



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΔΙΟΙΚΗΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΑΡΤΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εξατομίκευση Chatbot σε πραγματικά δεδομένα

Στοιχεία φοιτητή:

Μουζάκι Ιωάννης

15ο εξάμηνο – ΑΜ: 1694 – email: th11816144@uoi.gr

Επιβλέπον καθηγητής: Καρβέλης Πέτρος,

ΔΕΠ ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Αθήνα, Οκτώβριος, 2025

Personalization of a Chatbot using Real Data

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, 14/11/25

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Πέτρος Καρβέλης,
2. Μέλος επιτροπής
Αλέξανδρος Τζάλλας,
3. Μέλος επιτροπής
Νικόλαος Γιαννακέας

© Μουζάκι, Ιωάννης, 2025.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Μουζάκι, Ιωάννης

Υπογραφή



Περίληψη

Η παρούσα εργασία παρουσιάζει την υλοποίηση ενός εκπαιδευτικού chat bot που απαντά τεκμηριωμένα σε ερωτήματα σχετιζόμενα με το Τμήμα, αξιοποιώντας προσέγγιση Retrieval Augmented Generation. Αρχικά, πραγματοποιείτε web scraping του ιστότοπου σε όλα τα τμήματα (ανακοινώσεις, προσωπικό, γραμματεία κ.λπ.) και λήψη – επεξεργασία των συνημμένων pdf. Τα έγγραφα μετατρέπονται σε καθαρό κείμενο χρησιμοποιώντας διάφορες τεχνικές. Αφού γίνουν οι απαραίτητες τροποποιήσεις, τα δεδομένα αποθηκεύονται σε SQLite βάση όπου κατασκευάζονται corpus σε chunks με ευρετήριο FTS5/BM25 για λεξική αναζήτηση. Παράλληλα, τα ίδια αποσπάσματα υπολογίζονται embeddings και αποθηκεύονται σε ChromaDB, επιτρέποντας σημασιολογική αναζήτηση. Η ανάκτηση είναι υβριδική, τα αποτελέσματα από τη λεξική και σημασιολογική αναζήτηση συγχωνεύονται με Reciprocal Rank Fusion, ενώ χρησιμοποιούνται εργαλεία για την διατήρηση συμφραζομένων και πηγών. Έπειτα, τα καλύτερα αποτελέσματα τροφοδοτούν ένα προσεκτικά σχεδιασμένο prompt προς το LLM, το οποίο παράγει σύντομες και ελέγξιμες απαντήσεις. Η διεπαφή υλοποιείτε σε Gradio, προσφέροντας ρυθμίσεις παραμέτρων και εξαγωγή συνομιλίας.

Λέξεις κλειδιά: RAG, ChromaDB, Embeddings, LLM, BM25

Abstract

This thesis presents the implementation of an educational chatbot that provides evidence-based answers to department-related queries using Retrieval-Augmented Generation (RAG) approach. Firstly, the department website is scraped across all sections (announcements, staff, secretariat, etc.) and attached PDFs are downloaded and processed.

Documents are converted to clean text using various techniques. After the necessary preprocessing, the data are stored in an SQLite database, where the corpus is split into chunks and indexed with FTS5/BM25 for lexical search. In parallel, the same chunks are embedded and stored in ChromaDB to enable semantic search. Retrieval is hybrid: results from lexical and semantic search are fused via Reciprocal Rank Fusion, with additional steps to preserve context and source traceability. The best results then feed a carefully designed prompt to the LLM, which produces concise, certifiable answers. The user interface is implemented in Gradio, offering parameter controls and chat exports.

