισαγωγή.....	5
Ανασκόπηση βιβλιογραφίας.....	5
Μεθοδολογία	21
Υλοποίηση.....	25
Αποτελέσματα και ανάλυση.....	29
Συζήτηση και συμπεράσματα.....	39
Βιβλιογραφικές αναφορές	47
Παράρτημα: Κώδικας υλοποίησης.....	50

Εισαγωγή

Σκοπός της εργασίας

Η παρούσα διπλωματική εργασία εξετάζει την αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models – LLMs) στην εκπαίδευση, με έμφαση στην προστασία της ιδιωτικότητας και της ασφάλειας των δεδομένων. Σε ένα περιβάλλον όπου οι περισσότερες λύσεις Τεχνητής Νοημοσύνης (Artificial Intelligence – AI) βασίζονται σε υπηρεσίες υπολογιστικού νέφους (cloud) και απαιτούν την αποστολή δεδομένων σε απομακρυσμένους διακομιστές, ανακύπτουν σημαντικοί προβληματισμοί σχετικά με τη διαχείριση προσωπικών ή ευαίσθητων πληροφοριών. Ο σκοπός της εργασίας είναι να διερευνήσει κατά πόσο ένα τοπικό σύστημα Γενετικό Προεκπαιδευμένο Μετασχηματιστικό Μοντέλο (Generative Pre-trained Transformer – GPT), το οποίο λειτουργεί αποκλειστικά σε υπολογιστικό περιβάλλον του χρήστη και αντλεί πληροφορίες μόνο από το υλικό που του παρέχεται, μπορεί να αποτελέσει μία ασφαλή και αποτελεσματική λύση για εκπαιδευτικές εφαρμογές.

Η μελέτη επικεντρώνεται στην ανάπτυξη και αξιολόγηση μιας υλοποίησης που βασίζεται σε LLMs και την τεχνική RAG, με στόχο να εξεταστεί η εκπαιδευτική χρησιμότητά της σε πραγματικά σενάρια: αναζήτηση γνώσης, υποστήριξη αυτοκατευθυνόμενης μάθησης, προετοιμασία διδασκαλίας και πρόσβαση σε εξειδικευμένο εκπαιδευτικό περιεχόμενο. Παράλληλα εξετάζονται οι παιδαγωγικές δυνατότητες και οι περιορισμοί τέτοιων συστημάτων, υπό το πρίσμα της χρήσης τους από μαθητές, φοιτητές και εκπαιδευτικούς, καθώς και οι προϋποθέσεις για την ορθή ενσωμάτωσή τους σε εκπαιδευτικά πλαίσια.

Ανασκόπηση Βιβλιογραφίας

Η ενσωμάτωση της τεχνητής νοημοσύνης και των μεγάλων γλωσσικών μοντέλων (LLMs) έχει εγκαινιάσει μια σημαντική μεταμόρφωση σε πολλούς τομείς, συμπεριλαμβανομένης της εκπαίδευσης και της διαχείρισης πληροφοριών (Ma et al., 2024; Swacha & Gracel, 2025). Η πρόσφατη βιβλιογραφία μελετά την εξέλιξη της τεχνολογίας σε μαθησιακά περιβάλλοντα (Pergantis et al., 2025; Park et al., 2022), τις δυνατότητες και τα μειονεκτήματα των LLMs (Ma

et al., 2024; You et al., 2024), καθώς και τις ανησυχίες περί απορρήτου που σχετίζονται με την εφαρμογή τους (Ieva et al., 2024; Raghu et al., 2021), υπογραμμίζοντας ιδιαίτερα τη σημασία των τεχνικών Retrieval-Augmented Generation (RAG) για τη βελτίωση της ακρίβειας και της αξιοπιστίας των συστημάτων (Ma et al., 2024; Moric et al., 2024).

ΤΠΕ στην Εκπαίδευση και η ανάπτυξη της Τεχνητής Νοημοσύνης

Η εξέλιξη της ψηφιακής τεχνολογίας έχει συμβάλει καθοριστικά στη διεύρυνση των δυνατοτήτων μάθησης και στην ενίσχυση της πνευματικής ανάπτυξης του ατόμου (Pergantis et al., 2025). Στο πλαίσιο αυτό, οι Τεχνολογίες Πληροφορίας και Επικοινωνιών (ΤΠΕ) ενσωματώθηκαν συστηματικά στην εκπαίδευση, με κύριο στόχο την οργάνωση, τη διαχείριση και τη διάχυση του εκπαιδευτικού περιεχομένου. Χαρακτηριστικό παράδειγμα αποτελούν τα Συστήματα Διαχείρισης Μάθησης (Learning Management Systems - LMSs), τα οποία λειτουργούν ως κεντρικές ψηφιακές πλατφόρμες για την παροχή μαθημάτων, την επικοινωνία και τη διοικητική υποστήριξη των φοιτητών στα πανεπιστήμια, όπως το Moodle (Lee & Rew, 2025 · Dahal et al., 2025). Οι υποδομές αυτές κρίνονται απαραίτητες για την παροχή επεκτάσιμων και οργανωμένων μαθησιακών εμπειριών (Lee & Rew, 2025).

Παρά τη λειτουργική τους αξία, τα παραδοσιακά LMS σχεδιάστηκαν κυρίως για τη διαχείριση περιεχομένου και όχι για τη δυναμική προσαρμογή της μαθησιακής εμπειρίας στις ανάγκες του εκάστοτε εκπαιδευόμενου. Η βιβλιογραφία επισημαίνει ότι η μετάβαση προς πιο εξατομικευμένες μορφές μάθησης συνεπάγεται αυξημένο κόστος και οργανωτική πολυπλοκότητα, ενώ τα LMS παρουσιάζουν εγγενείς περιορισμούς ως προς την παροχή άμεσης, προσαρμοσμένης ανατροφοδότησης και τη διαλογική υποστήριξη της μάθησης (Dahal et al., 2025). Οι περιορισμοί αυτοί ανέδειξαν την ανάγκη αξιοποίησης τεχνολογιών που θα μπορούσαν να υπερβούν τη στατική φύση των παραδοσιακών ψηφιακών εργαλείων και να υποστηρίξουν πιο ενεργές και εξατομικευμένες μαθησιακές διαδικασίες.

Σε αυτό το πλαίσιο, η Τεχνητή Νοημοσύνη αναδύθηκε ως φυσική εξέλιξη των ΤΠΕ στην εκπαίδευση, καθώς επιτρέπει τη δημιουργία συστημάτων ικανών να προσαρμόζονται στη συμπεριφορά και τις ανάγκες των χρηστών. Τα συστήματα ΤΝ έχουν μελετηθεί εκτενώς ως μέσα ενίσχυσης της μαθησιακής εμπειρίας, ιδίως μέσω της εξατομίκευσης του περιεχομένου και της αύξησης της αφοσίωσης των εκπαιδευομένων (Pergantis et al., 2025 · Wölfel et al., 2024 · Yao & González-Vélez, 2025). Ενδεικτικό παράδειγμα αποτελούν τα ευφυή συστήματα διδασκαλίας (Intelligent Tutoring Systems - ITS), τα οποία αξιοποιούν τεχνικές ΤΝ για την

προσαρμογή της διδασκαλίας στο επίπεδο και την πρόοδο του μαθητή (Yao & González-Vélez, 2025). Ωστόσο, πολλές από τις πρώιμες εφαρμογές TN επικεντρώνονταν κυρίως σε μηχανισμούς πρόβλεψης και κανόνων, περιορίζοντας τη δυνατότητα φυσικής αλληλεπίδρασης και επεξήγησης της γνώσης.

Παραγωγική TN, τα Μεγάλα Γλωσσικά Μοντέλα και τους Εικονικούς Βοηθούς

Η ανάγκη για πιο φυσική, διαλογική και επεξηγηματική αλληλεπίδραση με τα εκπαιδευτικά συστήματα οδήγησε στη σταδιακή υιοθέτηση της Παραγωγικής Τεχνητής Νοημοσύνης (Generative Artificial Intelligence - GenAI), η οποία αποτελεί υποκατηγορία της TN. Σε αντίθεση με παραδοσιακές εφαρμογές TN που περιορίζονται στην ταξινόμηση ή την πρόβλεψη δεδομένων, η Παραγωγική TN έχει τη δυνατότητα να παράγει νέο περιεχόμενο, προσομοιώνοντας την ανθρώπινη παραγωγή λόγου και γνώσης (Zafar, Shaheen & Rehan, 2024). Η ιδιότητα αυτή καθιστά τη Παραγωγική TN ιδιαίτερα κατάλληλη για εκπαιδευτικά περιβάλλοντα, όπου η επεξήγηση, η αναδιτύπωση και η προσαρμογή της γνώσης αποτελούν βασικές παιδαγωγικές απαιτήσεις.

Η πρακτική υλοποίηση της Παραγωγικής TN στον τομέα της εκπαίδευσης βασίζεται κυρίως στα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models - LLMs), τα οποία εκπαιδεύονται σε εκτεταμένα σύνολα κειμενικών δεδομένων και λειτουργούν ως μηχανισμοί κατανόησης και παραγωγής φυσικής γλώσσας. Τα LLMs επιτρέπουν την ανάπτυξη συστημάτων που μπορούν να απαντούν σε ερωτήματα, να παρέχουν επεξηγήσεις και να δημιουργούν πρωτότυπες απαντήσεις προσαρμοσμένες στο πλαίσιο της ερώτησης, καλύπτοντας έτσι το κενό που άφηναν προηγούμενες, πιο στατικές μορφές TN (Kasneci et al., 2023). Εφαρμογές μοντέλων όπως τα GPT-4 και Llama-3 (Large Language Model Meta AI) έχουν δείξει ότι μπορούν να λειτουργήσουν ως συμπληρωματικοί γνωσιακοί πόροι, παρέχοντας ακριβείς απαντήσεις και αναλυτικές, βήμα-προς-βήμα εξηγήσεις σε εξειδικευμένα γνωστικά αντικείμενα (Torphel et al., 2025).

Η παιδαγωγικά αξιοποιήσιμη μορφή των LLMs αναδεικνύεται κυρίως μέσω της ενσωμάτωσής τους σε εικονικούς βοηθούς και συνομιλητές (chatbots) που βασίζονται σε TN. Οι εικονικοί αυτοί βοηθοί μετατρέπουν τις αφηρημένες δυνατότητες των LLMs σε πρακτικά εργαλεία υποστήριξης της μάθησης, προσφέροντας διαλογική αλληλεπίδραση και άμεση ανατροφοδότηση στους φοιτητές (Pergantis et al., 2025 · Qiu et al., 2025). Η βιβλιογραφία τους αναγνωρίζει ως ψηφιακούς συνεργάτες μάθησης, οι οποίοι συμβάλλουν όχι μόνο στη

γνωστική αλλά και στη μεταγνωστική και κοινωνική ανάπτυξη των εκπαιδευομένων (Pergantis et al., 2025).

Η αποτελεσματικότητα των εικονικών βοηθών τεκμηριώνεται περαιτέρω από τη δυνατότητά τους να απαντούν σε ερωτήματα πέρα από τους χρονικούς περιορισμούς της παραδοσιακής διδασκαλίας και να αποσυμφορούν τον φόρτο εργασίας των εκπαιδευτικών, αναλαμβάνοντας τη διαχείριση επαναλαμβανόμενων ερωτήσεων (Morici et al., 2024; Pergantis et al., 2025). Παράλληλα, οι εν δυνάμει εκπαιδευτικοί εκφράζουν θετική στάση απέναντι στη χρήση της Παραγωγικής ΤΝ, αντιλαμβανόμενοι τους εικονικούς βοηθούς κυρίως ως υποστηρικτικά εργαλεία μάθησης και όχι ως υποκατάστατα της διδασκαλίας (Reichert, 2025). Χαρακτηριστικό παράδειγμα αποτελεί το RAGMan, ένα σύστημα βασισμένο σε LLM που αναπτύχθηκε για την υποστήριξη πρωτοετών φοιτητών προγραμματισμού και αξιολογήθηκε θετικά για το μη επικριτικό και υποστηρικτικό του περιβάλλον (Ma et al., 2024).

Μεγάλα Γλωσσικά Μοντέλα: Δυνατότητες και εγγενείς περιορισμοί

Τα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models - LLMs) αποτελούν προηγμένα συστήματα Τεχνητής Νοημοσύνης με εξαιρετική ικανότητα στη δημιουργία, δομή και κατανόηση μεγάλων ποσοτήτων κειμένου (Ehrlich-Sommer et al., 2025). Η λειτουργία τους βασίζεται στη στατιστική και ακολουθιακή πρόβλεψη λέξεων, μέσω της οποίας παράγεται κείμενο που προσομοιάζει σε μεγάλο βαθμό τον ανθρώπινο λόγο (Ehrlich-Sommer et al., 2025). Χάρη στην εκπαίδευσή τους σε τεράστια σύνολα δεδομένων, τα LLMs εμφανίζουν αναδυόμενες ικανότητες συλλογισμού, σχεδιασμού και μάθησης από συμπραζόμενα, γεγονός που τα καθιστά ιδιαίτερα ελκυστικά για εφαρμογές που απαιτούν φυσική γλωσσική αλληλεπίδραση (Ehrlich-Sommer et al., 2025).

Σύγχρονα μοντέλα όπως τα GPT-4, Llama-3 και Claude 3.5 υποστηρίζουν εκτεταμένα συμπραζόμενα, επιτρέποντας την επεξεργασία πλούσιου ιστορικού πληροφορίας και την υποστήριξη πολυγύρων διαλόγων (Ehrlich-Sommer et al., 2025; Torhel et al., 2025). Η δυνατότητα αυτή ενισχύει τη χρησιμότητά τους σε εκπαιδευτικά περιβάλλοντα, όπου η κατανόηση του πλαισίου και η συνέχεια της αλληλεπίδρασης αποτελούν βασικές προϋποθέσεις αποτελεσματικής μάθησης.

Ωστόσο, οι ίδιες αρχιτεκτονικές ιδιότητες που καθιστούν τα LLMs ισχυρά εργαλεία παραγωγής λόγου εισάγουν και σημαντικούς περιορισμούς. Ένας από τους πλέον κρίσιμους

είναι η τάση παραγωγής «παραισθήσεων», δηλαδή απαντήσεων που εμφανίζονται πειστικές αλλά είναι λανθασμένες ή στερούνται τεκμηρίωσης (Alkaissi & McFarlane, 2023; Ji et al., 2023; You et al., 2024; Torhel et al., 2025; Koller et al., 2025). Το φαινόμενο αυτό συνδέεται άμεσα με την εξάρτηση των LLMs από παραμετρική γνώση και με την πιθανοτική φύση της παραγωγής κειμένου, καθώς τα μοντέλα δεν «γνωρίζουν» με την αυστηρή έννοια, αλλά προβλέπουν την πιο πιθανή συνέχεια ενός κειμένου (Koller et al., 2025; You et al., 2024).

Η απουσία πρόσβασης σε επικαιροποιημένη και επαληθεύσιμη εξωτερική γνώση καθιστά τα LLMs ιδιαίτερα ευάλωτα σε σφάλματα σε εργασίες ερωταποκρίσεων (Question Answering – QA), ειδικά σε εξειδικευμένα γνωστικά πεδία (Khasanova Zafar kizi & Suh, 2025). Οι περιορισμοί αυτοί αποτελούν σοβαρό εμπόδιο για εκπαιδευτικές και επιστημονικές εφαρμογές, όπου η ακρίβεια, η αξιοπιστία και η δυνατότητα επαλήθευσης των απαντήσεων αποτελούν βασικές απαιτήσεις (Ma et al., 2024).

Τοπικά Μοντέλα και ζητήματα απορρήτου: ανάγκη ελέγχου της γνώσης και των δεδομένων

Πέρα από τα ζητήματα αξιοπιστίας, η ευρεία εφαρμογή των LLMs αναδεικνύει σημαντικές προκλήσεις που σχετίζονται με το απόρρητο και την ασφάλεια των δεδομένων, ιδιαίτερα όταν τα συστήματα αυτά αξιοποιούνται σε περιβάλλοντα που διαχειρίζονται ευαίσθητες ή πνευματικά προστατευμένες πληροφορίες (Ehrlich-Sommer et al., 2025; Ieva et al., 2024). Σε εκπαιδευτικά, ιατρικά ή ερευνητικά πλαίσια, η ανεξέλεγκτη αποστολή δεδομένων σε εξωτερικές υπηρεσίες LLM εγείρει σοβαρά ηθικά και νομικά ζητήματα.

Η βιβλιογραφία υπογραμμίζει ότι το απόρρητο πρέπει να αποτελεί θεμελιώδη προτεραιότητα στον σχεδιασμό συστημάτων βασισμένων σε LLMs, απαιτώντας την εφαρμογή μηχανισμών κρυπτογράφησης τόσο κατά την αποθήκευση όσο και κατά τη μετάδοση των δεδομένων (Ieva et al., 2024). Για τον λόγο αυτό, προτείνεται όλες οι διαδικασίες ανάλυσης, επεξεργασίας και οπτικοποίησης να πραγματοποιούνται τοπικά, αποφεύγοντας την εξάρτηση από τρίτους παρόχους και την έκθεση ευαίσθητων πληροφοριών (Ieva et al., 2024).

Η ανάγκη αυτή έχει οδηγήσει πολλούς οργανισμούς στην προτίμηση τοπικής ή ενδοεταιρικής ανάπτυξης LLMs, προκειμένου να διατηρείται πλήρης έλεγχος τόσο των δεδομένων όσο και της συμπεριφοράς του μοντέλου (Ieva et al., 2024). Ενδεικτικά, έχει καταγραφεί η επιλογή τοπικής εγκατάστασης του Llama 2 αντί της χρήσης εξωτερικών υπηρεσιών όπως το GPT-4, με

στόχο την ενίσχυση του απορρήτου μέσω τοπικού backend για εικονικούς βοηθούς (Ieva et al., 2024).

Η σημασία του τοπικού ελέγχου αναδεικνύεται ακόμη εντονότερα σε τομείς όπως η ιατρική, όπου η κλινική πρακτική συχνά απαγορεύει τη μεταφορά πραγματικών δεδομένων ασθενών σε εξωτερικά chatbots, καθιστώντας αναγκαία τη δημιουργία ασφαλών και ιδιωτικών ψηφιακών περιβαλλόντων (Choi et al., 2025; Fink et al., 2025). Παρόμοια, chatbots που διαχειρίζονται ευαίσθητα δεδομένα, όπως γενετικές πληροφορίες, απαιτούν ιδιαίτερα προσεκτικό σχεδιασμό ώστε να διασφαλίζεται η ακρίβεια και η αξιοπιστία των απαντήσεων (Coen et al., 2025). Ωστόσο, η τοπική ανάπτυξη μοντέλων, παρότι επιλύει ζητήματα απορρήτου, δεν αντιμετωπίζει από μόνη της το πρόβλημα των παραισθήσεων και της έλλειψης τεκμηριωμένης γνώσης.

Πλεονεκτήματα και Περιορισμοί των Συστημάτων Τεχνητής Νοημοσύνης στην Εκπαίδευση

Η ανάλυση των Μεγάλων Γλωσσικών Μοντέλων και των ζητημάτων απορρήτου που προκύπτουν από την εφαρμογή τους ανέδειξε ότι η ενσωμάτωση της Τεχνητής Νοημοσύνης στην εκπαίδευση δεν αποτελεί απλώς τεχνολογική επιλογή, αλλά μια σύνθετη παιδαγωγική παρέμβαση με πολλαπλές επιπτώσεις. Εφόσον τα συστήματα TN και οι εικονικοί βοηθοί υιοθετούνται ως εργαλεία υποστήριξης της μάθησης, καθίσταται αναγκαία η συστηματική αποτίμηση τόσο των πλεονεκτημάτων όσο και των περιορισμών τους, προκειμένου να αξιολογηθεί η εκπαιδευτική τους αξία και να καθοριστούν οι όροι υπεύθυνης αξιοποίησής τους.