Keywords: RAG, ChromaDB, Embeddings, LLM, BM25

Περιεχόμενα:

Κατάλογος εικόνων:.....	ix
Κατάλογος Συντομογραφιών	x
Γλωσσάριο αγγλικών όρων	xi
Κεφάλαιο 1 – Εισαγωγή:.....	1
1.1 Γενικό Τεχνολογικό Πλαίσιο.....	1
1.2 Το πρόβλημα και η ανάγκη παρέμβασης.....	1
1.3 Μεθοδολογική Προσέγγιση	2
1.4 Σκοπός και Στόχοι της Εργασίας.....	2
Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο	3
2.1 Chatbots – Ορισμός και Ιστορική Εξέλιξη.....	3
2.2 Επεξεργασία Φυσικής Γλώσσας και LLMs	4
2.3 Chatbots στην Ανώτατη Εκπαίδευση.....	5
2.4 Πλεονεκτήματα και Προκλήσεις.....	6
Κεφάλαιο 3: Ανάλυση και Σχεδίαση Συστήματος.....	6
3.1: Συνοπτική περιγραφή.....	6
3.2: Αρχιτεκτονική συστήματος και βάση δεδομένων.....	7
3.3: Λειτουργικές και μη Λειτουργικές απαιτήσεις	8
Κεφάλαιο 4: Υλοποίηση	9
4.1: Web scraping: Πώς συλλέγω τα δεδομένα (modules & helpers).....	9
4.2: Επεξεργασία pdf στο Google Colab.....	12
4.3: Δημιουργία Βάσης Δεδομένων – Σχήμα, εισαγωγή, chunking και FTS5.....	13
4.4: Ανάκτηση και παραγωγή – Hybrid search, prompt, LLM	15
4.5: Εκτέλεση & User Interface – Gradio app, ρυθμίσεις, ροή αιτήματος.....	17
4.6: Παραδείγματα και Εκτέλεση του έργου.....	19
Κεφάλαιο 5 – Συμπεράσματα και προτάσεις βελτίωσης	33
5.1: Περιορισμοί και βελτιώσεις	33
5.2: Μελλοντικές ιδέες.....	34

5.3 Συμπεράσματα.....	34
Βιβλιογραφία.....	34

Κατάλογος εικόνων:

Εικόνα 1. ELIZA, το πρώτο ChatBot.....	3
Εικόνα 2. Λειτουργία ενός RAG συστήματος.....	7
Εικόνα 3. Σχεδίαση της Βάσης σε StarUML.	15
Εικόνα 4. Πόροι Colab (L4 GPU).....	20
Εικόνα 5. Φόρτωση εξαρτήσεων.	20
Εικόνα 6. Εξαγωγή των δεδομένων από την ιστοσελίδα του τμήματος.	21
Εικόνα 7. Λήψη των αρχείων.....	22
Εικόνα 8. Φόρτωση εξαρτήσεων για τα γλωσσικά μοντέλα.....	23
Εικόνα 9. Επεξεργασία αρχείων pdf.	24
Εικόνα 10. Δημιουργία schema της βάσης.	25
Εικόνα 11. Embeddings σε ChromaDB.	26
Εικόνα 12. Υβριδική ανάκτηση.	27
Εικόνα 13. Εκκίνηση του chatbot.	28
Εικόνα 14. Αποθήκευση διαλόγου και εξαγωγή.....	28
Εικόνα 15. Ερώτηση 1.	29
Εικόνα 16. Ερώτηση 2.	30
Εικόνα 17. Ερώτηση 3.	31
Εικόνα 18. Ερώτηση 4.	32
Εικόνα 19. Ερώτηση 5.	33

Κατάλογος Συντομογραφιών

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
BM25	Best Match 25
FTS5	Full Text Search v5
SQL	Structured Query Language
UI	User Interface
URL	Uniform Resource Locator
PDF	Portable Document Format
CSV	Comma-Separated Values
YAML	YAML Ain't Markup Language
DOM	Document Object Model
OCR	Optical Character Recognition
GPU	Graphics Processing Unit
RRF	Reciprocal Rank Fusion
POS	Part of Speech
GDRP	General Data Protection Regulation
API	Application Programming Interface
REST	Representational State Transfer
SHA-256	Secured Hash Algorithm
CUDA	Compute Unified Device Architecture (Nvidia)
GPT	Generative Pre-Trained Transformer
LORA/QLORA	Low Rank Adaptation

Γλωσσάριο αγγλικών όρων

Token	Μονάδα κειμένου
Tokenization	Διαδικασία διάσπασης κειμένου σε tokens
Lemmatization	Μετατρέπει την κάθε λέξη στη βασική της μορφή
Stemming	Αποκοπή καταλήξεων
POS tagging	Αναγνώριση γραμματικών κατηγοριών.
Sentiment analysis	Ανάλυση συναισθήματος
Embeddings	Διανυσματική αναπαράσταση κειμένου για σημασιολογική αναζήτηση
Vector Search	Αναζήτηση με διάνυσμα
Vector Store	Αποθήκη διανυσμάτων
Chunking	Τεμαχισμός κειμένου σε αποσπάσματα
Overlap	Επικάλυψη μεταξύ αποσπασμάτων
Top-k	Επιλογή των k κορυφαίων αποτελεσμάτων
Idempotency	Κάνει το σύστημα ασφαλές σε επανεκτελέσεις
Prompt	Οδηγίες προς LLM
Token Budgeting	Διαχείριση χώρου tokens στο prompt
Neighbor expansion	Επέκταση σε γειτονικά Chunks
Diversity	Ποικιλία πηγών στο context
Persistent Collection	Μόνιμη συλλογή διανυσμάτων
Manifest	Συγκεντρωτικός κατάλογος αρχείων
Quality Gates	Έλεγχοι ποιότητας
Quality Score	Βαθμολογία
Upsert	Εισαγωγή ή ενημέρωση εγγραφής στη βάση
Headless Chrome	Εκτέλεση browser χωρίς γραφικό περιβάλλον
Lazy-Loading / Expanders	Τεχνικές φόρτωσης και ανάπτυξης δυναμικού περιεχομένου
Sentence Transformer	Βιβλιοθήκη παραγωγής embeddings
Post-Processing	Βήματα καθαρισμού ή μορφοποίησης

Κεφάλαιο 1 – Εισαγωγή:

1.1 Γενικό Τεχνολογικό Πλαίσιο

Κάθε τομέας της κοινωνίας, την τελευταία δεκαετία, έχει επηρεαστεί ραγδαία από την εξέλιξη των τεχνολογιών πληροφορικής και τηλεπικοινωνιών, με κύριο στόχο την εκπαίδευση. Ο τρόπος ο οποίος οι φοιτητές αλληλοεπιδρούν με το πανεπιστημιακό περιβάλλον έχει αλλάξει εξαιτίας της συνεχούς ψηφιοποίησης υπηρεσιών και διαδικασιών το οποίο έχει δημιουργήσει νέες απαιτήσεις για πιο γρήγορη και αποτελεσματική πρόσβαση στην πληροφορία. Η αξιοποίηση σύγχρονων τεχνολογιών κρίνεται απαραίτητη λόγω του αυξημένου όγκου δεδομένων που διαχειρίζονται οι φοιτητές σε συνδυασμό με τις ανάγκες επικοινωνίας με τις υπηρεσίες και τους διδάσκοντες. Στο πλαίσιο αυτό, η Τεχνητή Νοημοσύνη (AI) και οι κλάδοι της, Μηχανική Μάθηση (ML) και η Βαθιά Μάθηση (DL), αναδεικνύονται ως ισχυρά εργαλεία που προσφέρουν καινοτόμες λύσεις στις ανάγκες της εκπαίδευσης.

1.2 Το πρόβλημα και η ανάγκη παρέμβασης

Στα πανεπιστήμια παρατηρούνται σημαντικές δυσκολίες στην άμεση και αξιόπιστη ενημέρωση των φοιτητών για ακαδημαϊκά θέματα παρά την χρήση ποικίλων ψηφιακών πλατφόρμων όπως η ιστοσελίδα της σχολής. Οι διοικητικές υπηρεσίες καλούνται να διαχειριστούν μεγάλο αριθμό αιτημάτων που συχνά επαναλαμβάνονται, όπως ερωτήσεις για προθεσμία δηλώσεων μαθημάτων. Με την ίδια προοπτική, και οι διδάσκοντες δέχονται πληθώρα μηνύματα και email για ζητήματα τα οποία θα μπορούσαν να απαντηθούν άμεσα μέσω ενός αυτοματοποιημένου συστήματος. Λόγω του φόρτου αυτού, το αποτέλεσμα είναι καθυστερήσεις και επιβάρυνση των υπηρεσιών το οποίο καταλήγει σε δυσαρέσκεια τόσο από την πλευρά των φοιτητών όσο και από το προσωπικό. Έτσι, πολλοί φοιτητές στρέφονται σε άτυπα μέσα, όπως ομάδες στα κοινωνικά δίκτυα, όπου η εγκυρότητα της πληροφορίας συχνά δεν είναι έγκυρη (Winkler & Soellner, 2018). Στο πλαίσιο αυτό, η εργασία προσπαθεί να αναπτύξει ένα ευφρές σύστημα, chatbot ειδικά σχεδιασμένο για τις ανάγκες που αφορούν το περιβάλλον του πανεπιστημίου. Ένα τέτοιο σύστημα μπορεί να αποσυμφορήσει το έργο των διοικητικών υπηρεσιών όπως και τους φοιτητές αφού θα αποτελέσει ένα σημείο άμεσης και αξιόπιστης πληροφόρησης. Η ενσωμάτωση του chatbot σε πλατφόρμες που