Η σύγχρονη βιβλιογραφία καταδεικνύει ότι η χρήση συστημάτων TN –και ειδικότερα εφαρμογών που βασίζονται σε Μεγάλα Γλωσσικά Μοντέλα– έχει μεταβάλει ουσιαστικά τον τρόπο με τον οποίο οι εκπαιδευόμενοι προσεγγίζουν, κατανοούν και επεξεργάζονται τη γνώση (Ma et al., 2024 · Swacha & Gracel, 2025 · Wölfel et al., 2024). Τα οφέλη που προκύπτουν είναι σημαντικά, ωστόσο συνοδεύονται από τεχνικούς, παιδαγωγικούς και οργανωτικούς περιορισμούς, οι οποίοι πρέπει να λαμβάνονται υπόψη κατά τον σχεδιασμό και την αξιολόγηση τέτοιων συστημάτων.

Πλεονεκτήματα των συστημάτων TN στην εκπαίδευση

Ένα από τα σημαντικότερα πλεονεκτήματα της TN στην εκπαίδευση είναι η δυνατότητα παροχής εξατομικευμένης μάθησης. Τα συστήματα TN επιτρέπουν την προσαρμογή του εκπαιδευτικού περιεχομένου στις ανάγκες, το επίπεδο γνώσεων και το μαθησιακό προφίλ κάθε εκπαιδευόμενου, προσφέροντας δυναμικές εξηγήσεις, διαφοροποιημένη δυσκολία και στοχευμένη καθοδήγηση (Dahal et al., 2025 · Lee & Rew, 2025 · Yao & González-Vélez, 2025). Η προσαρμοστικότητα αυτή ενισχύει την κατανόηση σύνθετων εννοιών, ιδιαίτερα σε τεχνικά και απαιτητικά γνωστικά αντικείμενα όπου απαιτείται συστηματική επεξήγηση και επανατροφοδότηση (Ma et al., 2024 · Keshtkar et al., 2025).

Παράλληλα, τα συστήματα TN προσφέρουν άμεση και συνεχή πρόσβαση σε πληροφορίες και υποστηρικτικό υλικό, μειώνοντας τον χρόνο αναζήτησης και διευκολύνοντας τη μελέτη και την πλοήγηση σε εκτεταμένα εκπαιδευτικά αποθέματα (Ho et al., 2025 · Nandy et al., 2025). Για πολλούς εκπαιδευόμενους, η δυνατότητα διατύπωσης ερωτημάτων σε ένα μη επικριτικό ψηφιακό περιβάλλον λειτουργεί ενισχυτικά για τη συμμετοχή και την αυτενέργεια, ιδιαίτερα σε περιπτώσεις όπου η ανθρώπινη αλληλεπίδραση ενδέχεται να συνοδεύεται από άγχος ή δισταγμό (Pergantis et al., 2025 · Ma et al., 2024).

Τα οφέλη της TN δεν περιορίζονται στους εκπαιδευόμενους, αλλά επεκτείνονται και στους διδάσκοντες. Τα συστήματα TN μπορούν να αυτοματοποιήσουν την απάντηση συχνών ερωτήσεων, να υποστηρίξουν την προετοιμασία διδακτικού υλικού και να συμβάλουν στη δημιουργία βοηθητικών πόρων, όπως κοιζ, παραδείγματα ή συνοπτικές σημειώσεις (Reichert, 2025 · Nyaaba et al., 2024 · León-Paredes et al., 2025). Με τον τρόπο αυτό μειώνεται η γραφειοκρατική και επαναλαμβανόμενη εργασία, επιτρέποντας στους εκπαιδευτικούς να επικεντρωθούν σε πιο ουσιαστικές παιδαγωγικές δραστηριότητες (Khasanova Zafar kizi & Suh, 2025 · Son et al., 2025). Σε ευρύτερο παιδαγωγικό επίπεδο, η TN μπορεί να ενισχύσει την αυτορρυθμιζόμενη μάθηση και να συμβάλει στη μείωση των ανισοτήτων πρόσβασης στη γνώση, μέσω της διαρκούς διαθεσιμότητας υποστήριξης (Pergantis et al., 2025).

Περιορισμοί και προκλήσεις της TN στην εκπαίδευση

Παρά τα παραπάνω πλεονεκτήματα, η αξιοποίηση της TN στην εκπαίδευση συνοδεύεται από σημαντικούς περιορισμούς. Ένα βασικό ζήτημα αφορά τις αυξημένες υπολογιστικές απαιτήσεις των Μεγάλων Γλωσσικών Μοντέλων, τα οποία απαιτούν σημαντικούς πόρους για να λειτουργήσουν αποδοτικά, ιδιαίτερα σε περιβάλλοντα που απαιτούν διαδραστικότητα και άμεσες αποκρίσεις (Khasanova Zafar kizi & Suh, 2025 · Moric et al., 2024). Οι απαιτήσεις αυτές

μπορούν να αποτελέσουν ανασταλτικό παράγοντα για την ευρεία υιοθέτησή τους, ιδίως σε εκπαιδευτικά ιδρύματα με περιορισμένες τεχνικές υποδομές.

Επιπλέον, η αποτελεσματικότητα των συστημάτων ΤΝ εξαρτάται σε μεγάλο βαθμό από την ποιότητα, τη δομή και την οργάνωση των δεδομένων που χρησιμοποιούνται. Ελλιπώς δομημένο εκπαιδευτικό υλικό, ανεπαρκής τμηματοποίηση ή ακατάλληλη ανάκτηση πληροφορίας μπορεί να οδηγήσουν σε μειωμένη ακρίβεια και υποβάθμιση της ποιότητας των παραγόμενων απαντήσεων (You & González-Vélez, 2025 · Moric et al., 2024 · León-Paredes et al., 2025). Οι περιορισμοί αυτοί δεν αφορούν αποκλειστικά συγκεκριμένες τεχνικές, αλλά αναδεικνύουν τη γενικότερη εξάρτηση των συστημάτων ΤΝ από την ποιότητα των δεδομένων εισόδου.

Ένας ακόμη κρίσιμος περιορισμός αφορά τον κίνδυνο παραγωγής ανακριβών ή παραπλανητικών απαντήσεων, γνωστών ως «hallucinations». Τα LLMs είναι δυνατό να παράγουν συνεκτικό λόγο που στερείται επαρκούς τεκμηρίωσης, γεγονός που ενδέχεται να οδηγήσει σε παραπληροφόρηση ή λανθασμένη κατανόηση (Alkaissi & McFarlane, 2023 · Ji et al., 2023 · You et al., 2024 · Koller et al., 2025). Ακόμη και σε περιπτώσεις όπου εφαρμόζονται τεχνικές ενίσχυσης της γνώσης, όπως η ανάκτηση εξωτερικών πηγών, αποτυχίες στη διαδικασία ανάκτησης ή ανεπαρκές συμφραζόμενο μπορούν να επηρεάσουν αρνητικά την ακρίβεια των απαντήσεων (León-Paredes et al., 2025 · Moric et al., 2024). Ως αποτέλεσμα, η εμπιστοσύνη των χρηστών στα συστήματα αυτά παραμένει συχνά επιφυλακτική και εξαρτάται σε μεγάλο βαθμό από το γνωστικό αντικείμενο και το πλαίσιο χρήσης (Wölfel et al., 2024 · Ehrlich-Sommer et al., 2025).

Σε παιδαγωγικό επίπεδο, εκφράζονται ανησυχίες σχετικά με την πιθανότητα υπερεξάρτησης των εκπαιδευομένων από την ΤΝ, η οποία ενδέχεται να περιορίσει την ανάπτυξη κριτικής σκέψης ή να μειώσει την ενασχόληση με τη διαδικασία επίλυσης προβλημάτων (Ma et al., 2024 · Pergantis et al., 2025). Επιπλέον, η μειωμένη ανθρώπινη αλληλεπίδραση μπορεί να επηρεάσει την ποιότητα της διδακτικής σχέσης και τον ρόλο του διδάσκοντα ως παιδαγωγικού διαμεσολαβητή.

Τέλος, ένας ακόμη περιορισμός αφορά τη γενικευσιμότητα των αποτελεσμάτων, καθώς πολλά συστήματα ΤΝ αξιολογούνται σε συγκεκριμένα γνωστικά αντικείμενα ή εκπαιδευτικά πλαίσια και δεν μεταφέρονται εύκολα σε άλλες βαθμίδες ή θεματικές περιοχές (Khasanova Zafar kizi & Suh, 2025 · Ehrlich-Sommer et al., 2025). Παράγοντες όπως το μέγεθος του context

window περιορίζουν επίσης τον όγκο πληροφορίας που μπορεί να αξιοποιηθεί ταυτόχρονα, επηρεάζοντας τη συνολική απόδοση των συστημάτων.

Συνολικά, η Τεχνητή Νοημοσύνη συνιστά μια τεχνολογία με σημαντικές δυνατότητες βελτίωσης των εκπαιδευτικών πρακτικών, προσφέροντας ενισχυμένη μάθηση, άμεση πρόσβαση στη γνώση και ουσιαστική υποστήριξη για εκπαιδευτικούς και φοιτητές. Ωστόσο, η αποτελεσματική αξιοποίησή της προϋποθέτει προσεκτικό σχεδιασμό και συνεκτίμηση των τεχνικών, παιδαγωγικών και ηθικών περιορισμών, ώστε να διασφαλίζεται η αξιοπιστία, η διαφάνεια και η παιδαγωγική συνοχή των εφαρμογών της.

Χρήση Μεγάλων Γλωσσικών Μοντέλων στην Εκπαίδευση: Τεχνολογίες, εφαρμογές και αρχιτεκτονικές τάσεις

Η πρόσφατη βιβλιογραφία καταδεικνύει εκτενή αξιοποίηση Μεγάλων Γλωσσικών Μοντέλων (LLMs) σε εκπαιδευτικά περιβάλλοντα, με εφαρμογές που κυμαίνονται από εξατομικευμένη διδασκαλία έως υποστήριξη μαθημάτων και διοικητικών διαδικασιών. Οι μελέτες αναδεικνύουν τόσο εμπορικά μοντέλα, όπως τα GPT-3.5 και GPT-4, όσο και ανοικτά μοντέλα, όπως το Llama 2, τα οποία χρησιμοποιούνται είτε ως αυτόνομοι διαλογικοί μηχανισμοί είτε ενταγμένα σε πιο σύνθετες αρχιτεκτονικές.

Τα μοντέλα της οικογένειας GPT αξιοποιούνται ευρέως ως διαλογικοί πυρήνες εκπαιδευτικών chatbots, με διαφορετικές οδηγίες εισόδου (prompt) και ρόλους (π.χ. εκπαιδευτής, βοηθός, ανατροφοδότης). Μελέτες δείχνουν ότι τέτοιες διαμορφώσεις μπορούν να βελτιώσουν την εμπειρία χρήστη και την αποδοχή από τους μαθητές, ιδίως όταν το σύστημα λειτουργεί ως προσωποποιημένος εκπαιδευτικός (Lieb & Goel, 2024). Παράλληλα, τα GPT-4 και GPT-3.5 χρησιμοποιούνται σε πιο σύνθετα σενάρια, όπως υποστήριξη μαθημάτων προγραμματισμού, όπου απαιτείται ανάλυση ερωτημάτων και επεξήγηση σύνθετων εννοιών (Alario-Hoyos et al., 2024).

Ιδιαίτερο ενδιαφέρον παρουσιάζει η χρήση ανοικτών μοντέλων και τεχνικών λεπτομερής προσαρμογής (fine-tuning), κυρίως για λόγους προσαρμογής στο πρόγραμμα σπουδών και τοπικής ανάπτυξης. Fine-tuned ανοικτά LLMs έχουν αξιοποιηθεί για την υλοποίηση σωκρατικών ψηφιακών διδασκόντων, με στόχο την ενίσχυση της κριτικής σκέψης μέσω διερευνητικών ερωτήσεων, καθώς και για συστήματα ευθυγραμμισμένα με σχολικά εγχειρίδια και συγκεκριμένους μαθησιακούς στόχους (Favero et al., 2024 · Li et al., 2023). Ωστόσο, η

παραμετρική ενσωμάτωση γνώσης μέσω *fine-tuning* συνοδεύεται από αυξημένο κόστος, περιορισμένη επαληθευσσιμότητα και δυσκολία επικαιροποίησης.

Πέρα από την επιλογή μοντέλων, η βιβλιογραφία αναδεικνύει σαφή μετατόπιση προς υβριδικές αρχιτεκτονικές που συνδυάζουν ανάκτηση πληροφορίας και γενετική παραγωγή λόγου. Πολλές μελέτες υιοθετούν αρχιτεκτονικές Retrieval-Augmented Generation (RAG), στις οποίες τα LLMs αντλούν περιεχόμενο από επιμελημένα έγγραφα μαθημάτων, βάσεις γνώσης ή υλικό LMS, προκειμένου να «γειώσουν» τις απαντήσεις τους σε συγκεκριμένες πηγές (Alario-Hoyos et al., 2024 · Pasquarelli et al., 2024). Εναλλακτικές προσεγγίσεις περιλαμβάνουν υβριδικά σχήματα ανάκτηση πληροφορίας (Information Retrieval - IR) +LLM ή την ενσωμάτωση γραφημάτων γνώσης, τα οποία βελτιώνουν την ακρίβεια και την προσαρμοστικότητα, αλλά αυξάνουν τη σχεδιαστική και υπολογιστική πολυπλοκότητα (Villegas-Ch et al., 2024).

Σε επίπεδο εφαρμογών, τα LLM-based chatbots χρησιμοποιούνται ως ψηφιακοί διδάσκοντες, βοηθοί εξάσκησης, συστήματα ερωταποκρίσεων, υποστηρικτικά εργαλεία προγραμματισμού και LMS βοηθοί. Τα αποτελέσματα δείχνουν ότι τα συστήματα αυτά μπορούν να υποστηρίξουν μεγάλο όγκο ερωτημάτων και να βελτιώσουν την πρόσβαση στη γνώση, ενώ η αποδοχή από τους χρήστες είναι γενικά θετική (Salminen et al., 2024 · Lieb & Goel, 2024). Παρά ταύτα, τα μαθησιακά οφέλη εμφανίζονται συχνά εξαρτώμενα από τον παιδαγωγικό σχεδιασμό, τη φύση της δραστηριότητας και τον βαθμό ενσωμάτωσης στο εκπαιδευτικό πλαίσιο (Farah et al., 2023).

Συνολικά, η βιβλιογραφία συγκλίνει στο συμπέρασμα ότι τα γενικά LLMs, όταν χρησιμοποιούνται χωρίς μηχανισμούς γείωσης και ελέγχου, παρουσιάζουν περιορισμούς αξιοπιστίας, επαληθευσσιμότητας και προσαρμογής σε συγκεκριμένα εκπαιδευτικά συμφραζόμενα. Οι τεχνικές *fine-tuning* και *prompt engineering* μπορούν να βελτιώσουν μερικώς την απόδοση, αλλά δεν επιλύουν δομικά το πρόβλημα της πρόσβασης σε επικαιροποιημένη και ελεγχόμενη γνώση. Αντιθέτως, οι αρχιτεκτονικές Retrieval-Augmented Generation αναδεικνύονται ως κυρίαρχη τάση, καθώς επιτρέπουν τη σύνδεση της γλωσσικής ισχύος των LLMs με επιμελημένο εκπαιδευτικό υλικό, ενισχύοντας την ακρίβεια, τη διαφάνεια και την εμπιστοσύνη των χρηστών.

Στο πλαίσιο αυτό, η παρούσα εργασία υιοθετεί μια αρχιτεκτονική RAG, επιδιώκοντας να αξιοποιήσει τις δυνατότητες των Μεγάλων Γλωσσικών Μοντέλων ως εικονικών

εκπαιδευτικών βοηθών, χωρίς να θυσιάζεται η αξιοπιστία, ο έλεγχος των πηγών και η ιδιωτικότητα των δεδομένων. Η επιλογή αυτή ευθυγραμμίζεται με τις κυρίαρχες ερευνητικές τάσεις και ανταποκρίνεται στις ιδιαίτερες απαιτήσεις του εκπαιδευτικού περιβάλλοντος που εξετάζεται όπως φαίνεται στον Πίνακα 1.

Πίνακας 1. Πίνακας Σύγκρισης Τεχνολογιών

Κατηγορία	Τεχνολογία / Τεχνική	Ρόλος / Εκπαίδευση	στην Πλεονεκτήματα	Περιορισμοί	Ενδεικτική Βιβλιογραφία
Εμπορικά LLMs	GPT-3.5, GPT-4	Διαλογικοί εκπαιδευτικοί βοηθοί, επεξήγηση εννοιών, ανατροφοδότηση	Υψηλή γλωσσική ποιότητα, ευχρηστία, ευρεία αποδοχή	Εξάρτηση από τρίτους, περιορισμένος έλεγχος και δεδομένων	(Lieb & Goel, 2024 · Alario-Hoyos et al., 2024)
Ανοικτά LLMs	Llama (7B/13B)	2 προσαρμοσμένοι ρόλοι (π.χ. σωκρατικός διάλογος)	Τοπική ανάπτυξη, παιδαγωγικά προσαρμοσμένοι ρόλοι (π.χ. ιδιωτικότητα, ευελιξία)	Έλεγχος δεδομένων, γενική απόδοση χωρίς προσαρμογή	(Favero et al., 2024)
Fine-tuning	Παραμετρική προσαρμογή μοντέλων	Ευθυγράμμιση πρόγραμμα σπουδών μαθησιακούς στόχους	με Βελτιωμένη και εξειδίκευση προσαρμογή	Υψηλό υπολογιστικό κόστος, δύσκολη επικαιροποίηση, περιορισμένη επαληθευσσιμότητα	(Li et al., 2023 · Favero et al., 2024)
Prompt engineering	Δομημένες οδηγίες και ρόλοι	Βελτίωση και απαντήσεων	Βελτίωση ποιότητας συνοχής άμεση εφαρμογή	Χαμηλό κόστος, Δεν εξαλείφει παραισθήσεις, εξαρτάται από τον χρήστη	(Farah et al., 2023)

Κατηγορία	Τεχνολογία / Τεχνική	Ρόλος / Εκπαίδευση	στην Πλεονεκτήματα	Περιορισμοί	Ενδεικτική Βιβλιογραφία
Hybrid IR + LLM	Συνδυασμός αναζήτησης και γενετικής απόκρισης	Υποστήριξη ερωταποκρίσεων και πληροφόρησης φοιτητών	Καλύτερη συνάφεια και εμπιστοσύνη	Αυξημένη και αρχιτεκτονική πολυπλοκότητα	(Wang et al., 2023 · Pasquarelli et al., 2024)
Knowledge Graph LLM	Δομημένο + συμφραζόμενο με LLM	Αυξημένη ακρίβεια, προσαρμοστικότητα σε LMS	Υψηλή σχεδιαστική πολυπλοκότητα	Πολύπλοκη υλοποίηση	(Villegas-Ch et al., 2024)
Retrieval-Augmented Generation (RAG)	Ανάκτηση επιμελημένης γνώσης + LLM	Εκπαιδευτικοί βοηθοί τεκμηριωμένες απαντήσεις	Αξιοπιστία, με διαφάνεια, έλεγχος ιδιωτικότητα	Εξάρτηση ποιότητα, οργάνωση δεδομένων	(Alario-Hoyos από et al., 2024 · και Pasquarelli et al., 2024 · Salminen et al., 2024)

Retrieval-Augmented Generation (RAG) και Ανάκτηση Πληροφοριών

Η ανάγκη για ταυτόχρονη αντιμετώπιση των περιορισμών αξιοπιστίας των Μεγάλων Γλωσσικών Μοντέλων (LLMs) και των απαιτήσεων απορρήτου οδήγησε στην ανάπτυξη τεχνικών που συνδυάζουν τη γλωσσική ισχύ των γενετικών μοντέλων με ελεγχόμενες εξωτερικές πηγές γνώσης. Στο πλαίσιο αυτό, μία από τις πλέον διαδεδομένες και αποτελεσματικές προσεγγίσεις είναι το Retrieval-Augmented Generation (RAG), το οποίο στοχεύει στη μείωση των παραισθήσεων, της έλλειψης επικαιροποιημένης γνώσης και της ασάφειας συμφραζομένων στα αυτόνομα LLMs (Lewis et al., 2020 · Morić et al., 2024 · You et al., 2024 · Temsah et al., 2025).

Το RAG συνδυάζει ένα γενετικό μοντέλο με ένα σύστημα ανάκτησης πληροφοριών, επιτρέποντας στο LLM να αντλεί σχετικό και επιμελημένο περιεχόμενο από εξωτερικές βάσεις γνώσης πριν από τη δημιουργία της απάντησης (Morić et al., 2024). Με τον τρόπο αυτό, οι παραγόμενες απαντήσεις δεν βασίζονται αποκλειστικά στη στατιστική πρόβλεψη λέξεων, αλλά εδράζονται σε συγκεκριμένα αποσπάσματα τεκμηριωμένης γνώσης, γεγονός που

ενισχύει σημαντικά την ακρίβεια (Morić et al., 2024), την αξιοπιστία (You et al., 2024) και την εμπιστοσύνη των χρηστών (Fink et al., 2025). Η αποτελεσματικότητα της προσέγγισης έχει τεκμηριωθεί σε εξειδικευμένα πεδία, όπως η πανεπιστημιακή εκπαίδευση (León-Paredes et al., 2025) και η ιατρική πρακτική (Fink et al., 2025).

Φάσεις της αρχιτεκτονικής RAG

Η αρχιτεκτονική του RAG διαρθρώνεται σε τρεις διακριτές αλλά αλληλένδετες φάσεις, οι οποίες καθορίζουν τη συνολική ποιότητα και αξιοπιστία του συστήματος.

1. Ευρετηρίαση (Προετοιμασία δεδομένων): Στην αρχική φάση, το εκπαιδευτικό ή θεματικό υλικό (π.χ. αρχεία PDF, εγχειρίδια, συχνές ερωτήσεις, σημειώσεις μαθημάτων) υποβάλλεται σε διαδικασίες καθαρισμού και κανονικοποίησης του κειμένου. Στη συνέχεια, το περιεχόμενο τμηματοποιείται σε μικρότερα, νοηματικά συνεκτικά αποσπάσματα (chunks), ώστε να διευκολύνεται η αποτελεσματική ανάκτηση πληροφορίας. Κάθε απόσπασμα μετατρέπεται σε διανυσματική αναπαράσταση (embedding) μέσω μοντέλων ενσωμάτωσης και αποθηκεύεται σε εξειδικευμένες βάσεις διανυσμάτων, όπως οι FAISS, Qdrant ή PostgreSQL/pgvector. Η ποιότητα της τμηματοποίησης και των embeddings αποτελεί κρίσιμο παράγοντα για τη συνολική απόδοση του συστήματος.

2. Ανάκτηση πληροφορίας: Κατά τη φάση της ανάκτησης, το ερώτημα του χρήστη μετατρέπεται επίσης σε embedding και συγκρίνεται με τα αποθηκευμένα διανύσματα χρησιμοποιώντας μετρικές ομοιότητας, όπως η cosine similarity. Το σύστημα ανακτά τα πλέον συναφή αποσπάσματα γνώσης, τα οποία λειτουργούν ως τεκμηριωμένο συμφραζόμενο για το επόμενο στάδιο. Προηγμένες τεχνικές, όπως το MultiQuery retrieval ή τα υβριδικά σχήματα ανάκτησης (συνδυασμός BM25 και dense retrieval), αξιοποιούνται για τη βελτίωση της κάλυψης και της συνάφειας των ανακτημένων πληροφοριών, μειώνοντας τον κίνδυνο απώλειας κρίσιμου περιεχομένου.

3. Δημιουργία απόκρισης: Στην τελική φάση, το LLM λαμβάνει το αρχικό ερώτημα εμπλουτισμένο με τα ανακτημένα αποσπάσματα γνώσης και παράγει την απάντηση βασισμένη στο παρεχόμενο συμφραζόμενο. Με τον τρόπο αυτό, η διαδικασία παραγωγής λόγου περιορίζεται σε τεκμηριωμένο περιεχόμενο, γεγονός που μειώνει στατιστικά σημαντικά την εμφάνιση παραισθήσεων ($p < 0.001$) και ενισχύει την ακρίβεια των απαντήσεων (You et al., 2024 · Morić et al., 2024).

Ένα από τα πλέον κρίσιμα πλεονεκτήματα του RAG είναι η δυνατότητα επαλήθευσης των πηγών, καθώς οι παραγόμενες απαντήσεις μπορούν να συσχετιστούν άμεσα με συγκεκριμένα έγγραφα ή αποσπάσματα γνώσης. Η ιδιότητα αυτή καθιστά το RAG ιδιαίτερα κατάλληλο για εκπαιδευτικά περιβάλλοντα, όπου η διαφάνεια, η τεκμηρίωση και ο έλεγχος της πληροφορίας αποτελούν θεμελιώδεις παιδαγωγικές απαιτήσεις. Παράλληλα, η δυνατότητα τοπικής υλοποίησης αρχιτεκτονικών RAG ευθυγραμμίζεται με τις απαιτήσεις απορρήτου και ασφάλειας που αναδείχθηκαν στις προηγούμενες ενότητες, καθιστώντας το RAG συμβατό με ιδιωτικά και ελεγχόμενα εκπαιδευτικά περιβάλλοντα.

Σύνθεση ευρημάτων της βιβλιογραφίας

Η βιβλιογραφία καταδεικνύει ότι τα σύγχρονα συστήματα τεχνητής νοημοσύνης και ειδικότερα τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) προσφέρουν σημαντικές εκπαιδευτικές προοπτικές, ενισχύοντας την εξατομικευμένη μάθηση, τη διαθεσιμότητα υποστήριξης και την προσβασιμότητα σε σύνθετη γνώση. Η ικανότητά τους να παράγουν φυσική γλώσσα, να προσαρμόζονται στο προφίλ των χρηστών και να διαχειρίζονται μεγάλους όγκους πληροφοριών τα καθιστά ιδιαίτερα ελκυστικά ως εκπαιδευτικούς βοηθούς. Ωστόσο, η πρακτική αξιοποίησή τους σε πραγματικά εκπαιδευτικά περιβάλλοντα αποδεικνύεται σημαντικά πιο σύνθετη, καθώς τα LLMs συνοδεύονται από ουσιαστικά προβλήματα αξιοπιστίας, αβεβαιότητας περιεχομένου και συστημικών περιορισμών (Alkaissi & McFarlane, 2023 · Ji et al., 2023 · You et al., 2024 · Topfel et al., 2025 · Koller et al., 2025).

Οι περιορισμοί αυτοί καθίστανται ιδιαίτερα κρίσιμοι στο πεδίο της εκπαίδευσης, όπου η εγκυρότητα των παρεχόμενων πληροφοριών συνδέεται άμεσα με την ανάπτυξη δεξιοτήτων, την απόκτηση επιστημονικής γνώσης και την ψυχολογική ασφάλεια των μαθητών. Επιπλέον, η ευρεία χρήση cloud-based γλωσσικών μοντέλων, όπως τα GPT-4, Claude ή Gemini, συνεπάγεται συχνά τη μεταφορά εκπαιδευτικών δεδομένων σε τρίτους παρόχους. Ακόμη και η αποστολή ανωνυμοποιημένων δεδομένων έχει αναδειχθεί ως δυνητικός κίνδυνος απορρήτου, ιδίως όταν το εκπαιδευτικό υλικό περιλαμβάνει προσωπικά στοιχεία, βαθμολογίες, σημειώσεις ή ιδιόκτητο περιεχόμενο (Ehrlich-Sommer et al., 2025 · Ieva et al., 2024 · Choi et al., 2025 · Fink et al., 2025). Σε ευαίσθητα περιβάλλοντα, όπως η πανεπιστημιακή ή ιατρική εκπαίδευση, η χρήση εξωτερικών LLMs δεν είναι απλώς προβληματική αλλά συχνά ασύμβατη με κανονισμούς, δεοντολογικά πρότυπα και νομικά πλαίσια (Coen et al., 2025).

Στο πλαίσιο αυτό, η βιβλιογραφία προτείνει διάφορες τεχνικές για τη βελτίωση της αξιοπιστίας και της εκπαιδευτικής καταλληλότητας των LLMs. Το *prompt engineering* αποτελεί μία διαδεδομένη προσέγγιση, μέσω της οποίας η συμπεριφορά του μοντέλου καθορίζεται από προσεκτικά σχεδιασμένες οδηγίες ή ρόλους. Αν και η μέθοδος αυτή μπορεί να βελτιώσει τη συνοχή ή τον παιδαγωγικό τόνο των απαντήσεων, παραμένει περιορισμένη ως προς την τεκμηρίωση και δεν αντιμετωπίζει ουσιαστικά το πρόβλημα των παραιθήσεων, καθώς το μοντέλο εξακολουθεί να βασίζεται αποκλειστικά στην παραμετρική του γνώση (Farah et al., 2023 · Ji et al., 2023).

Μία δεύτερη κατηγορία τεχνικών αφορά το *fine-tuning* γλωσσικών μοντέλων σε εκπαιδευτικά δεδομένα ή συνθετικά σύνολα διαλόγων. Οι προσεγγίσεις αυτές μπορούν να βελτιώσουν την ευθυγράμμιση με το πρόγραμμα σπουδών και συγκεκριμένες παιδαγωγικές στρατηγικές (Li et al., 2023 · Favero et al., 2024), ωστόσο παρουσιάζουν σημαντικούς περιορισμούς ως προς το κόστος, τις υπολογιστικές απαιτήσεις και την ευελιξία, καθώς κάθε τροποποίηση του εκπαιδευτικού υλικού απαιτεί νέα διαδικασία εκπαίδευσης. Επιπλέον, το *fine-tuning* δεν παρέχει εγγενείς μηχανισμούς επαλήθευσης πηγών ούτε διασφαλίζει την επικαιρότητα της γνώσης.

Άλλες προσεγγίσεις, όπως η ενσωμάτωση *knowledge graphs* ή δομημένων βάσεων γνώσης σε συνδυασμό με LLMs, προσφέρουν υψηλό επίπεδο ελέγχου και ακρίβειας, αλλά χαρακτηρίζονται από αυξημένη πολυπλοκότητα σχεδίασης και περιορισμένη προσαρμοστικότητα σε ανοιχτού τύπου ερωτήματα και φυσικό διάλογο (Villegas-Ch et al., 2024). Αντίστοιχα, υβριδικά σχήματα IR+LLM χωρίς ενοποιημένη αρχιτεκτονική συχνά αποτυγχάνουν να διασφαλίσουν συνεκτική σύνδεση μεταξύ της διαδικασίας ανάκτησης και της παραγόμενης απάντησης (Pasquarelli et al., 2024).

Μέσα σε αυτό το τοπίο, το *Retrieval-Augmented Generation (RAG)* αναδεικνύεται ως μία ισορροπημένη και λειτουργικά αποδοτική προσέγγιση, καθώς συνδυάζει τη γλωσσική ευχέρεια των LLMs με δυναμική, ελεγχόμενη και επαληθεύσιμη πρόσβαση σε εξωτερική γνώση, χωρίς να απαιτεί επανεκπαίδευση του μοντέλου (Lewis et al., 2020 · Moric et al., 2024 · You et al., 2024). Ο διαχωρισμός μεταξύ γλωσσικού μοντέλου και βάσης γνώσης, καθώς και η δυνατότητα τοπικής υλοποίησης, καθιστούν το RAG ιδιαίτερα κατάλληλο για εκπαιδευτικά περιβάλλοντα με αυξημένες απαιτήσεις απορρήτου, διαφάνειας και προσαρμοστικότητας (León-Paredes et al., 2025).

Ωστόσο, η βιβλιογραφία επισημαίνει ότι ακόμη και οι αρχιτεκτονικές RAG συχνά υλοποιούνται μέσω cloud υποδομών, διατηρώντας εξαρτήσεις από τρίτους παρόχους και εμπορικές διεπαφές προγραμματισμού εφαρμογών (Application Programming Interface – API). Οι εξαρτήσεις αυτές αυξάνουν το κόστος λειτουργίας, περιορίζουν τον έλεγχο των δεδομένων και υποβαθμίζουν τη δυνατότητα προσαρμογής ενός συστήματος στις ανάγκες συγκεκριμένων μαθημάτων ή ιδρυμάτων (Ma et al., 2024). Παράλληλα, αστοχίες στην ανάκτηση ή ελλιπής προετοιμασία των δεδομένων μπορούν να οδηγήσουν σε ανακριβείς απαντήσεις, ενώ οι μαθητές συχνά δεν διαθέτουν τις δεξιότητες για να αξιολογήσουν την εγκυρότητα της παραγόμενης πληροφορίας (Pergantis et al., 2025).

Συνεπώς, καθίσταται σαφές ότι η ουσιαστική αξιοποίηση της Τεχνητής Νοημοσύνης στην εκπαίδευση προϋποθέτει αρχιτεκτονικές που διασφαλίζουν πλήρη έλεγχο των δεδομένων, διαφανή ενσωμάτωση τεκμηριωμένου εκπαιδευτικού υλικού και ανεξαρτησία από εξωτερικές υπηρεσίες. Η ανάπτυξη ιδιωτικών, τοπικά εκτελούμενων συστημάτων RAG ανταποκρίνεται σε αυτές τις απαιτήσεις, επιτρέποντας τη δημιουργία γνώσης αποκλειστικά από εγκεκριμένο εκπαιδευτικό υλικό και ενισχύοντας την παιδαγωγική αξιοπιστία και την ασφάλεια. Στο πλαίσιο αυτό, η παρούσα εργασία προτείνει την υλοποίηση ενός τοπικού εκπαιδευτικού συστήματος βασισμένου σε αρχιτεκτονική RAG, το οποίο επιχειρεί να γεφυρώσει τις δυνατότητες των σύγχρονων LLMs με τις ανάγκες ελέγχου, διαφάνειας και αξιοπιστίας που χαρακτηρίζουν τα πραγματικά εκπαιδευτικά περιβάλλοντα.

Μεθοδολογία

Με βάση τη σύνθεση των ευρημάτων της βιβλιογραφίας που παρουσιάστηκε στο προηγούμενο κεφάλαιο, καθίσταται σαφές ότι η ουσιαστική αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) στην εκπαίδευση δεν εξαρτάται αποκλειστικά από τις δυνατότητες παραγωγής φυσικής γλώσσας, αλλά κυρίως από τον τρόπο με τον οποίο αυτά ενσωματώνονται σε αρχιτεκτονικές που διασφαλίζουν την αξιοπιστία της πληροφορίας, τον έλεγχο της γνώσης και την προστασία των δεδομένων. Η βιβλιογραφική ανασκόπηση ανέδειξε ότι, παρά την αυξανόμενη υιοθέτηση cloud-based LLMs και εκπαιδευτικών ψηφιακών βοηθών, παραμένουν σημαντικά ερευνητικά και πρακτικά κενά σε σχέση με τη δημιουργία τοπικών, ιδιωτικών συστημάτων που μπορούν να αξιοποιηθούν με ασφάλεια σε ευαίσθητα εκπαιδευτικά περιβάλλοντα.

Στο πλαίσιο αυτό, το παρόν κεφάλαιο παρουσιάζει τη μεθοδολογία που ακολουθήθηκε για τον σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός τοπικού συστήματος εικονικού βοηθού βασισμένου στην αρχιτεκτονική Retrieval-Augmented Generation (RAG). Η μεθοδολογική προσέγγιση συνδέει τη θεωρητική τεκμηρίωση της βιβλιογραφίας με την πρακτική ανάπτυξη ενός λειτουργικού συστήματος, εξετάζοντας τόσο τις τεχνικές παραμέτρους της υλοποίησης όσο και την παιδαγωγική του αξία σε ρεαλιστικά σενάρια εκπαιδευτικής χρήσης.

Ερευνητικός Σχεδιασμός

Η παρούσα έρευνα ακολουθεί μια εφαρμοσμένη μεθοδολογική προσέγγιση, η οποία συνδυάζει στοιχεία σχεδιασμού συστημάτων (systems design), τεχνολογικής υλοποίησης και αξιολόγησης με βάση τη χρήση (use-based evaluation). Πρωταρχικός στόχος αποτελεί η ανάπτυξη και η διερεύνηση ενός τοπικού συστήματος εικονικού βοηθού βασισμένου σε τεχνικές Μεγάλων Γλωσσικών Μοντέλων (LLMs) και Retrieval-Augmented Generation (RAG), το οποίο λειτουργεί αποκλειστικά εντός τοπικής υπολογιστικής υποδομής, χωρίς διασύνδεση με υπηρεσίες διαδικτύου.

Ο ερευνητικός σχεδιασμός διαρθρώνεται γύρω από τρεις κεντρικές συνιστώσες:

1. Αρχιτεκτονικός και τεχνικός σχεδιασμός του συστήματος, με έμφαση στη διασφάλιση της ιδιωτικότητας των δεδομένων και στη διαλειτουργικότητα των επιμέρους υποσυστημάτων.

2. Ανάπτυξη μηχανισμών σημασιολογικής ανάκτησης πληροφορίας, μέσω δημιουργίας και ευρετηρίασης διανυσματικών αναπαραστάσεων (embeddings).
3. Αξιολόγηση της λειτουργικότητας και της εκπαιδευτικής αξίας, με βάση πραγματικά σενάρια χρήσης σε εκπαιδευτικά περιβάλλοντα.

Η μεθοδολογική αυτή δομή επιτρέπει όχι μόνο την υλοποίηση ενός ολοκληρωμένου συστήματος, αλλά και την αποτίμηση της πρακτικής του συνεισφοράς σε σχέση με τους στόχους της εργασίας.

Στόχοι και Ερευνητικοί Άξονες

Η παρούσα έρευνα οργανώνεται γύρω από τρεις βασικούς ερευνητικούς άξονες, οι οποίοι συνδέουν την τεχνολογική υλοποίηση με τη διδακτική αξιοποίηση και την αποτίμηση της λειτουργικής αξίας του συστήματος. Οι άξονες αυτοί αποτυπώνουν τη συνολική φιλοσοφία της μελέτης, όπου το τεχνολογικό εργαλείο δεν αντιμετωπίζεται μόνο ως τεχνικό σύστημα, αλλά ως παιδαγωγικό μέσο με συγκεκριμένη εκπαιδευτική λειτουργία.

Ο πρώτος άξονας αφορά την ανάπτυξη και τεχνική διερεύνηση ενός τοπικού συστήματος εικονικού βοηθού βασισμένου στην προσέγγιση Retrieval-Augmented Generation (RAG). Σε αυτό το πλαίσιο εξετάζεται η δυνατότητα ενός μοντέλου γενετικής τεχνητής νοημοσύνης να λειτουργεί αποκλειστικά σε τοπικό περιβάλλον, χωρίς διασύνδεση με διαδικτυακές υπηρεσίες, προσφέροντας ακρίβεια απαντήσεων βασισμένων σε πραγματικά δεδομένα. Η έρευνα εστιάζει στις συνθήκες εκτέλεσης, στις επιδόσεις του συστήματος σε ερωτήματα που προέρχονται από το σώμα κειμένων (corpus) του χρήστη και στη σταθερότητα των αποτελεσμάτων κατά τη διάρκεια της λειτουργίας του.