χρησιμοποιούνται ήδη όπως η ιστοσελίδα της σχολής, μπορεί να συμβάλλει στην δημιουργία ενός πιο συνοπτικού και λειτουργικού συστήματος εκπαίδευσης.

1.3 Μεθοδολογική Προσέγγιση

Η ανάπτυξη του συστήματος προϋποθέτει την κατανόηση βασικών αρχών της Τεχνητής Νοημοσύνης. Έτσι, η ΑΙ ορίζεται ως το επιστημονικό πεδίο που επιδιώκει να κατασκευάσει μηχανές με δυνατότητες λήψης αποφάσεων και προσαρμογής στη μάθηση, σε βαθμό που προσομοιάζει την ανθρώπινη νοημοσύνη (Stuart J. Russel & Peter Norvig, 2022). Η Μηχανική Μάθηση, ως υπό επίπεδο της ΑΙ, δίνει την δυνατότητα στα συστήματα να βελτιώσουν την απόδοση τους μέσω της εμπειρίας, χρησιμοποιώντας δεδομένα για να ανιχνεύσουν πρότυπα και να κάνουν προβλέψεις (Moors et al., 2015). Τέλος, η Βαθιά Μάθηση, με την βοήθεια πολύ επίπεδων νευρωνικών δικτύων, καθιστά εφικτή την κατανόηση πολύπλοκων δεδομένων, όπως η φυσική γλώσσα και η επεξεργασία εικόνας. Για αυτό τον λόγο, θεωρείται καταλυτής για την πρόοδο συγχρόνων chatbots (Baldi & Sadowski, 2016).