Ο δεύτερος άξονας εξετάζει την εκπαιδευτική διάσταση των Large Language Models ως γνωσιακών συνεργατών (cognitive partners). Μελετάται ο τρόπος με τον οποίο το σύστημα μπορεί να υποστηρίξει διαδικασίες μάθησης, όπως η αναζήτηση και κατανόηση πληροφοριών, η μεταγνώση και η αυτορρυθμιζόμενη μάθηση. Στο πλαίσιο αυτό η τεχνητή νοημοσύνη δεν αντιμετωπίζεται ως εργαλείο αυτόματης παραγωγής κειμένου, αλλά ως μηχανισμός ενίσχυσης της διερευνητικής μελέτης, αναδιατύπωσης σύνθετων εννοιών και προσαρμογής του περιεχομένου στις ανάγκες του χρήστη. Η διερεύνηση στοχεύει στην κατανόηση του αντίκτυπου που έχει η τεχνολογία στη μαθησιακή διαδικασία, ιδίως όταν η πληροφορία περιορίζεται αποκλειστικά στο υλικό που ο ίδιος ο χρήστης εισάγει.

Ο τρίτος άξονας αφορά την αποτίμηση των παιδαγωγικών δυνατοτήτων και περιορισμών των συστημάτων LLM σε πραγματικά εκπαιδευτικά περιβάλλοντα. Εξετάζονται οι ευκαιρίες που προσφέρουν –όπως η προσβασιμότητα σε ογκώδες υλικό, η δυνατότητα εξατομικευμένης υποστήριξης και η διευκόλυνση της διαχείρισης γνώσης– αλλά και οι προκλήσεις που προκύπτουν, όπως η εξάρτηση από το corpus του χρήστη, οι ανάγκες υπολογιστικών πόρων και η ανάγκη επιμέλειας των παραγόμενων απαντήσεων. Η έρευνα δεν περιορίζεται μόνο σε τεχνικά χαρακτηριστικά, αλλά αξιολογεί το σύστημα ως μαθησιακό εργαλείο και εξετάζει τα όρια εντός των οποίων μπορεί να αξιοποιηθεί αποτελεσματικά.

Ερευνητικά Ερωτήματα και Κριτήρια Αξιολόγησης

Με βάση τους ερευνητικούς άξονες που αναλύθηκαν, διαμορφώνονται συγκεκριμένα ερευνητικά ερωτήματα (Research Questions - RQ) τα οποία καθοδηγούν τη διαδικασία αξιολόγησης και συνδέουν τη λειτουργία του συστήματος με τα αναμενόμενα παιδαγωγικά οφέλη.

Τα ερωτήματα αυτά δεν αναπαράγουν τους θεωρητικούς στόχους, αλλά εστιάζουν στην εμπειρική διερεύνηση της απόδοσης, της χρηστικής αξίας και των περιορισμών του συστήματος σε πραγματικές συνθήκες χρήσης.

RQ1 – Λειτουργική απόδοση ενός τοπικού GPT συστήματος στην ανάκτηση και παραγωγή γνώσης.

Σε ποιο βαθμό ένα τοπικό σύστημα LLM-RAG μπορεί να παρέχει συνεκτικές και έγκυρες απαντήσεις βασισμένες αποκλειστικά στα έγγραφα του χρήστη;

Το ερώτημα στοχεύει στην αξιολόγηση της τεχνικής δυνατότητας του συστήματος να ενσωματώνει μηχανισμούς σημασιολογικής ανάκτησης και να παράγει περιεχόμενο με ακρίβεια και συνοχή. Ιδιαίτερη έμφαση δίνεται στη σταθερότητα των απαντήσεων, στην ικανότητα αναπαραγωγής των αποτελεσμάτων και στην αποφυγή εισαγωγής εξωγενούς γνώσης (hallucinations).

RQ2 – Παιδαγωγική αξία και διαδικασίες ενίσχυσης της μάθησης.

Πώς η χρήση ενός τοπικού GPT συστήματος επηρεάζει τη μελέτη, την προετοιμασία διδασκαλίας και την αυτοκατευθυνόμενη μάθηση όταν το corpus προέρχεται από εκπαιδευτικό υλικό του ίδιου του χρήστη;

Στόχος του ερωτήματος είναι η διερεύνηση του κατά πόσο η τεχνολογία λειτουργεί ως

εργαλείο υποστήριξης της μαθησιακής διαδικασίας, προσφέροντας εξατομίκευση, προσβασιμότητα σε υλικό και κατανόηση σύνθετων εννοιών.

RQ3 – Περιορισμοί και πρακτικές προκλήσεις κατά την εκπαιδευτική χρήση. *Ποιες είναι οι τεχνολογικές, παιδαγωγικές ή λειτουργικές δυσκολίες που προκύπτουν από την εφαρμογή ενός τοπικού GPT συστήματος στην εκπαίδευση;* Το ερώτημα εστιάζει σε ζητήματα όπως απαιτήσεις υπολογιστικών πόρων, ποιότητα εισαγόμενου υλικού, δυσκολίες σημασιολογικής ανάκτησης, αλλά και όρια στη χρήση από μαθητές και εκπαιδευτικούς.

Τα ερευνητικά ερωτήματα, η μεθοδολογική τους προσέγγιση και οι αντίστοιχοι δείκτες αξιολόγησης συνοψίζονται στον Πίνακα 2, προκειμένου να αποτυπωθεί με σαφήνεια η αντιστοιχία μεταξύ στόχων και διαδικασιών αξιολόγησης.

Πίνακας 2. Πίνακας ερευνητικών ερωτημάτων

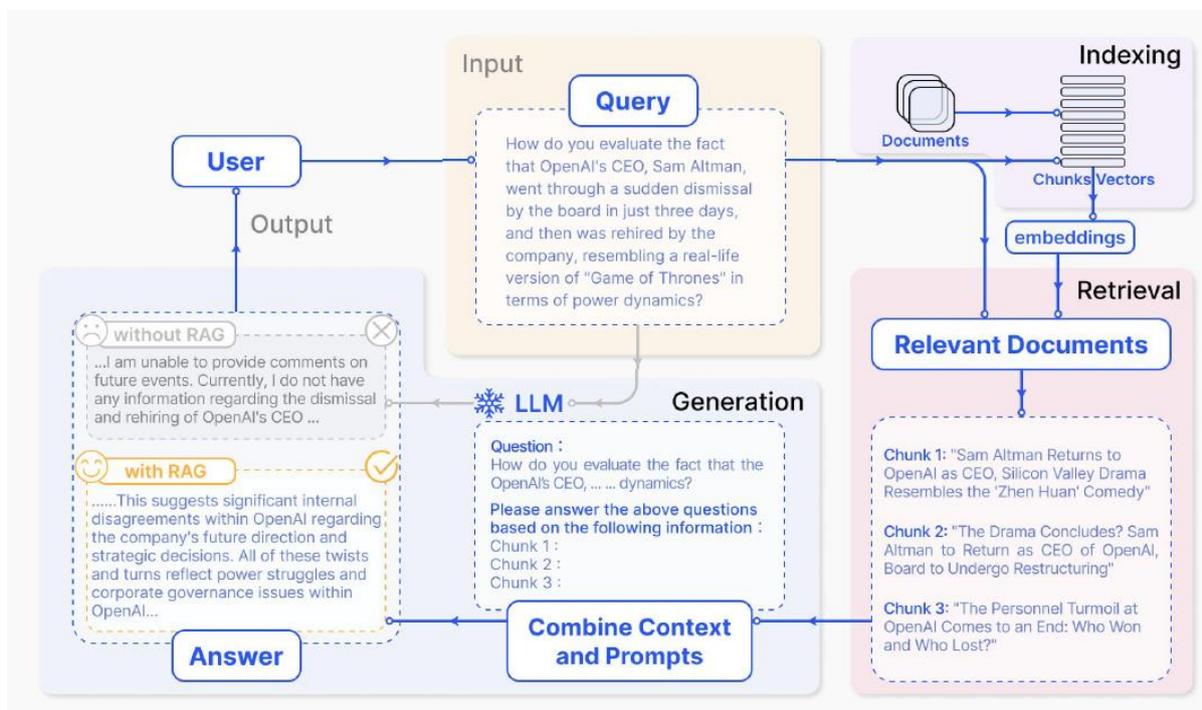
Ερευνητικό ερώτημα	Μεθοδολογική προσέγγιση	Δείκτες μέτρησης
RQ1 Λειτουργική απόδοση	– Τεχνικές δοκιμές με συνόλα εγγράφων, ερωτήματα ανά επαναληψιμότητα εκτελέσεων	– Διαφορετικά Συνοχή απαντήσεων, ακρίβεια σε σχέση με πρωτότυπο κείμενο, χρόνος απόκρισης, ποσοστό hallucinations, αναπαραγωγιμότητα
RQ2 Παιδαγωγική αξία	– Εφαρμογή εκπαιδευτικές εργασίες, επεξήγηση εννοιών, σύννοψη υλικού, παρατήρηση εμπειρίας χρήσης	– Βαθμός κατανόησης, ποιότητα απαντήσεων, χρησιμότητα στη μελέτη, υποστήριξη προετοιμασίας διδασκαλίας
RQ3 Περιορισμοί συστήματος	– Καταγραφή προβλημάτων υλικού	– Υπολογιστικό κόστος, χρόνος εκτέλεση, μέτρηση επιδόσεων, ποιότητα ανάλυση διαδικασίας προετοιμασίας υλικού, παραμετροποίησης, ανομοιογενών εγγράφων, ingestion, embeddings, ανάγκες επιδράσεις

Υλοποίηση

Συνολική Αρχιτεκτονική του Συστήματος

Σε συνέχεια του μεθοδολογικού πλαισίου που παρουσιάστηκε, το παρόν κεφάλαιο εστιάζει στην τεχνική υλοποίηση του προτεινόμενου τοπικού συστήματος Retrieval-Augmented Generation (RAG). Η υλοποίηση σχεδιάστηκε με γνώμονα την πλήρη τοπική εκτέλεση, τη διαχωριστικότητα των λειτουργικών υποσυστημάτων και τη δυνατότητα αναπαραγωγής των αποτελεσμάτων, ώστε να ανταποκρίνεται στις απαιτήσεις ιδιωτικότητας και αξιοπιστίας που αναδείχθηκαν στη βιβλιογραφική ανασκόπηση.

Στο Σχήμα 1, φαίνεται πως η αρχιτεκτονική του συστήματος είναι πολυεπίπεδη και βασίζεται στον σαφή διαχωρισμό μεταξύ της επεξεργασίας δεδομένων, της σημασιολογικής ανάκτησης και της παραγωγής απάντησης. Όλες οι υπολογιστικές διεργασίες πραγματοποιούνται αποκλειστικά σε τοπικό περιβάλλον, χωρίς διασύνδεση με εξωτερικές διαδικτυακές υπηρεσίες, διασφαλίζοντας τον πλήρη έλεγχο των δεδομένων και των παραγόμενων αποτελεσμάτων.



Σχήμα 1. Ένα αντιπροσωπευτικό παράδειγμα της διαδικασίας RAG που εφαρμόζεται στην απάντηση ερωτήσεων Gao et al. [2024]

Σε υψηλό επίπεδο, το σύστημα αποτελείται από τα εξής υποσυστήματα:

- (α) μηχανισμό εισαγωγής και προεπεξεργασίας εγγράφων,
- (β) μηχανισμό δημιουργίας και αποθήκευσης διανυσματικών αναπαραστάσεων,
- (γ) μηχανισμό σημασιολογικής ανάκτησης,
- (δ) γλωσσικό μοντέλο παραγωγής απάντησης,
- (ε) διεπαφή χρήστη και επίπεδο API.

Τεχνολογική Υποδομή και Πλαίσιο PrivateGPT

Η υλοποίηση του συστήματος πραγματοποιήθηκε στο πλαίσιο του PrivateGPT, ένα περιβάλλον ανάπτυξης εφαρμογών RAG που υποστηρίζει πλήρη τοπική εκτέλεση και δίνει έμφαση στην προστασία της ιδιωτικότητας. Η αρχιτεκτονική του PrivateGPT βασίζεται σε δύο διακριτά επίπεδα λειτουργίας: το επίπεδο διεπαφής προγραμματισμού εφαρμογών (Application Programming Interface - API Layer) και επίπεδο επιμέρους συνιστωσών (Components Layer).

Το API Layer λειτουργεί ως το σημείο ενοποίησης των υποσυστημάτων και ορίζει τα τελικά σημεία πρόσβασης (endpoints) μέσω των οποίων ο χρήστης αλληλεπιδρά με το σύστημα. Υλοποιείται με χρήση του FastAPI και ακολουθεί τη λογική των προδιαγραφών διεπαφών OpenAI (OpenAI API specifications), γεγονός που διευκολύνει τη διαχείριση ερωτημάτων και την ανταλλαγή δεδομένων.

Το Components Layer περιλαμβάνει τις επιμέρους τεχνικές υλοποιήσεις: το γλωσσικό μοντέλο, το μοντέλο embeddings και τη βάση διανυσμάτων. Η χρήση αφηρημένων διεπαφών (interfaces) επιτρέπει την αντικατάσταση οποιουδήποτε υποσυστήματος χωρίς τροποποίηση της συνολικής αρχιτεκτονικής, αποτρέποντας φαινόμενα τεχνολογικού εγκλωβισμού (vendor lock-in).

Αγωγός εισαγωγής δεδομένων (Ingestion Pipeline) και προεπεξεργασία εγγράφων

Η διαδικασία εισαγωγής εγγράφων αποτελεί κρίσιμο στάδιο της υλοποίησης, καθώς επηρεάζει άμεσα την ποιότητα της σημασιολογικής ανάκτησης. Τα έγγραφα που εισάγονται

από τον χρήστη υποβάλλονται σε τυποποιημένη διαδικασία προεπεξεργασίας, ανεξαρτήτως μορφής αρχείου (PDF, DOCX, παρουσιάσεις, επιστημονικά άρθρα).

Αρχικά πραγματοποιείται εξαγωγή και ενοποίηση του κειμένου. Ακολουθεί καθαρισμός, κατά τον οποίο αφαιρούνται στοιχεία χωρίς σημασιολογική αξία, όπως επικεφαλίδες, υποσέλιδα και σφάλματα κατά την οπτική αναγνώριση χαρακτήρων (Optical Character Recognition - OCR). Στη συνέχεια εφαρμόζεται τεμαχισμός του κειμένου σε συνεκτικές νοηματικές ενότητες (chunks), με στόχο τη διατήρηση της εννοιολογικής συνοχής και την αποφυγή υπερβολικά μεγάλων τμημάτων.

Κάθε τμήμα κειμένου μετατρέπεται σε διανυσματική αναπαράσταση (embedding) και αποθηκεύεται στη βάση διανυσμάτων μαζί με μεταδεδομένα, όπως το αρχείο προέλευσης και η θέση στο κείμενο. Το αποτέλεσμα είναι η δημιουργία ενός ιδιωτικού σημασιολογικού αποθετηρίου γνώσης.

Μοντέλο Embeddings και βάση διανυσμάτων (Vector Store)

Το μοντέλο embeddings διαδραματίζει κεντρικό ρόλο στη λειτουργία του συστήματος, καθώς κωδικοποιεί τη σημασιολογική πληροφορία των κειμένων σε πυκνά διανύσματα υψηλής διάστασης. Η διαδικασία δημιουργίας embeddings εφαρμόζεται τόσο στα εισαγόμενα έγγραφα όσο και στα ερωτήματα των χρηστών, διασφαλίζοντας ενιαίο σημασιολογικό χώρο αναπαράστασης.

Τα embeddings αποθηκεύονται σε εξειδικευμένη βάση διανυσμάτων, η οποία υποστηρίζει αποδοτικές επιχειρήσεις αναζήτησης με βάση μετρικές ομοιότητας, όπως η συνημιτονική ομοιότητα (cosine similarity). Η ευρετηρίαση συνοδεύεται από μεταδεδομένα, επιτρέποντας ιχνηλασιμότητα και ελεγχόμενη ανάκτηση πληροφοριών. Οι παράμετροι διαδικασία τεμαχισμού (chunking) ρυθμίστηκαν ώστε να επιτυγχάνεται ισορροπία μεταξύ σημασιολογικής συνοχής και υπολογιστικού κόστους.

Μηχανισμός Ανάκτησης και RAG Pipeline

Κατά την υποβολή ενός ερωτήματος από τον χρήστη, ενεργοποιείται ο μηχανισμός σημασιολογικής ανάκτησης. Το ερώτημα μετατρέπεται αρχικά σε embedding και συγκρίνεται με τα διανύσματα της βάσης. Τα πιο συναφή τμήματα επιλέγονται και συγκροτούν το σύνολο συμφραζομένων που θα χρησιμοποιηθεί για την παραγωγή της απάντησης.

Το σύνολο αυτό ενσωματώνεται στο prompt του γλωσσικού μοντέλου, διαμορφώνοντας ένα εμπλουτισμένο πλαίσιο εισόδου. Η αρχιτεκτονική του RAG επιβάλλει τον περιορισμό ότι η παραγόμενη απάντηση βασίζεται αποκλειστικά στα ανακτηθέντα συμφραζόμενα, αποτρέποντας την εισαγωγή εξωγενούς γνώσης. Με τον τρόπο αυτό, το σύστημα λειτουργεί ως σημασιολογικός «μεγεθυντής» του corpus και όχι ως αυτόνομος παραγωγός πληροφοριών.

Διαμόρφωση του Γλωσσικού Μοντέλου (LLM)

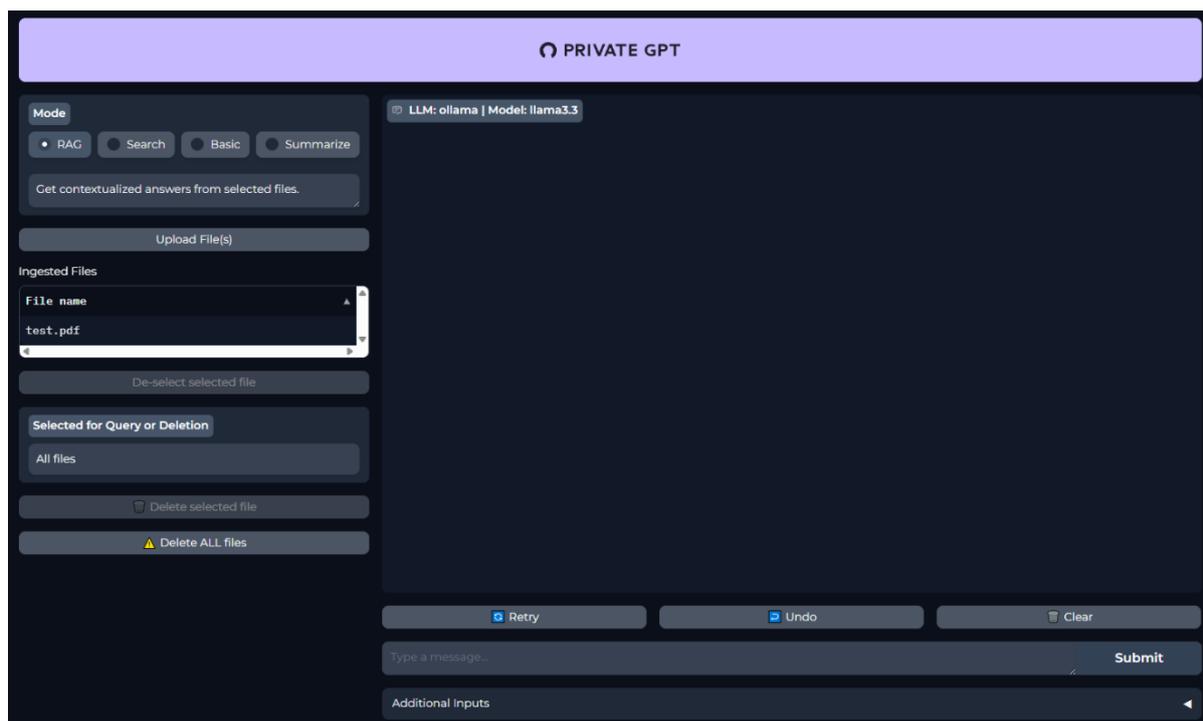
Για την παραγωγή απαντήσεων επιλέχθηκε μοντέλο της οικογένειας LLaMA, το οποίο προσφέρει ικανοποιητική ποιότητα φυσικής γλώσσας σε ελληνικά συμφραζόμενα με σχετικά περιορισμένες υπολογιστικές απαιτήσεις. Το μοντέλο εκτελείται αποκλειστικά σε τοπικό περιβάλλον και ενσωματώθηκε μέσω του LLMComponent.