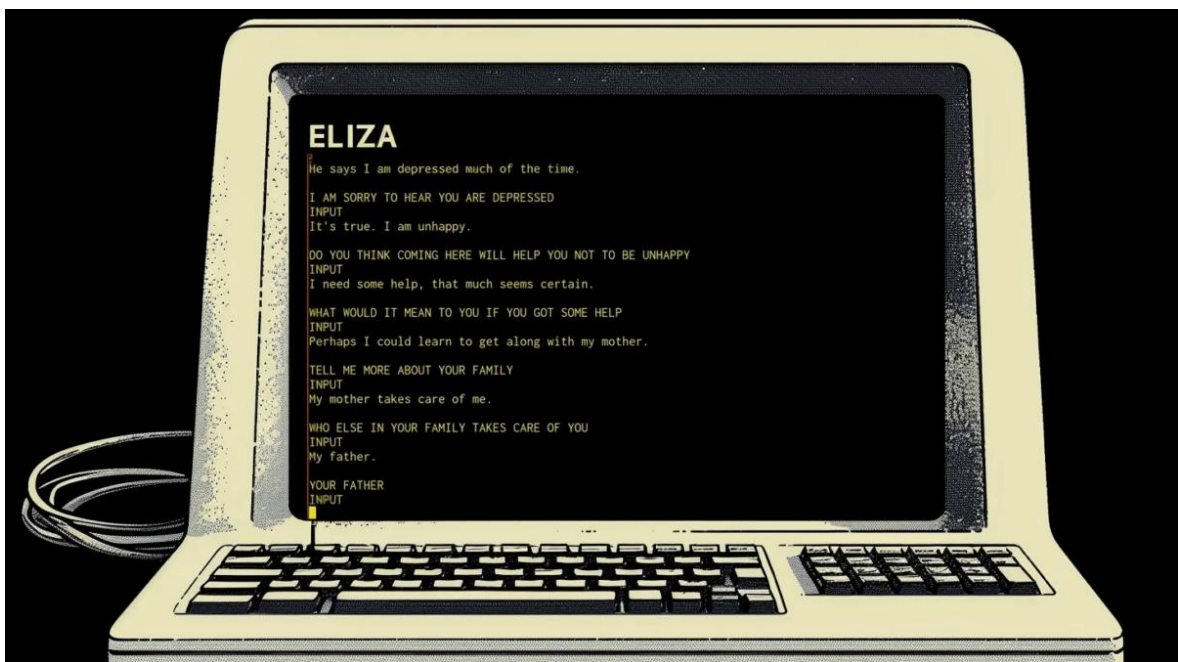
1.4 Σκοπός και Στόχοι της Εργασίας

Η ανάπτυξη της συγκεκριμένης διπλωματικής εργασίας δεν προορίζεται μόνο στην αποτύπωση σύγχρονων τάσεων και βιβλιογραφίας γύρω από τα chatbot, αλλά προχωρά στην υλοποίηση ενός συστήματος το οποίο θα αξιολογηθεί από φοιτητές και τις διοικητικές υπηρεσίες και θα προσφέρει χρήσιμα συμπεράσματα για τους περιορισμούς και της δυνατότητες της τεχνολογίας αυτή. Η εργασία δεν μένει σε θεωρητική διάσταση, αλλά προσπαθεί να αποδείξει στην πράξη ότι η Τεχνητή Νοημοσύνη μπορεί να βελτιώσει την καθημερινή φοιτητική εμπειρία, προσφέροντας ένα άμεσο, διαδραστικό και αποτελεσματικό μοντέλο επικοινωνίας.

Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο

2.1 Chatbots – Ορισμός και Ιστορική Εξέλιξη

Τα chatbots, η αλλιώς συστήματα συνομιλίας, τα οποία ορίζονται ως προγράμματα υπολογιστών που έχουν δημιουργηθεί για να αλληλεπιδρούν με τον χρήστη μέσα από τη φυσική γλώσσα η οποία μπορεί να είναι είτε γραπτή είτε σε προφορική μορφή. Ο βασικός στόχος τους είναι, να μιμηθούν την ανθρώπινή συμπεριφορά και να προσφέρουν χρήσιμες πληροφορίες, καθοδήγηση ή υποστήριξη σε διάφορους τομείς, από την εξυπηρέτηση πελατών μέχρι και την εκπαίδευση. Η ιδέα του συστήματος αυτού δεν είναι καινούρια αφού ήδη από την δεκαετία του 1960 παρουσιάστηκε το ELIZA, που δημιουργήθηκε από τον Joseph Weizenbaum στο MIT. Το ELIZA επιχειρούσε να προσομοιάσει ένα ψυχοθεραπευτή χρησιμοποιώντας κανόνες αντιστοίχισης λέξεων – κλειδίων, αποδεικνύοντας ότι ακόμα και απλοί αλγόριθμοι έχουν δυνατότητα ενός ‘έξυπνου’ διαλόγου (Joseph Weizenbaum, 1966).



Εικόνα 1. ELIZA, το πρώτο ChatBot.

Στη συνέχεια, ακολουθήσε το PARRY στην δεκαετία του 1970, το οποίο ο ρόλος του ήταν να αναπαραστήσει τη συμπεριφορά ενός ανθρώπου με παρανοϊκή διαταραχή. Το PARRY ήταν πιο πειστικό αφού χρησιμοποιούσε συνθέτες στρατηγικές διαλόγου κάτι το οποίο το ELIZA δεν έκανε. Με την πάροδο του χρόνου, τα chatbots συνέχισαν να εξελίσσονται, αλλά παρέμειναν περιορισμένα στις δυνατότητες τους διότι στηρίζονταν κυρίως σε κανόνες (rule