Η παραμετροποίηση περιλαμβάνει τον καθορισμό του παραθύρου συμφραζομένων, των παραμέτρων δειγματοληψίας (sampling) και της στρατηγικής διαμόρφωσης οδηγιών (prompting). Ιδιαίτερη έμφαση δόθηκε στον περιορισμό της παραγωγής, ώστε οι απαντήσεις να αναγάζονται αποκλειστικά στο παρεχόμενο πλαίσιο συμφραζομένων (context), μειώνοντας δραστικά την πιθανότητα παραισθήσεων.

Διεπαφή Χρήστη και Διασύνδεση Υποσυστημάτων

Η διεπαφή χρήστη υλοποιήθηκε ως βασισμένο στο διαδίκτυο (web-based) περιβάλλον, το οποίο επικοινωνεί με το API Layer μέσω προτυποποιημένων endpoints. Παρέχει δυνατότητα υποβολής ερωτημάτων, διαχείρισης του corpus εγγράφων και οπτικοποίησης των ανακτηθέντων τμημάτων που χρησιμοποιήθηκαν για τη δημιουργία της απάντησης.

Η διεπαφή, που απεικονίζεται στο Σχήμα 2, λειτουργεί ως παιδαγωγικό εργαλείο, καθιστώντας ορατή τη διαδικασία ανάκτησης και παραγωγής γνώσης. Με τον τρόπο αυτό, ο χρήστης δεν λαμβάνει απλώς μια απάντηση, αλλά μπορεί να κατανοήσει τη διαδρομή που ακολουθήθηκε για την παραγωγή της.



Σχήμα 2. Η διεπαφή του Private GPT

Περιορισμοί και Προκλήσεις της Υλοποίησης

Παρά τη λειτουργική πληρότητα του συστήματος, εντοπίζονται ορισμένοι περιορισμοί. Η απόδοση εξαρτάται από τις υπολογιστικές δυνατότητες του τοπικού περιβάλλοντος, ενώ η ανομοιογένεια των εγγράφων μπορεί να επηρεάσει την ποιότητα των embeddings και της ανάκτησης. Επιπλέον, όταν τα ερωτήματα υπερβαίνουν το περιεχόμενο του corpus, οι απαντήσεις ενδέχεται να είναι ελλιπείς.

Οι παράμετροι chunking και top-k αποτελούν κρίσιμους παράγοντες απόδοσης και απαιτούν προσεκτική ρύθμιση. Η αναγνώριση των περιορισμών αυτών παρέχει κατευθύνσεις για μελλοντική βελτιστοποίηση και επέκταση του συστήματος.

Αποτελέσματα και ανάλυση

Το παρόν κεφάλαιο παρουσιάζει και αναλύει τα αποτελέσματα της αξιολόγησης του τοπικού συστήματος LLM-RAG που υλοποιήθηκε στο πλαίσιο της παρούσας εργασίας, βασισμένο σε τοπικό περιβάλλον. Η αξιολόγηση οργανώθηκε σύμφωνα με τα τρία ερευνητικά ερωτήματα που διατυπώθηκαν σε προηγούμενο κεφάλαιο και αποσκοπεί στη διερεύνηση: (α) της λειτουργικής απόδοσης του συστήματος, (β) της παιδαγωγικής του αξίας στο πλαίσιο της

διδασκαλίας αλγορίθμων και δομών δεδομένων και (γ) των περιορισμών και πρακτικών προκλήσεων που προκύπτουν από την εφαρμογή του σε πραγματικές συνθήκες χρήσης.

Η ανάλυση βασίζεται σε συστηματικό πειραματισμό με ποικιλία ερωτημάτων, τα οποία προήλθαν τόσο από αυθεντικό εκπαιδευτικό υλικό όσο και από στοχευμένες δοκιμασίες ελέγχου συνοχής και αξιοπιστίας. Συγκεκριμένα, πραγματοποιήθηκαν δοκιμές επανάληψης ίδιων ερωτήσεων με διαφορετική διατύπωση, ερωτήσεις εκτός του διαθέσιμου corpus, καθώς και ερωτήματα αυξημένης παιδαγωγικής απαίτησης, όπως επεξήγηση, σύνοψη και μετασχηματισμός γνώσης. Η προσέγγιση αυτή επιτρέπει την αποτίμηση του συστήματος όχι μόνο ως τεχνικής υλοποίησης, αλλά και ως λειτουργικού μαθησιακού εργαλείου.

Πίνακας αποτελεσμάτων

Ο Πίνακας 3 συνοψίζει τα βασικά ευρήματα, πριν αυτά αναλυθούν εκτενώς στις επόμενες υποενότητες:

Πίνακας 3. Πίνακας Ευρημάτων

Διάσταση αξιολόγησης	Εμπειρικά Ευρήματα
Λειτουργική Απόδοση (RQ1)	Το σύστημα παρήγαγε συνεκτικές και τεχνικά ορθές απαντήσεις σε όλες τις ερωτήσεις που βασιζονταν σε διαθέσιμο υλικό. Όταν έγινε ερώτηση για μη υπάρχον περιεχόμενο, απάντησε σωστά δηλώνοντας «not found», χωρίς να παράγει παραπλανητική πληροφορία. Ο χρόνος απόκρισης ήταν σταθερός και η αναπαραγωγικότητα των απαντήσεων υψηλή.
Παιδαγωγική Αξία (RQ2)	Οι απαντήσεις σε παιδαγωγικές ερωτήσεις (π.χ. «εξήγησε Bubble Sort», «ποια είναι τα χαρακτηριστικά της δομής») ήταν σαφείς, δομημένες και συμβατές με το διδακτικό υλικό. Το σύστημα έδειξε ικανότητα μετασχηματισμού γνώσης, παραγωγής παραδειγμάτων και καθοδήγησης της μάθησης.
Περιορισμοί (RQ3)	Η ποιότητα εξαρτάται πλήρως από το περιεχόμενο που έχει ενσωματωθεί. Όταν δεν υπήρχε σχετικό υλικό, το σύστημα αρνήθηκε να απαντήσει σωστά, γεγονός που δείχνει ασφάλεια αλλά και εγγενή περιορισμό. Η απάντηση σε ορισμένες αφηρημένες παιδαγωγικές ερωτήσεις παρουσίασε μικρό βαθμό γενίκευσης.

Ο πίνακας λειτουργεί ως συνοπτική αποτύπωση, ενώ τα ευρήματα τεκμηριώνονται εκτενώς στις παρακάτω ενότητες.

Αποτελέσματα ως προς το RQ1 – Λειτουργική Απόδοση

Η αξιολόγηση της λειτουργικής απόδοσης κατέδειξε ότι το τοπικό σύστημα LLM-RAG παρουσίασε υψηλή σταθερότητα, συνοχή και ακρίβεια, ιδίως σε ερωτήσεις που σχετίζονταν άμεσα με το ενσωματωμένο εκπαιδευτικό υλικό. Στο Σχήμα 3, βλέπουμε πως σε δοκιμές με τεχνικά ερωτήματα, όπως η παραγωγή ερωτήσεων αξιολόγησης για αλγορίθμους, το σύστημα παρήγαγε θεματικά σωστές και παιδαγωγικά κατάλληλες απαντήσεις, γεγονός που υποδηλώνει αποτελεσματική σημασιολογική ανάκτηση και επιτυχή ενσωμάτωση του corpus στο RAG pipeline.

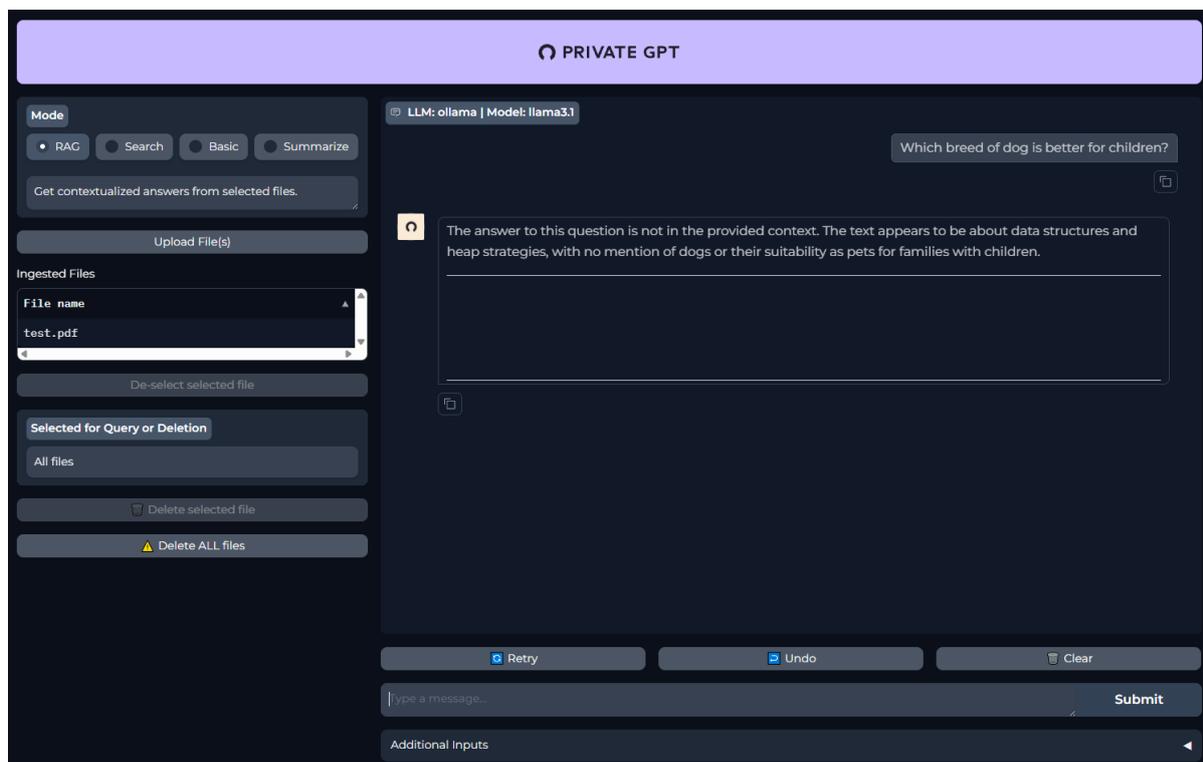
Ιδιαίτερη σημασία έχει το γεγονός ότι σε περιπτώσεις άκυρων ή εκτός θέματος ερωτημάτων, όπως φαίνεται στο Σχήμα 4, το σύστημα δεν επιχείρησε να παραγάγει απάντηση, αλλά δήλωσε ρητά την αδυναμία εύρεσης σχετικής πληροφορίας. Το εύρημα αυτό αποτελεί κρίσιμο δείκτη αξιοπιστίας, καθώς αποδεικνύει ότι ο μηχανισμός RAG λειτουργεί με αυστηρό περιορισμό στο διαθέσιμο περιεχόμενο και δεν οδηγεί σε παραγωγή παραισθήσεων, ένα από τα συχνότερα προβλήματα των γενικής χρήσης LLMs.

The image displays two screenshots of the PRIVATE GPT web interface. The top screenshot shows a user query: "Can you provide me 10 questions from the whole file for an exam and their answers? I want to test my students". The interface includes a sidebar with file management options (Upload, De-select, Delete) and a main chat area. The response from the LLM (ollama | Model: llama3.1) is titled "Exam Questions" and lists 10 multiple-choice questions based on the provided text. The bottom screenshot shows the same interface with the LLM's response titled "Answers", which provides the correct answers for the 10 questions. The interface also includes a "Sources" section at the bottom of the answer, listing the specific pages from the uploaded file (test.pdf) used for the answers.

Σχήμα 3. 10 ερωτήσεις και απαντήσεις

Επιπλέον, οι απαντήσεις ήταν συνεπείς και αναπαραγωγίμες: παρόμοια ερωτήματα παρήγαγαν απαντήσεις ισοδύναμης ποιότητας και δομής, στοιχείο που επιβεβαιώνει τη σταθερότητα της αρχιτεκτονικής και τη σωστή συνεργασία μεταξύ embedding model και

vector store. Ο χρόνος απόκρισης παρέμεινε σταθερός σε όλες τις δοκιμές, χωρίς αισθητές καθυστερήσεις, γεγονός αναμενόμενο για ένα τοπικό σύστημα χωρίς εξωτερικές εξαρτήσεις. Συνολικά, τα αποτελέσματα συνάδουν με τη διεθνή βιβλιογραφία που αναδεικνύει τα πλεονεκτήματα των τοπικών RAG συστημάτων ως προς την ακρίβεια και τον έλεγχο της παραγόμενης γνώσης.



Σχήμα 4. Ερώτηση εκτός θέματος

Αποτελέσματα ως προς το RQ2 – Παιδαγωγική Αξία

Η διερεύνηση της παιδαγωγικής διάστασης του συστήματος επικεντρώθηκε σε ερωτήματα αυξημένης γνωστικής απαίτησης, όπως επεξηγήσεις αλγορίθμων, εννοιολογική ανάλυση και μετασχηματισμό γνώσης. Το σύστημα παρήγαγε απαντήσεις πλήρως συμβατές με το εκπαιδευτικό υλικό, διατηρώντας την ορολογία, τη δομή και την ακρίβεια των αρχικών πηγών.

The image displays two screenshots of the PRIVATE GPT interface, demonstrating its ability to generate pseudocode and provide explanations for the BubbleSort algorithm based on provided context.

Top Screenshot: The user asks, "Give me the pseudocode for BubbleSort". The system responds with a detailed pseudocode block:

```

1) algorithm BubbleSort( list)
2) Pre: list≠∅
3) Post: list has been sorted into values of ascending order
4) for i ← 0 to list.Count - 1
5)   for j ← 0 to list.Count - 1
6)     if list[i] < list[j]
7)       Swap (list[i], list[j])
8)     end if
9)   end for
10) end for
11) return list
12) end BubbleSort
**<hr>Sources:
1. test.pdf (page 74)
2. test.pdf (page 7)
<hr>

```

Bottom Screenshot: The user asks, "Can you explain BubbleSort to me?". The system provides a clear explanation:

According to the context information provided, Bubble Sort can be explained as follows:

Bubble sort is one of the simplest forms of sorting algorithms. It works by comparing each item with every other item in a list and swapping them if they are out of order.

The algorithm involves two loops: an outer loop that runs from 0 to `list.Count - 1`, and an inner loop that also runs from 0 to `list.Count - 1`. Inside the inner loop, it checks each pair of elements at indices `i` and `j` if they are in ascending order. If not, it swaps them.

The process is repeated until no more swaps need to be made (indicating that the list has been sorted), which happens when the outer loop completes its iterations without any swaps being performed.

However, as mentioned in the context information, Bubble Sort has a time complexity of $O(n^2)$ and is not particularly efficient for large lists.

Sources:

1. test.pdf (page 74)

Σχήμα 5. Εξήγηση αλγορίθμου BubbleSort

Το σύστημα παρήγαγε απαντήσεις πλήρως συμβατές με τα αποθηκευμένα εκπαιδευτικά κείμενα, διατηρώντας τη δομή, την ορολογία και την ακρίβεια των ορισμών. Στο Σχήμα 5 βλέπουμε πως στην περίπτωση του αλγορίθμου Bubble Sort, η εξήγηση ήταν πλήρης,

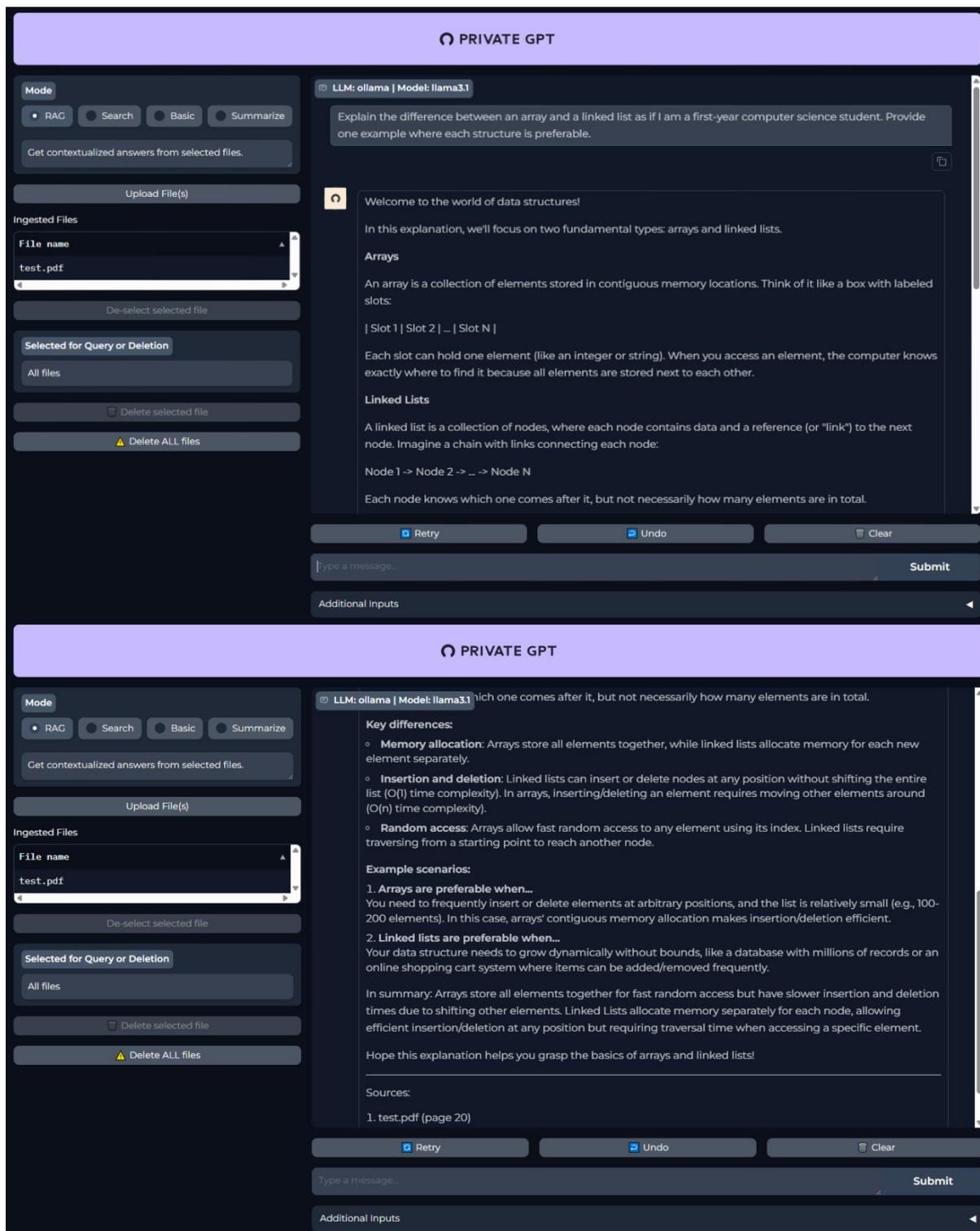
περιλαμβάνοντας όχι μόνο τον βασικό μηχανισμό του αλγορίθμου αλλά και τα χαρακτηριστικά του, όπως η υπολογιστική πολυπλοκότητα και η εκπαιδευτική χρήση του ως παραδείγματος εισαγωγής στην αλγοριθμική σκέψη.

Παράλληλα, στο Σχήμα 6 φαίνεται πως το εργαλείο απέδειξε την ικανότητά του να λειτουργεί ως συμπληρωματικό παιδαγωγικό μέσο: παρείχε παραδείγματα, αναδιατύπωση δύσκολων εννοιών και σύντομες περιλήψεις, οι οποίες μπορούν να υποστηρίξουν τη μελέτη των μαθητών. Η λογοδοσία αυτή της γνώσης σε διαφορετικές μορφές (επεξήγηση, περίληψη, επιπλέον παραδείγματα) συνάδει με τη βιβλιογραφία για τα οφέλη των LLMs στη μάθηση, όπως η ενίσχυση της κατανόησης και η εξατομικευμένη υποστήριξη.

The screenshot shows the PRIVATE GPT interface. At the top, it says "PRIVATE GPT". Below that, there are mode selection buttons: "RAG" (selected), "Search", "Basic", and "Summarize". A text box contains the instruction: "Summarize the QuickSort algorithm in simple terms, preferably in a bullet-point format suitable for lecture slides. Include the main steps and its time complexity." Below this is an "Upload File(s)" button and a list of "Ingested Files" with "test.pdf" listed. There are also buttons for "De-select selected file", "Selected for Query or Deletion" (with "All files" selected), "Delete selected file", and "Delete ALL files". The main chat area shows the response: "Here's a summary of the QuickSort algorithm: QuickSort Algorithm. Purpose: Sort an array of elements in ascending order. Main Steps: Choose a pivot element from the array (e.g., middle element), Partition the array around the pivot, so that all elements less than the pivot are on its left and greater-than or equal to the pivot are on its right, Recursively apply QuickSort to the subarrays of elements less than and greater than the pivot. Key Properties: Divide-and-Conquer approach (divide array into smaller parts, sort each part recursively), In-place sorting (no additional storage needed). Time Complexity: Best-case time complexity: $O(n \log n)$ when the partitioning step is done efficiently and the recursion tree has a balanced structure, Average-case time complexity: $O(n \log n)$, assuming random data distribution, Worst-case time complexity: $O(n^2)$ if the pivot choice leads to poor partitioning, causing an unbalanced recursion tree. Note: QuickSort's performance can be improved by using techniques like: Choosing a good pivot (e.g., median of three), Using insertion sort for small subarrays, Implementing hybrid sorting algorithms that combine QuickSort with other methods. Sources: 1. test.pdf (page 97), 2. test.pdf (page 14). At the bottom, there are "Retry", "Undo", and "Clear" buttons, a "Type a message..." input field, and a "Submit" button. There is also an "Additional Inputs" field at the very bottom.

Σχήμα 6. Εξήγηση QuickSort

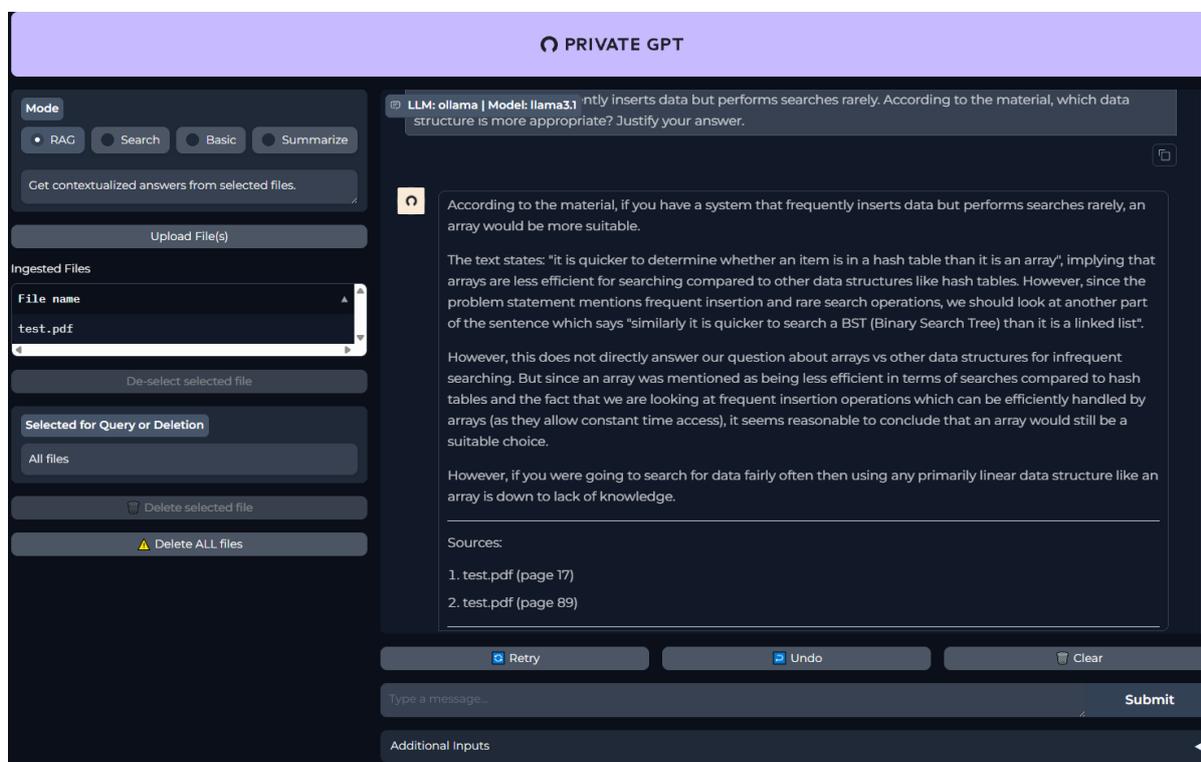
Ένα επιπλέον χαρακτηριστικό του συστήματος που αναπτύχθηκε είναι η δυνατότητά του να ανακτά πληροφορίες από πολλαπλές θέσεις εντός του αρχείου και να τις συγκρίνει προκειμένου να παράσχει ολοκληρωμένη και συνεκτική απάντηση, κάτι το οποίο φαίνεται στο Σχήμα 7. Αυτή η ικανότητα επιτρέπει στο σύστημα όχι μόνο να συνθέτει πληροφορίες από διαφορετικά τμήματα του corpus, αλλά και να αξιολογεί τυχόν διαφοροποιήσεις ή συμπληρωματικά στοιχεία, ενισχύοντας την ακρίβεια και την πληρότητα των απαντήσεων.



Σχήμα 7. Σύγκριση Array και List

Μια σημαντική δυνατότητα του συστήματος είναι η ικανότητά του να αναλύει τις πληροφορίες που περιέχονται στα αρχεία με βάση τις οδηγίες που παρέχει ο χρήστης. Στο Σχήμα 8 , αφού περιέγραψα τις ανάγκες, το σύστημα εξέτασε το περιεχόμενο, συνέκρινε σχετικά τμήματα και συνδύασε τα διαθέσιμα στοιχεία προκειμένου να παράσχει μια

στοχευμένη και ακριβή απάντηση. Η διαδικασία αυτή αναδεικνύει την ικανότητα του συστήματος να αξιοποιεί πλήρως το corpus, όχι μόνο για ανάκτηση πληροφοριών, αλλά και για σύνθεση και επεξεργασία δεδομένων ώστε να ανταποκρίνεται στις συγκεκριμένες απαιτήσεις του χρήστη.



Σχήμα 8. Κατάλληλη δομή δεδομένων

Ωστόσο, πρέπει να σημειωθεί ότι η παιδαγωγική αξία προκύπτει αποκλειστικά από την ποιότητα του ενσωματωμένου υλικού. Το σύστημα δεν παράγει νέες παιδαγωγικές γνώσεις πέρα από το corpus, αλλά δύνανται να επεξεργαστεί, να αναδιατυπώσει και να ανακαλέσει τις υπάρχουσες με υψηλή πιστότητα. Αυτό αποτελεί ταυτόχρονα πλεονέκτημα και περιορισμό.

Αποτελέσματα ως προς το RQ3 – Περιορισμοί και Προκλήσεις

Η τρίτη φάση της αξιολόγησης ανέδειξε σειρά πρακτικών περιορισμών που σχετίζονται κυρίως με τη φύση της τοπικής εκτέλεσης. Ο σημαντικότερος περιορισμός αφορά τις απαιτήσεις σε υπολογιστικούς πόρους, καθώς η επεξεργασία μεγάλων ή σύνθετων ερωτημάτων μπορεί να οδηγήσει σε επιβράδυνση της απόκρισης ή προσωρινή αστάθεια του συστήματος, ανάλογα με τις δυνατότητες του τοπικού περιβάλλοντος.

Επιπλέον, σε περιπτώσεις όπου ζητήθηκε αφηρημένη γενίκευση εννοιών που δεν υπήρχαν στο corpus, το σύστημα παρήγαγε περιορισμένες ή μη ικανοποιητικές απαντήσεις. Το εύρημα αυτό είναι αναμενόμενο για ένα αυστηρά τοπικό σύστημα χωρίς πρόσβαση σε γενικής χρήσης εξωτερική γνώση και επιβεβαιώνει ότι η αποτελεσματικότητά του μεγιστοποιείται όταν λειτουργεί εντός σαφώς ορισμένων ορίων περιεχομένου.

Τέλος, η ποιότητα της ανάκτησης επηρεάζεται σημαντικά από τη δομή και την ομοιογένεια των εγγράφων. Ανομοιογενές ή αποσπασματικό υλικό μειώνει την ποιότητα των embeddings και κατ' επέκταση την ακρίβεια των απαντήσεων. Οι περιορισμοί αυτοί δεν αναιρούν τη χρησιμότητα του συστήματος, αλλά οριοθετούν το πλαίσιο εντός του οποίου μπορεί να αξιοποιηθεί αποτελεσματικά και παρέχουν σαφή κατεύθυνση για μελλοντικές βελτιώσεις.

Συζήτηση και συμπεράσματα

Συζήτηση των αποτελεσμάτων

Η παρούσα διπλωματική εργασία είχε ως στόχο τη διερεύνηση της δυνατότητας αξιοποίησης ενός τοπικά εκτελούμενου συστήματος Μεγάλων Γλωσσικών Μοντέλων (LLMs), βασισμένου σε αρχιτεκτονική Retrieval-Augmented Generation (RAG), ως εκπαιδευτικού εικονικού βοηθού. Η συζήτηση των αποτελεσμάτων οργανώνεται σε συνάρτηση με τα ερευνητικά ερωτήματα που τέθηκαν στο Κεφάλαιο 3, ενώ τα ευρήματα τοποθετούνται σε συνάφεια με σχετικές προσεγγίσεις της σύγχρονης βιβλιογραφίας.

Σε σχέση με το RQ1 (λειτουργική απόδοση), τα αποτελέσματα δείχνουν ότι το σύστημα παρουσίασε υψηλή σταθερότητα και ακρίβεια, εφόσον τα ερωτήματα βασιζόνταν στο ενσωματωμένο εκπαιδευτικό corpus. Η αυστηρή σύνδεση της παραγωγής απαντήσεων με τα ανακτημένα αποσπάσματα γνώσης λειτούργησε αποτελεσματικά ως μηχανισμός αποτροπής παραισθήσεων, στοιχείο που διαφοροποιεί το προτεινόμενο σύστημα από γενικής χρήσης cloud-based LLMs. Το εύρημα αυτό ευθυγραμμίζεται με τη διεθνή βιβλιογραφία, όπου η αρχιτεκτονική RAG αναδεικνύεται ως βασική προσέγγιση για τη βελτίωση της αξιοπιστίας και της επαληθευσσιμότητας των απαντήσεων σε εκπαιδευτικά περιβάλλοντα.

Όσον αφορά το RQ2 (παιδαγωγική αξία), το σύστημα απέδειξε ότι μπορεί να λειτουργήσει ως ουσιαστικός γνωσιακός συνεργάτης. Η ικανότητά του να εξηγεί έννοιες, να αναδιατυπώνει σύνθετο περιεχόμενο, να συγκρίνει δομές δεδομένων και να παράγει παραδείγματα βασισμένα αποκλειστικά στο διδακτικό υλικό υποστηρίζει τη μάθηση με τρόπο διαφανή και

ελεγχόμενο. Ιδιαίτερα σημαντικό είναι το γεγονός ότι το σύστημα δεν αντικαθιστά τη μαθησιακή διαδικασία, αλλά τη διευκολύνει, ενισχύοντας την κατανόηση και την αυτοκατευθυνόμενη μελέτη. Το εύρημα αυτό συνάδει με μελέτες που αντιμετωπίζουν τα LLMs ως εργαλεία υποστήριξης της μάθησης και όχι ως αυτόνομους εκπαιδευτές.

Σε σχέση με το RQ3 (περιορισμοί και προκλήσεις), επιβεβαιώνεται ότι η αποτελεσματικότητα του συστήματος εξαρτάται άμεσα από την ποιότητα, τη δομή και την πληρότητα του corpus. Η αυστηρή τοπική εκτέλεση, αν και ενισχύει την ιδιωτικότητα και τον έλεγχο των δεδομένων, περιορίζει τη δυνατότητα γενίκευσης και πρόσβασης σε εξωτερική γνώση. Παράλληλα, οι απαιτήσεις σε υπολογιστικούς πόρους αποτελούν έναν πρακτικό περιορισμό, ιδιαίτερα σε περιβάλλοντα με περιορισμένες τεχνικές υποδομές.

Πίνακας 4: Συγκριτική ανάλυση αρχιτεκτονικών χαρακτηριστικών εκπαιδευτικών chatbots

Σύστημα Μελέτη	Αρχιτεκτονική	Εκτέλεση (Deployment)	Ιδιωτικότητα Δεδομένων	Κόστος Λειτουργίας	Διαχείριση Παραισθησεων (Hallucinations)
Προτεινόμενο Σύστημα	Local RAG	100% Τοπικά (Offline)	Μέγιστη (Local Sovereign)	Μηδενικό (Open Source)	Αυστηρός Περιορισμός Πλαισίου (Strict Context)
NOVA (León- Paredes et al., RAG Assistant 2025)		Hybrid Cloud	Μέτρια / (εξάρτηση από APIs)	Μεταβλητό (API calls)	Μηχανισμός RAG
Socratic Chatbot (Favero et al., 2024)	Fine-tuned LLM (Llama-2)	Τοπικά (Local)	Μέγιστη	Μηδενικό (Open Weights)	Σωκρατική Μέθοδος (μέσω Fine-tuning)
EduBot (Li et al., 2024)	Hybrid (Fine- tuning + RAG)	Cloud / Server	Μέτρια	Υψηλό (Training compute)	Βάσει Αναλυτικού Προγράμματος (Curriculum- driven)

Σύστημα Μελέτη	/ Αρχιτεκτονική	Εκτέλεση (Deployment)	Ιδιωτικότητα Δεδομένων	Κόστος Λειτουργίας	Διαχείριση Παραισθήσεων (Hallucinations)
Cipherbot (Salminen et al., 2024)	API Wrapper (GPT-based)	Cloud (SaaS)	Χαμηλή (Third-party)	Κόστος ανά token	Εξαρτάται από το βασικό μοντέλο (Black box)

Σε αυτό το σημείο κρίνεται σκόπιμη η τοποθέτηση των παραπάνω ευρημάτων σε συγκριτικό πλαίσιο με υφιστάμενες προσεγγίσεις της βιβλιογραφίας, όπως παρουσιάζονται στον Πίνακα 4, που αφορούν εκπαιδευτικά chatbots και βοηθούς βασισμένους σε LLMs. Πολλές σύγχρονες υλοποιήσεις στηρίζονται είτε σε εμπορικά APIs (wrappers), είτε σε διαδικασίες fine-tuning μεγάλων γλωσσικών μοντέλων. Προσεγγίσεις όπως το Cipherbot βασίζονται σε cloud-based μοντέλα γενικής χρήσης, γεγονός που περιορίζει τον έλεγχο της γνώσης και εγείρει ζητήματα ιδιωτικότητας και κόστους. Αντίστοιχα, συστήματα όπως το EduBot συνδυάζουν fine-tuning και RAG, επιτυγχάνοντας υψηλό βαθμό προσαρμογής, αλλά με αυξημένες απαιτήσεις σε υπολογιστικούς πόρους και διαδικασίες επανεκπαίδευσης.

Σε σύγκριση με το Socratic Chatbot των Favero et al., το οποίο επιτυγχάνει τοπική εκτέλεση μέσω fine-tuning, η παρούσα εργασία υιοθετεί μια διαφορετική φιλοσοφία συντηρησιμότητας. Ενώ το fine-tuning επιτρέπει την ενσωμάτωση παιδαγωγικών στρατηγικών στο ίδιο το μοντέλο, καθιστά τη διαδικασία ενημέρωσης της γνώσης πιο απαιτητική. Αντιθέτως, η RAG αρχιτεκτονική που προτείνεται εδώ επιτρέπει τη δυναμική ανανέωση του εκπαιδευτικού περιεχομένου με απλή εισαγωγή νέων εγγράφων, χωρίς τροποποίηση των παραμέτρων του μοντέλου. Το χαρακτηριστικό αυτό καθιστά την προσέγγιση ιδιαίτερα κατάλληλη για ακαδημαϊκά περιβάλλοντα όπου η ύλη εξελίσσεται.

Παράλληλα, σε αντίθεση με υβριδικές ή cloud-based προσεγγίσεις όπως το NOVA, η παρούσα υλοποίηση δίνει έμφαση στην πλήρη τοπική εκτέλεση σε καταναλωτικό υλικό, χωρίς εξάρτηση από εξωτερικές υπηρεσίες. Η επιλογή αυτή ενισχύει την τεχνολογική κυριαρχία και την προστασία των δεδομένων, αλλά συνοδεύεται από περιορισμούς ως προς την υπολογιστική κλιμάκωση. Ωστόσο, για το εξεταζόμενο εκπαιδευτικό πλαίσιο, τα αποτελέσματα δείχνουν ότι ο συγκεκριμένος συμβιβασμός είναι λειτουργικά αποδεκτός.

Στον τομέα της ακρίβειας, η επιλογή ντετερμινιστικών μηχανισμών ανάκτησης και αυστηρού περιορισμού πλαισίου διαφοροποιεί το προτεινόμενο σύστημα από πιθανοτικές ή στοχαστικές προσεγγίσεις RAG. Η στρατηγική αυτή αποδείχθηκε ιδιαίτερα σημαντική για αντικείμενα όπως οι αλγόριθμοι και οι δομές δεδομένων, όπου η ανακρίβεια ή η «δημιουργική» απόκλιση μπορεί να οδηγήσει σε λανθασμένη κατανόηση. Με τον τρόπο αυτό, το σύστημα προσεγγίζει την αξιοπιστία ενός κλασικού εγχειριδίου, διατηρώντας ταυτόχρονα τη διαδραστικότητα ενός σύγχρονου ψηφιακού βοηθού.

Συνολικά, η συγκριτική ανάλυση δείχνει ότι η επιλογή μιας τοπικής αρχιτεκτονικής RAG αποτελεί έναν συνειδητό λειτουργικό συμβιβασμό μεταξύ ακρίβειας, παιδαγωγικής αξιοπιστίας και περιορισμένης ευελιξίας. Ο συμβιβασμός αυτός είναι ιδιαίτερα κατάλληλος για εκπαιδευτικά περιβάλλοντα με αυξημένες απαιτήσεις απορρήτου, ελέγχου και διαφάνειας.

Συμπεράσματα

Με βάση τα ευρήματα της παρούσας έρευνας, προκύπτει ότι η ανάπτυξη και αξιοποίηση ενός τοπικού εκπαιδευτικού εικονικού βοηθού βασισμένου σε αρχιτεκτονική Retrieval-Augmented Generation είναι τεχνικά εφικτή και παιδαγωγικά αξιοποιήσιμη. Το σύστημα που υλοποιήθηκε απέδειξε ότι μπορεί να παρέχει αξιόπιστες, τεκμηριωμένες και συνεκτικές απαντήσεις, εφόσον η γνώση περιορίζεται σε επιμελημένο εκπαιδευτικό υλικό.

Η βασική συνεισφορά της εργασίας έγκειται στην ανάδειξη ενός εναλλακτικού μοντέλου χρήσης των LLMs στην εκπαίδευση, το οποίο δεν βασίζεται σε εξωτερικά APIs ή cloud υπηρεσίες, αλλά λειτουργεί αποκλειστικά σε τοπικό περιβάλλον. Το μοντέλο αυτό απαντά σε σύγχρονες ανησυχίες που σχετίζονται με την προστασία προσωπικών δεδομένων, την παιδαγωγική αξιοπιστία και τον έλεγχο της γνώσης.