-based systems). Η μεγάλη ανάπτυξη ήρθε με την πρόοδο της Μηχανικής Μάθησης (Machine Learning- ML) και υστέρη της Βαθιάς Μάθησης (Deep Learning - DL). Τα πολλά επίπεδα νευρωνικών δικτύων και η δυνατότητα χειρισμού μεγάλου όγκου δεδομένων επέτρεψαν στην κατασκευή των chatbots, που βοήθησαν τα συστήματα αυτά όχι απλώς να ακολουθούν προκαθορισμένους κανόνες αλλά να μπορούν να ‘μαθαίνουν’ τα μοτίβα της γλώσσας. Στη δεκαετία του 2010, με την εμφάνιση των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) , όπως το GPT, τα μοντέλα αυτά απέκτησαν την ικανότητα κατανόησης και δημιουργίας της φυσικής γλώσσας (Vaswani et al., 2023). Στη σύγχρονη εποχή, τα chatbots αποτελούν ώριμη τεχνολογία με εφαρμογές που αγγίζουν ένα ευρύ φάσμα τομέων. Ειδικότερα στην εκπαίδευση, χρησιμοποιούνται για να διευκολύνουν τη πρόσβαση στις πληροφορίες, για να ενισχύσουν την επικοινωνία με τα πανεπιστήμια αλλά και για να παρέχουν εξατομικευμένη υποστήριξη μάθησης (Winkler & Soellner, 2018). Τα πειραματικά στάδια της δεκαετίας του 1960 έως και τα σημερινά LLMs, καθρεπτίζει την πορεία της Τεχνητής Νοημοσύνης και την αλληλεπίδραση της με τις ανάγκες της εκπαίδευσης και της κοινωνίας.

2.2 Επεξεργασία Φυσικής Γλώσσας και LLMs

Η επεξεργασία της Φυσικής Γλώσσας (Natural Language Processing- NLP) καθίσταται ως ένας από τους σημαντικότερους κλάδους της τεχνητής Νοημοσύνης, με κύριο στόχο την ανάπτυξη μεθόδων και αλγορίθμων που επιτρέπουν στους υπολογιστές να κατανοούν, να ερμηνεύουν και να παράγουν την ανθρώπινη γλωσσά. Η NLP, συνδυάζει γλωσσολογικές θεωρίες με στατιστικά και μαθηματικά δεδομένα. Έτσι, έχει βρει εφαρμογή σε μεγάλο πλήθος καθημερινών τεχνολογιών, από τα συστήματα αυτόματης μετάφρασης έως και τα chatbots τα οποία χρησιμοποιούνται στην εκπαίδευση (Jurafsky & Martin, n.d.). Ένας από τους πιο βασικούς τρόπους επεξεργασίας και ανάλυσης κειμένου είναι το tokenization , δηλαδή η διάσπαση μιας πρότασης σε μικρότερα τμήματα (tokens). Ανάλογα με την εφαρμογή, τα tokens μπορεί να είναι λέξεις, μικρότερες λέξεις αλλά και χαρακτήρες. Σήμερα, τα γλωσσικά μοντέλα χρησιμοποιούν συχνά μικρότερες λέξεις (sub tokenization), η οποία βοηθά στην κατανόηση πολλών και αγνώστων λέξεων. Για παράδειγμα, η λέξη ‘ανθυγιεινός’ μπορεί να χωριστεί σε μικρότερες λέξεις (‘ανθ’, ‘υγιειν’, ‘ος’), κάτι το οποίο επιτρέπει το μοντέλο να αναγνωρίσει το νόημα ακόμα και αν δεν έχει δει ποτέ αυτή την λέξη. Στη συνέχεια, έχουμε τη διαδικασία της lemmatization το οποίο συνδέει κάθε λέξη με τη βασική μορφή (λήμμα). Έτσι, οι διαφορετικές μορφές μιας λέξης ομαδοποιούνται, μειώνοντας την πολυπλοκότητα των δεδομένων. Σε γλώσσες όπως η ελληνική, που κατέχει