Επιπλέον, η εργασία αναδεικνύει ότι τα LLMs μπορούν να λειτουργήσουν αποτελεσματικά ως εργαλεία υποστήριξης της μάθησης, όταν εντάσσονται σε σαφώς ορισμένες αρχιτεκτονικές και παιδαγωγικά πλαίσια. Η χρήση τους δεν αναιρεί τον ρόλο του εκπαιδευτικού, αλλά μπορεί να ενισχύσει τη διδασκαλία και τη μελέτη, προσφέροντας εξατομικευμένη υποστήριξη και άμεση πρόσβαση σε εκπαιδευτικό περιεχόμενο.

Περιορισμοί της έρευνας

Παρά τα θετικά αποτελέσματα που προέκυψαν από την παρούσα μελέτη, είναι απαραίτητο να επισημανθούν ορισμένοι περιορισμοί που επηρεάζουν την ερμηνεία και τη γενικευσιμότητα των ευρημάτων. Καταρχάς, η αξιολόγηση του συστήματος πραγματοποιήθηκε με βάση συγκεκριμένο και περιορισμένο εκπαιδευτικό υλικό, το οποίο αφορούσε κυρίως αλγορίθμους και δομές δεδομένων. Ως εκ τούτου, τα συμπεράσματα της έρευνας δεν μπορούν να γενικευθούν άκριτα σε άλλα γνωστικά αντικείμενα με διαφορετικά χαρακτηριστικά, όπως μαθήματα θεωρητικού ή γλωσσικού περιεχομένου, χωρίς περαιτέρω εμπειρική διερεύνηση.

Επιπλέον, η αποτίμηση της παιδαγωγικής αξίας του συστήματος βασίστηκε κυρίως σε ποιοτική ανάλυση της λειτουργικότητας και της χρηστικής εμπειρίας, μέσα από στοχευμένα σενάρια χρήσης. Δεν πραγματοποιήθηκε συστηματική πειραματική αξιολόγηση με συμμετοχή μεγάλου αριθμού μαθητών ή εκπαιδευτικών, ούτε συγκριτική μελέτη με εναλλακτικά εργαλεία ή παραδοσιακές μεθόδους διδασκαλίας. Ο περιορισμός αυτός καθιστά τα παιδαγωγικά συμπεράσματα ενδεικτικά και όχι απολύτως τεκμηριωμένα με ποσοτικούς δείκτες μάθησης.

Ένας ακόμη σημαντικός περιορισμός αφορά τη φύση της τοπικής εκτέλεσης του συστήματος. Παρότι η επιλογή αυτή ενισχύει ουσιαστικά την ιδιωτικότητα και τον έλεγχο των δεδομένων, καθιστά την απόδοση του συστήματος άμεσα εξαρτώμενη από τις διαθέσιμες υπολογιστικές υποδομές. Σε περιβάλλοντα με περιορισμένους πόρους, η απόκριση του συστήματος ενδέχεται να είναι βραδύτερη ή να μην επιτρέπει την αξιοποίηση μεγαλύτερων γλωσσικών μοντέλων. Τέλος, η αποτελεσματικότητα της αρχιτεκτονικής RAG αποδείχθηκε άρρηκτα συνδεδεμένη με την ποιότητα, τη δομή και την επιμέλεια του εισαγόμενου εκπαιδευτικού υλικού. Ανομοιογενή, αποσπασματικά ή ελλιπώς οργανωμένα δεδομένα μπορούν να επηρεάσουν αρνητικά τη σημασιολογική ανάκτηση και, κατ' επέκταση, την ακρίβεια των παραγόμενων απαντήσεων.

Προτάσεις για περαιτέρω μελέτη

Λαμβάνοντας υπόψη τους παραπάνω περιορισμούς, προκύπτουν σημαντικές δυνατότητες για περαιτέρω ερευνητική εμβάθυνση και επέκταση της παρούσας εργασίας. Μία προφανής κατεύθυνση αφορά την εφαρμογή και αξιολόγηση της προτεινόμενης αρχιτεκτονικής σε διαφορετικά γνωστικά αντικείμενα και εκπαιδευτικά πλαίσια, ώστε να διερευνηθεί η γενικευσιμότητα των ευρημάτων. Η αξιοποίηση τοπικών συστημάτων RAG σε μαθήματα

θεωρητικού χαρακτήρα, στη γλωσσική εκπαίδευση ή στις φυσικές επιστήμες θα μπορούσε να αναδείξει νέες παιδαγωγικές χρήσεις αλλά και διαφορετικούς περιορισμούς.

Παράλληλα, μελλοντικές έρευνες θα μπορούσαν να ενσωματώσουν συστηματικές εμπειρικές μεθοδολογίες, όπως πειραματικά ή ημιπειραματικά σχέδια, με τη συμμετοχή μαθητών και εκπαιδευτικών. Μέσα από τη σύγκριση ομάδων ελέγχου και πειραματικών ομάδων, θα ήταν δυνατό να μετρηθεί πιο αντικειμενικά η επίδραση ενός τοπικού LLM-RAG συστήματος στη μαθησιακή επίδοση, στην κατανόηση εννοιών και στην ανάπτυξη δεξιοτήτων αυτορρυθμιζόμενης μάθησης.

Επιπλέον, ιδιαίτερο ενδιαφέρον παρουσιάζει η περαιτέρω διερεύνηση της παιδαγωγικής ενσωμάτωσης τέτοιων συστημάτων. Η σύνδεση της αρχιτεκτονικής RAG με συγκεκριμένες διδακτικές στρατηγικές, όπως ο σωκρατικός διάλογος, η σταδιακή καθοδήγηση ή η παροχή ανατροφοδότησης πολλαπλών επιπέδων, θα μπορούσε να ενισχύσει τον ρόλο του συστήματος ως ενεργού μαθησιακού υποστηρικτή. Σε τεχνολογικό επίπεδο, μελλοντικές μελέτες θα μπορούσαν να εξετάσουν τη βελτιστοποίηση των μηχανισμών ανάκτησης, τη χρήση υβριδικών τεχνικών αναζήτησης ή την προσαρμογή των embeddings στο εκάστοτε γνωστικό αντικείμενο.

Τέλος, μία ακόμη ερευνητική προοπτική αφορά τη διερεύνηση συνεργατικών και κοινωνικών διαστάσεων της χρήσης τοπικών LLM συστημάτων στην εκπαίδευση. Η μελέτη σεναρίων όπου το σύστημα λειτουργεί ως κοινό υποστηρικτικό εργαλείο σε ομαδοσυνεργατικές δραστηριότητες θα μπορούσε να προσφέρει νέα δεδομένα σχετικά με τον ρόλο της τεχνητής νοημοσύνης στη συλλογική οικοδόμηση της γνώσης και στη διαμεσολάβηση της μαθησιακής διαδικασίας.

Συμπερασματικός Επίλογος και Συμβολή της Εργασίας

Η παρούσα εργασία διερεύνησε τη δυνατότητα αξιοποίησης των Μεγάλων Γλωσσικών Μοντέλων στην εκπαίδευση μέσα από την ανάπτυξη και αξιολόγηση ενός τοπικού συστήματος LLM βασισμένου στην αρχιτεκτονική Retrieval-Augmented Generation (RAG). Κεντρικός άξονας της μελέτης υπήρξε η αναζήτηση μιας ισορροπίας ανάμεσα στη λειτουργική ακρίβεια, την παιδαγωγική χρησιμότητα και τη διασφάλιση της ιδιωτικότητας των δεδομένων, στοιχεία που συχνά βρίσκονται σε ένταση στα σύγχρονα συστήματα γενετικής τεχνητής νοημοσύνης.

Τα αποτελέσματα της έρευνας έδειξαν ότι ένα τοπικά εκτελούμενο σύστημα LLM-RAG μπορεί να παρέχει αξιόπιστες, συνεκτικές και παιδαγωγικά χρήσιμες απαντήσεις, εφόσον περιορίζεται αυστηρά στο εκπαιδευτικό corpus του χρήστη. Η λειτουργική αξιολόγηση ανέδειξε την αποτελεσματικότητα της σημασιολογικής ανάκτησης στην αποφυγή παραισθήσεων και στην παραγωγή απαντήσεων με υψηλή πιστότητα στο πρωτογενές υλικό, ενώ η παιδαγωγική ανάλυση κατέδειξε ότι το σύστημα μπορεί να λειτουργήσει ως γνωσιακός συνεργάτης, υποστηρίζοντας διαδικασίες κατανόησης, αναδιατύπωσης και σύνθεσης της γνώσης.

Η κύρια συμβολή της εργασίας έγκειται, αφενός, στην τεκμηριωμένη παρουσίαση μιας εναλλακτικής προσέγγισης στη χρήση των LLMs στην εκπαίδευση, η οποία απομακρύνεται από τα κλειστά, διαδικτυακά και μη ελεγχόμενα μοντέλα γενικής χρήσης. Αντίθετα, προτείνεται ένα σύστημα που λειτουργεί εντός τοπικής υποδομής, προσφέροντας πλήρη έλεγχο στο περιεχόμενο, στα δεδομένα και στη συμπεριφορά του μοντέλου. Με τον τρόπο αυτό, η εργασία συμβάλλει ουσιαστικά στη συζήτηση γύρω από την ηθική, τη διαφάνεια και την ασφάλεια στη χρήση της τεχνητής νοημοσύνης σε εκπαιδευτικά περιβάλλοντα.

Αφετέρου, η μελέτη αναδεικνύει τον παιδαγωγικό ρόλο που μπορούν να διαδραματίσουν τα LLMs όταν ενσωματώνονται με τρόπο που υπηρετεί τη μάθηση και όχι την αυτοματοποιημένη παραγωγή περιεχομένου. Η αυστηρή εξάρτηση του συστήματος από το εκπαιδευτικό υλικό του χρήστη μετατρέπει την τεχνητή νοημοσύνη από πηγή «έτοιμων απαντήσεων» σε εργαλείο διερεύνησης, ενίσχυσης της κατανόησης και υποστήριξης της αυτοκατευθυνόμενης μάθησης. Το εύρημα αυτό έχει ιδιαίτερη σημασία για την εκπαιδευτική πράξη, καθώς υποδεικνύει ότι η παιδαγωγική αξία των LLMs δεν προκύπτει από το μέγεθος ή την ισχύ τους, αλλά από τον τρόπο ενσωμάτωσής τους στο μαθησιακό πλαίσιο.

Συνολικά, η εργασία καταδεικνύει ότι τα τοπικά συστήματα LLM-RAG μπορούν να αποτελέσουν μια ρεαλιστική και παιδαγωγικά βιώσιμη λύση για την υποστήριξη της διδασκαλίας και της μάθησης, ιδίως σε περιβάλλοντα όπου η ιδιωτικότητα, η ακρίβεια και ο έλεγχος της πληροφορίας αποτελούν κρίσιμες προϋποθέσεις. Παρά τους περιορισμούς που αναγνωρίστηκαν, η προτεινόμενη προσέγγιση θέτει τα θεμέλια για περαιτέρω έρευνα και ανάπτυξη εκπαιδευτικών εφαρμογών τεχνητής νοημοσύνης που εστιάζουν όχι μόνο στην τεχνολογική καινοτομία, αλλά και στη ουσιαστική παιδαγωγική τους αξία.

Βιβλιογραφικές αναφορές

Alario-Hoyos, C., et al. (2024). Tailoring your code companion: Leveraging LLMs and RAG to develop a chatbot to support students in a programming course. *Proceedings of the IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 1-8. <https://doi.org/10.1109/tale62452.2024.10834365>

Aldhafeeri, L., Aljumah, F., Thabyan, F., Alabbad, M., AlShahrani, S., Alanazi, F., & Al-Nafjan, A. (2025). Generative AI Chatbots Across Domains: A Systematic Review. *Applied Sciences*, 15(20), 11220. <https://doi.org/10.3390/app152011220>

Alkaissi, H., & McFarlane, S. I. (2023). Artificial hallucinations in ChatGPT: Implications in scientific writing. *Cureus Journal of Medical Science*, 15(2). doi: 10.7759/cureus.35179

Amarnath, N. S., & Nagarajan, R. (2024). An intelligent retrieval augmented generation chatbot for contextually-aware conversations to guide high school students. *Proceedings of the 4th International Conference on Sustainable Expert Systems (International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems - ICSES 2024)*. 1393-1398. 10.1109/ICSES63445.2024.10762977.

Brünner, B., Ebner, M., & Schön, S. (2026). Exploring GenAI chatbots in MOOCs: Analyzing student interactions and self-regulated learning behaviors. In E. Hamonic & R. Sharrock (Eds.), *Digital education: Shaping sustainable lifelong learning for all in the era of AI (Lecture Notes in Computer Science)*. Springer. vol 15733 pp 14-24. Springer, Cham. https://doi.org/10.1007/978-3-032-00056-9_2

Choi, H., Lee, D., Kang, Yk. et al. Empowering PET imaging reporting with retrieval-augmented large language models and reading reports database: a pilot single center study. *European Journal of Nuclear Medicine and Molecular Imaging* 52, 2452-2462 (2025). <https://doi.org/10.1007/s00259-025-07101-9>

Coen, E., Del Fiol, G., Kaphingst, K. A., Borsato, E., Shannon, J., Smith, H., Masino, A., & Allen, C. G. (2025). Chatbot for the return of positive genetic screening results for hereditary cancer syndromes: Prompt engineering project. *Journal of Medical Internet Research (JMIR) Cancer*. doi: 10.2196/65848. PMID: 40493514; PMCID: PMC12172806.

Dahal, P., Nugroho, S., Schmidt, C., & Sängler, V. (2025). AI-based learning recommendations: Use in higher education. *Future Internet*. 17(7), 285. <https://doi.org/10.3390/fi17070285>

Ehrlich-Sommer, F., Eberhard, B., & Holzinger, A. (2025). ForestGPT and beyond: A trustworthy domain-specific large language model paving the way to Forestry 5.0. *Electronics*, 14(18), 3583. <https://doi.org/10.3390/electronics14183583>

Farah, J. C., Ingram, S., Spaenlehauer, B., Lasne, F. K.-L., & Gillet, D. (2024). Prompting large language models to power educational chatbots. In *Artificial intelligence in education* (pp. 169-188). Springer. https://doi.org/10.1007/978-981-99-8385-8_14

- Favero, L. D., Pérez-Ortiz, J. A., Käser, T., & Oliver, N. (2024). Enhancing critical thinking in education by means of a Socratic chatbot. *arXiv*. <https://doi.org/10.48550/arXiv.2409.05511>
- Fink, R. W., Russe, M. F., Mücke, B., Aubele, P., Kock, M., Eickelberg, L., & Wurnig, M. C. (2025). Retrieval-augmented generation improves precision and trust of a GPT-4 model for emergency radiology diagnosis and classification: A proof-of-concept study. *European Radiology*. 2025 Aug;35(8):5091-5098. doi: 10.1007/s00330-025-11445-z. Epub 2025 Feb 14. PMID: 39953150; PMCID: PMC12226682.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *IEEE Transactions on Audio, Speech, and Language Processing*. <https://doi.org/10.48550/arXiv.2312.10997>
- Ho, C. M., Guan, S., Mok, P. K., Lam, C. H., Ho, W. Y., Mak, C. H., Qin, H., Wong, A. K., & Hui, V. (2025). Development and validation of a large-language-model-powered chatbot for neurosurgery: Mixed methods study on enhancing perioperative patient education. *Journal of Medical Internet Research (JMIR)*. 2025 Jul 15;27:e74299. doi: 10.2196/74299. PMID: 40663377; PMCID: PMC12308165.
- Ieva, S., Loconte, D., Loseto, G., Ruta, M., Scioscia, F., Marche, D., & Notarnicola, M. (2024). A retrieval-augmented generation approach for data-driven energy infrastructure digital twins. *Smart Cities*. 7(6), 3095-3120. <https://doi.org/10.3390/smartcities7060121>
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*. <https://doi.org/10.1145/3571730>
- Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- Khasanova Zafar kizi, M., & Suh, Y. (2025). Design and performance evaluation of LLM-based RAG pipelines for chatbot services in international student admissions. *Electronics*. 14(15), 3095. <https://doi.org/10.3390/electronics14153095>
- Khan, U. H., Khan, M. H., & Ali, R. (2025). Large language model-based educational virtual assistant using RAG framework. *Procedia Computer Science*. 252. 905-911. 10.1016/j.procs.2025.01.051.
- Lewis, Patrick & Perez, Ethan & Piktus, Aleksandara & Petroni, Fabio & Karpukhin, Vladimir & Goyal, Naman & Küttler, Heinrich & Lewis, Mike & Yih, Wen-tau & Rocktäschel, Tim & Riedel, Sebastian & Kiela, Douwe. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 10.48550/arXiv.2005.11401.
- Swacha, J., & Gracel, M. (2025). Retrieval-augmented generation (RAG) chatbots for education: A survey of applications. *Applied Sciences*. 15(8), 4234. <https://doi.org/10.3390/app15084234>

Villegas-Ch, W., Govea, J., & Gutierrez, R. (2024). Optimizing language model-based educational assistants using knowledge graphs: Integration with Moodle LMS. *IEEE Access*. vol. 12, pp. 191994-192012, 2024 <https://doi.org/10.1109/ACCESS.2024.3518952>

Wang, K., Ramos, J., & Lawrence, R. (2023). ChatEd: A chatbot leveraging ChatGPT for an enhanced learning experience in higher education. <https://doi.org/10.48550/arXiv.2401.00052>

Zamani, H., & Bendersky, M. (2024). Stochastic RAG: End-to-end retrieval-augmented generation through expected utility maximization. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. (SIGIR '24). Association for Computing Machinery, New York, NY, USA, 2641–2646. <https://doi.org/10.1145/3626772.3657923>

Zylon by PrivateGPT (2023). PrivateGPT [Computer software]. <https://github.com/zylon-ai/private-gpt>

Παράρτημα: Κώδικας υλοποίησης

```
Server Settings:

server:

  env_name: ${APP_ENV:ollama}

llm:

  mode: ollama

  max_new_tokens: 512

  context_window: 3900

  temperature: 0.1 #The temperature of the model. Increasing the temperature will make the
model answer more creatively. A value of 0.1 would be more factual. (Default: 0.1)

embedding:

  mode: ollama

ollama:

  llm_model: llama3.1

  embedding_model: nomic-embed-text

  api_base: http://localhost:11434

  embedding_api_base: http://localhost:11434 # change if your embedding model runs on
another ollama
```

```
keep_alive: 5m

tfs_z: 1.0      # Tail free sampling is used to reduce the impact of less probable tokens from
the output. A higher value (e.g., 2.0) will reduce the impact more, while a value of 1.0 disables this
setting.

top_k: 40      # Reduces the probability of generating nonsense. A higher value (e.g. 100) will
give more diverse answers, while a lower value (e.g. 10) will be more conservative. (Default: 40)

top_p: 0.9     # Works together with top-k. A higher value (e.g., 0.95) will lead to more diverse
text, while a lower value (e.g., 0.5) will generate more focused and conservative text. (Default: 0.9)

repeat_last_n: 64  # Sets how far back for the model to look back to prevent repetition. (Default:
64, 0 = disabled, -1 = num_ctx)

repeat_penalty: 1.2  # Sets how strongly to penalize repetitions. A higher value (e.g., 1.5) will
penalize repetitions more strongly, while a lower value (e.g., 0.9) will be more lenient. (Default:
1.1)

request_timeout: 120.0 # Time elapsed until ollama times out the request. Default is 120s. Format
is float.

vectorstore:

database: qdrant

qdrant:

path: local_data/private_gpt/qdrant
```

Server

Run:

```
# start a fastapi server with uvicorn
```

```
import uvicorn

from private_gpt.main import app

from private_gpt.settings.settings import settings

# Set log_config=None to do not use the uvicorn logging configuration, and
# use ours instead. For reference, see below:
# https://github.com/tiangolo/fastapi/discussions/7457#discussioncomment-5141108

uvicorn.run(app, host="0.0.0.0", port=settings().server.port, log_config=None)
```

```
"""FastAPI app creation, logger configuration and main API routes."""

import logging

from fastapi import Depends, FastAPI, Request

from fastapi.middleware.cors import CORSMiddleware

from injector import Injector

from llama_index.core.callbacks import CallbackManager

from llama_index.core.callbacks.global_handlers import create_global_handler
```

```
from llama_index.core.settings import Settings as LlamaIndexSettings

from private_gpt.server.chat.chat_router import chat_router

from private_gpt.server.chunks.chunks_router import chunks_router

from private_gpt.server.completions.completions_router import completions_router

from private_gpt.server.embeddings.embeddings_router import embeddings_router

from private_gpt.server.health.health_router import health_router

from private_gpt.server.ingest.ingest_router import ingest_router

from private_gpt.server.recipes.summarize.summarize_router import summarize_router

from private_gpt.settings.settings import Settings

logger = logging.getLogger(__name__)

def create_app(root_injector: Injector) -> FastAPI:

    # Start the API

    async def bind_injector_to_request(request: Request) -> None:

        request.state.injector = root_injector

    app = FastAPI(dependencies=[Depends(bind_injector_to_request)])
```

```
app.include_router(completions_router)

app.include_router(chat_router)

app.include_router(chunks_router)

app.include_router(ingest_router)

app.include_router(summarize_router)

app.include_router(embeddings_router)

app.include_router(health_router)

# Add LlamaIndex simple observability

global_handler = create_global_handler("simple")

if global_handler:

    LlamaIndexSettings.callback_manager = CallbackManager([global_handler])

settings = root_injector.get(Settings)

if settings.server.cors.enabled:

    logger.debug("Setting up CORS middleware")

    app.add_middleware(

        CORSMiddleware,

        allow_credentials=settings.server.cors.allow_credentials,

        allow_origins=settings.server.cors.allow_origins,

        allow_origin_regex=settings.server.cors.allow_origin_regex,
```

```
    allow_methods=settings.server.cors.allow_methods,

    allow_headers=settings.server.cors.allow_headers,

)

if settings.ui.enabled:

    logger.debug("Importing the UI module")

    try:

        from private_gpt.ui.ui import PrivateGptUi

    except ImportError as e:

        raise ImportError(

            "UI dependencies not found, install with `poetry install --extras ui`"

        ) from e

    ui = root_injector.get(PrivateGptUi)

    ui.mount_in_app(app, settings.ui.path)

return app
```

Chatbot:

```
from dataclasses import dataclass
```

```
from typing import TYPE_CHECKING

from injector import inject, singleton

from llama_index.core.chat_engine import ContextChatEngine, SimpleChatEngine

from llama_index.core.chat_engine.types import (
    BaseChatEngine,
)

from llama_index.core.indices import VectorStoreIndex

from llama_index.core.indices.postprocessor import MetadataReplacementPostProcessor

from llama_index.core.llms import ChatMessage, MessageRole

from llama_index.core.postprocessor import (
    SentenceTransformerRerank,
    SimilarityPostprocessor,
)

from llama_index.core.storage import StorageContext

from llama_index.core.types import TokenGen

from pydantic import BaseModel

from private_gpt.components.embedding.embedding_component import
EmbeddingComponent

from private_gpt.components.llm.llm_component import LLMComponent
```

```
from private_gpt.components.node_store.node_store_component import NodeStoreComponent

from private_gpt.components.vector_store.vector_store_component import (
    VectorStoreComponent,
)

from private_gpt.open_ai.extensions.context_filter import ContextFilter

from private_gpt.server.chunks.chunks_service import Chunk

from private_gpt.settings.settings import Settings

if TYPE_CHECKING:
    from llama_index.core.postprocessor.types import BaseNodePostprocessor

class Completion(BaseModel):
    response: str

    sources: list[Chunk] | None = None

class CompletionGen(BaseModel):
    response: TokenGen

    sources: list[Chunk] | None = None

@dataclass

class ChatEngineInput:
```

```
system_message: ChatMessage | None = None

last_message: ChatMessage | None = None

chat_history: list[ChatMessage] | None = None

@classmethod

def from_messages(cls, messages: list[ChatMessage]) -> "ChatEngineInput":

    # Detect if there is a system message, extract the last message and chat history

    system_message = (

        messages[0]

        if len(messages) > 0 and messages[0].role == MessageRole.SYSTEM

        else None

    )

    last_message = (

        messages[-1]

        if len(messages) > 0 and messages[-1].role == MessageRole.USER

        else None

    )

    # Remove from messages list the system message and last message,

    # if they exist. The rest is the chat history.

    if system_message:

        messages.pop(0)
```

```
if last_message:

    messages.pop(-1)

chat_history = messages if len(messages) > 0 else None

return cls(

    system_message=system_message,

    last_message=last_message,

    chat_history=chat_history,

)

@singleton

class ChatService:

    settings: Settings

    @inject

    def __init__(

        self,

        settings: Settings,

        llm_component: LLMComponent,

        vector_store_component: VectorStoreComponent,

        embedding_component: EmbeddingComponent,
```

```
node_store_component: NodeStoreComponent,

) -> None:

    self.settings = settings

    self.llm_component = llm_component

    self.embedding_component = embedding_component

    self.vector_store_component = vector_store_component

    self.storage_context = StorageContext.from_defaults(

        vector_store=vector_store_component.vector_store,

        docstore=node_store_component.doc_store,

        index_store=node_store_component.index_store,

    )

    self.index = VectorStoreIndex.from_vector_store(

        vector_store_component.vector_store,

        storage_context=self.storage_context,

        llm=llm_component.llm,

        embed_model=embedding_component.embedding_model,

        show_progress=True,

    )

def _chat_engine(

    self,
```

```
system_prompt: str | None = None,

use_context: bool = False,

context_filter: ContextFilter | None = None,

) -> BaseChatEngine:

    settings = self.settings

    if use_context:

        vector_index_retriever = self.vector_store_component.get_retriever(

            index=self.index,

            context_filter=context_filter,

            similarity_top_k=self.settings.rag.similarity_top_k,

        )

        node_postprocessors: list[BaseNodePostprocessor] = [

            MetadataReplacementPostProcessor(target_metadata_key="window"),

        ]

        if settings.rag.similarity_value:

            node_postprocessors.append(

                SimilarityPostprocessor(

                    similarity_cutoff=settings.rag.similarity_value

                )

            )

    )
```

```
if settings.rag.rerank.enabled:

    rerank_postprocessor = SentenceTransformerRerank(

        model=settings.rag.rerank.model, top_n=settings.rag.rerank.top_n

    )

    node_postprocessors.append(rerank_postprocessor)

return ContextChatEngine.from_defaults(

    system_prompt=system_prompt,

    retriever=vector_index_retriever,

    llm=self.llm_component.llm, # Takes no effect at the moment

    node_postprocessors=node_postprocessors,

)

else:

    return SimpleChatEngine.from_defaults(

        system_prompt=system_prompt,

        llm=self.llm_component.llm,

    )

def stream_chat(

    self,

    messages: list[ChatMessage],
```

```
use_context: bool = False,

context_filter: ContextFilter | None = None,

) -> CompletionGen:

chat_engine_input = ChatEngineInput.from_messages(messages)

last_message = (

    chat_engine_input.last_message.content

    if chat_engine_input.last_message

    else None

)

system_prompt = (

    chat_engine_input.system_message.content

    if chat_engine_input.system_message

    else None

)

chat_history = (

    chat_engine_input.chat_history if chat_engine_input.chat_history else None

)

chat_engine = self._chat_engine(

    system_prompt=system_prompt,

    use_context=use_context,
```

```
        context_filter=context_filter,
    )

    streaming_response = chat_engine.stream_chat(

        message=last_message if last_message is not None else "",

        chat_history=chat_history,

    )

    sources = [Chunk.from_node(node) for node in streaming_response.source_nodes]

    completion_gen = CompletionGen(

        response=streaming_response.response_gen, sources=sources

    )

    return completion_gen

def chat(

    self,

    messages: list[ChatMessage],

    use_context: bool = False,

    context_filter: ContextFilter | None = None,

) -> Completion:

    chat_engine_input = ChatEngineInput.from_messages(messages)

    last_message = (

        chat_engine_input.last_message.content
```

```
if chat_engine_input.last_message

else None

)

system_prompt = (

    chat_engine_input.system_message.content

    if chat_engine_input.system_message

    else None

)

chat_history = (

    chat_engine_input.chat_history if chat_engine_input.chat_history else None

)

chat_engine = self._chat_engine(

    system_prompt=system_prompt,

    use_context=use_context,

    context_filter=context_filter,

)

wrapped_response = chat_engine.chat(

    message=last_message if last_message is not None else "",

    chat_history=chat_history,

)
```

```
sources = [Chunk.from_node(node) for node in wrapped_response.source_nodes]

completion = Completion(response=wrapped_response.response, sources=sources)

return completion
```

```
from fastapi import APIRouter, Depends, Request

from llama_index.core.llms import ChatMessage, MessageRole

from pydantic import BaseModel

from starlette.responses import StreamingResponse

from private_gpt.open_ai.extensions.context_filter import ContextFilter

from private_gpt.open_ai.openai_models import (

    OpenAICompletion,

    OpenAIMessage,

    to_openai_response,

    to_openai_sse_stream,

)

from private_gpt.server.chat.chat_service import ChatService

from private_gpt.server.utils.auth import authenticated

chat_router = APIRouter(prefix="/v1", dependencies=[Depends(authenticated)])
```

```
class ChatBody(BaseModel):

    messages: list[OpenAIMessage]

    use_context: bool = False

    context_filter: ContextFilter | None = None

    include_sources: bool = True

    stream: bool = False

    model_config = {

        "json_schema_extra": {

            "examples": [

                {

                    "messages": [

                        {

                            "role": "system",

                            "content": "You are a rapper. Always answer with a rap.",

                        },

                        {

                            "role": "user",

                            "content": "How do you fry an egg?",

                        },

                    ],

                },

            ],

        },

    }
```

```
    ],  
  
    "stream": False,  
  
    "use_context": True,  
  
    "include_sources": True,  
  
    "context_filter": {  
  
        "docs_ids": ["c202d5e6-7b69-4869-81cc-dd574ee8ee11"]  
  
    },  
  
    }  
  
    ]  
  
    }  
  
    }
```

```
@chat_router.post(  
  
    "/chat/completions",  
  
    response_model=None,  
  
    responses={200: {"model": OpenAICompletion}},  
  
    tags=["Contextual Completions"],  
  
    openapi_extra={  
  
        "x-fern-streaming": {  
  
            "stream-condition": "stream",  
  
            "response": {"$ref": "#/components/schemas/OpenAICompletion"},  
  
        },  
  
    },  
  
)
```

```
"response-stream": {"$ref": "#/components/schemas/OpenAICompletion"},  
  
}  
  
},  
  
)  
  
def chat_completion(  
  
    request: Request, body: ChatBody  
  
) -> OpenAICompletion | StreamingResponse:  
  
    """Given a list of messages comprising a conversation, return a response.  
  
    Optionally include an initial `role: system` message to influence the way  
  
    the LLM answers.  
  
    If `use_context` is set to `true`, the model will use context coming  
  
    from the ingested documents to create the response. The documents being used can  
  
    be filtered using the `context_filter` and passing the document IDs to be used.  
  
    Ingested documents IDs can be found using `/ingest/list` endpoint. If you want  
  
    all ingested documents to be used, remove `context_filter` altogether.  
  
    When using `include_sources: true`, the API will return the source Chunks used  
  
    to create the response, which come from the context provided.
```

When using ``stream': true``, the API will return data chunks following [OpenAI's streaming model](https://platform.openai.com/docs/api-reference/chat/streaming):

```
...
```

```
{"id":"12345","object":"completion.chunk","created":1694268190,
"model":"private-gpt","choices":[{"index":0,"delta":{"content":"Hello"},
"finish_reason":null}]}
```

```
...
```

```
"""
```

```
service = request.state.injector.get(ChatService)

all_messages = [
    ChatMessage(content=m.content, role=MessageRole(m.role)) for m in body.messages
]

if body.stream:
    completion_gen = service.stream_chat(
        messages=all_messages,
        use_context=body.use_context,
        context_filter=body.context_filter,
    )

    return StreamingResponse(
        to_openai_sse_stream(
            completion_gen.response,
```

```
        completion_gen.sources if body.include_sources else None,

    ),

    media_type="text/event-stream",

)

else:

    completion = service.chat(

        messages=all_messages,

        use_context=body.use_context,

        context_filter=body.context_filter,

    )

    return to_openai_response(

        completion.response, completion.sources if body.include_sources else None

    )
```

File

Ingest:

```
import logging

import tempfile

from pathlib import Path

from typing import TYPE_CHECKING, AnyStr, BinaryIO
```

```
from injector import inject, singleton

from llama_index.core.node_parser import SentenceWindowNodeParser

from llama_index.core.storage import StorageContext

from private_gpt.components.embedding.embedding_component import
EmbeddingComponent

from private_gpt.components.ingest.ingest_component import get_ingestion_component

from private_gpt.components.llm.llm_component import LLMComponent

from private_gpt.components.node_store.node_store_component import NodeStoreComponent

from private_gpt.components.vector_store.vector_store_component import (
    VectorStoreComponent,
)

from private_gpt.server.ingest.model import IngestedDoc

from private_gpt.settings.settings import settings

if TYPE_CHECKING:
    from llama_index.core.storage.docstore.types import RefDocInfo

logger = logging.getLogger(__name__)

@singleton
```

```
class IngestService:

    @inject

    def __init__(

        self,

        llm_component: LLMComponent,

        vector_store_component: VectorStoreComponent,

        embedding_component: EmbeddingComponent,

        node_store_component: NodeStoreComponent,

    ) -> None:

        self.llm_service = llm_component

        self.storage_context = StorageContext.from_defaults(

            vector_store=vector_store_component.vector_store,

            docstore=node_store_component.doc_store,

            index_store=node_store_component.index_store,

        )

        node_parser = SentenceWindowNodeParser.from_defaults()

        self.ingest_component = get_ingestion_component(

            self.storage_context,

            embed_model=embedding_component.embedding_model,

            transformations=[node_parser, embedding_component.embedding_model],
```

```
settings=settings(),

)

def _ingest_data(self, file_name: str, file_data: AnyStr) -> list[IngestedDoc]:

    logger.debug("Got file data of size=%s to ingest", len(file_data))

    # llama-index mainly supports reading from files, so

    # we have to create a tmp file to read for it to work

    # delete=False to avoid a Windows 11 permission error.

    with tempfile.NamedTemporaryFile(delete=False) as tmp:

        try:

            path_to_tmp = Path(tmp.name)

            if isinstance(file_data, bytes):

                path_to_tmp.write_bytes(file_data)

            else:

                path_to_tmp.write_text(str(file_data))

            return self.ingest_file(file_name, path_to_tmp)

        finally:

            tmp.close()

            path_to_tmp.unlink()

def ingest_file(self, file_name: str, file_data: Path) -> list[IngestedDoc]:
```

```
logger.info("Ingesting file_name=%s", file_name)

documents = self.ingest_component.ingest(file_name, file_data)

logger.info("Finished ingestion file_name=%s", file_name)

return [IngestedDoc.from_document(document) for document in documents]

def ingest_text(self, file_name: str, text: str) -> list[IngestedDoc]:

    logger.debug("Ingesting text data with file_name=%s", file_name)

    return self._ingest_data(file_name, text)

def ingest_bin_data(

    self, file_name: str, raw_file_data: BinaryIO

) -> list[IngestedDoc]:

    logger.debug("Ingesting binary data with file_name=%s", file_name)

    file_data = raw_file_data.read()

    return self._ingest_data(file_name, file_data)

def bulk_ingest(self, files: list[tuple[str, Path]]) -> list[IngestedDoc]:

    logger.info("Ingesting file_names=%s", [f[0] for f in files])

    documents = self.ingest_component.bulk_ingest(files)

    logger.info("Finished ingestion file_name=%s", [f[0] for f in files])

    return [IngestedDoc.from_document(document) for document in documents]
```

```
def list_ingested(self) -> list[IngestedDoc]:

    ingested_docs: list[IngestedDoc] = []

    try:

        docstore = self.storage_context.docstore

        ref_docs: dict[str, RefDocInfo] | None = docstore.get_all_ref_doc_info()

        if not ref_docs:

            return ingested_docs

        for doc_id, ref_doc_info in ref_docs.items():

            doc_metadata = None

            if ref_doc_info is not None and ref_doc_info.metadata is not None:

                doc_metadata = IngestedDoc.curate_metadata(ref_doc_info.metadata)

            ingested_docs.append(

                IngestedDoc(

                    object="ingest.document",

                    doc_id=doc_id,

                    doc_metadata=doc_metadata,

                )

            )

    )
```

```
except ValueError:

    logger.warning("Got an exception when getting list of docs", exc_info=True)

    pass

logger.debug("Found count=%s ingested documents", len(ingested_docs))

return ingested_docs

def delete(self, doc_id: str) -> None:

    """Delete an ingested document.

    :raises ValueError: if the document does not exist

    """

    logger.info(

        "Deleting the ingested document=%s in the doc and index store", doc_id

    )

    self.ingest_component.delete(doc_id)
```

Embeddings:

```
from typing import Literal

from injector import inject, singleton
```

```
from pydantic import BaseModel, Field

from private_gpt.components.embedding.embedding_component import
EmbeddingComponent

class Embedding(BaseModel):

    index: int

    object: Literal["embedding"]

    embedding: list[float] = Field(examples=[[0.0023064255, -0.009327292]])

@singleton

class EmbeddingsService:

    @inject

    def __init__(self, embedding_component: EmbeddingComponent) -> None:

        self.embedding_model = embedding_component.embedding_model

    def texts_embeddings(self, texts: list[str]) -> list[Embedding]:

        texts_embeddings = self.embedding_model.get_text_embedding_batch(texts)

        return [

            Embedding(

                index=texts_embeddings.index(embedding),
```

```
        object='embedding',

        embedding=embedding,

    )

    for embedding in texts_embeddings

]
```

Setup:

```
#!/usr/bin/env python3

import os

import argparse

from huggingface_hub import hf_hub_download, snapshot_download

from transformers import AutoTokenizer

from private_gpt.paths import models_path, models_cache_path

from private_gpt.settings.settings import settings

resume_download = True

if __name__ == '__main__':

    parser = argparse.ArgumentParser(prog='Setup: Download models from Hugging Face')
```

```
parser.add_argument('--resume', default=True, action=argparse.BooleanOptionalAction,
help='Enable/Disable resume_download options to restart the download progress interrupted')

args = parser.parse_args()

resume_download = args.resume

os.makedirs(models_path, exist_ok=True)

# Download Embedding model

embedding_path = models_path / "embedding"

print(f"Downloading embedding {settings().huggingface.embedding_hf_model_name}")

snapshot_download(

    repo_id=settings().huggingface.embedding_hf_model_name,

    cache_dir=models_cache_path,

    local_dir=embedding_path,

    token=settings().huggingface.access_token,

)

print("Embedding model downloaded!")

# Download LLM and create a symlink to the model file

print(f"Downloading LLM {settings().llamacpp.llm_hf_model_file}")

hf_hub_download(
```

```
repo_id=settings().llamacpp.llm_hf_repo_id,

filename=settings().llamacpp.llm_hf_model_file,

cache_dir=models_cache_path,

local_dir=models_path,

resume_download=resume_download,

token=settings().huggingface.access_token,

)

print("LLM model downloaded!")

# Download Tokenizer

if settings().llm.tokenizer:

    print(f"Downloading tokenizer {settings().llm.tokenizer}")

    AutoTokenizer.from_pretrained(

        pretrained_model_name_or_path=settings().llm.tokenizer,

        cache_dir=models_cache_path,

        token=settings().huggingface.access_token,

    )

    print("Tokenizer downloaded!")

print("Setup done")
```