πλούσιο λεξιλόγιο είναι ιδιαίτερα σημαντικό διότι μια λέξη μπορεί να εμφανιστεί σε πολλές κλιτικές μορφές. Η lemmatization εξασφαλίζει την ορθή γλωσσική επεξεργασία και αυξάνει την αξιοπιστία ανάλυσης σε αντίθεση με το stemming που αφαιρεί καταλήξεις χωρίς την γλωσσική ακρίβεια (Manning et al., n.d.). Με την μέθοδο του part – of – speech tagging (POS tagging) επιτρέπεται η αναγνώριση της γραμματική κατηγορίας κάθε λέξης μέσα σε μια πρόταση. Για παράδειγμα, η λέξη ‘διαβάζει’ αναγνωρίζεται ως ρήμα, ενώ η λέξη ‘διάβασμα’ ως ουσιαστικό. Η πληροφορία αυτή, είναι ιδιαίτερα σημαντική καθώς βοηθά στη κατανόηση της συντακτικής δομής και την αυτόματη μετάφραση ή την εξαγωγή κείμενων σε εφαρμογές. Στα LLMs, η POS ανάλυση ενσωματώνεται έμμεσα με την εκπαίδευση ,κάτι που εξακολουθεί να αποτελεί κρίσιμο στάδιο σε πολλές εφαρμογές NLP. Επιπλέον, μια από της σημαντικότερες εφαρμογής της NLP είναι η sentiment analysis. Μέσα από τους αλγορίθμους , δίνεται η δυνατότητα να προσδιοριστεί αν ο τόνος του κειμένου είναι θετικός, αρνητικός ή ουδέτερος Για παράδειγμα, μια αξιολόγηση από έναν φοιτητή όπως, ‘Το μάθημα ήταν ευχάριστο’ αναγνωρίζεται ως θετικό, σε αντίθεση με την πρόταση ‘Η επικοινωνία με τον καθηγητή ήταν δύσκολη’ που ταξινομείται ως αρνητική. Με την ανάλυση του συναισθήματος δίνεται η δυνατότητα από τις υπηρεσίες να αντιλαμβάνονται τις αντιδράσεις των φοιτητών και να προσαρμόζουν ανάλογα την υποστήριξη που παρέχουν. Έτσι, τα πανεπιστημιακά chatbots, έχουν ιδιαίτερη σημασία και αξία. Στα πρώτα συστήματα NLP οι τεχνικές tokenization, lemmatization και POS tagging υλοποιούνται ξεχωριστά με συμβολικούς ή στατικούς αλγορίθμους. Τα σύγχρονα LLMs σε αντίθεση, ενσωματώνουν μεγάλο ευρέως αυτών των κανόνων στην εκπαίδευση, μαθαίνοντας έμμεσα τους κανόνες γλωσσών. Αυτό οδηγεί σε μια πιο ‘φυσική’ αλληλεπίδραση όπου τα chatbots μπορούν να ανταποκρίνονται με αυξημένο επίπεδο κατανόησης και προσαρμοστικότητας (Vaswani et al., 2023).

2.3 Chatbots στην Ανώτατη Εκπαίδευση

Η χρήση των chatbots στην ανώτατη εκπαίδευση μπορούν να συμβάλλουν στην βελτίωση της ακαδημαϊκής εμπειρίας για τους φοιτητές όσο και την διευκόλυνση των διοικητικών διαδικασιών. Τα συστήματα αυτά, μπορούν να προσφέρουν άμεση πληροφορία στους φοιτητές σχετικά με τα προγράμματα σπουδών, προθεσμίες και εγγραφές ,μειώνοντας τον φόρτο των διοικητικών υπηρεσιών μέσα από τις τεχνικές NLP και LLMs (Stuart J. Russel & Peter Norvig, 2022). Έχει εφαρμοστεί με επιτυχία και για ακαδημαϊκή υποστήριξη, όπως φανερώνεται από το παράδειγμα του Jill Watson στο Georgia Institute of Technology των ΗΠΑ, που λειτούργησε και ως εικονικός βοηθός σε διαδικτυακά μαθήματα (Watson et al.,

n.d.). Μέσω των sentiment analysis ,μπορεί να προσφερθεί και ψυχολογική υποστήριξη κάτι το οποίο δεν αντικαθιστά τους ειδικούς αλλά λειτουργεί ως επιπλέον παράγοντας (Vaidyam et al., 2019).

2.4 Πλεονεκτήματα και Προκλήσεις

Σημαντικά οφέλη προσφέρει η χρήση του chatbot στην ανώτατη εκπαίδευση. Διευκολύνει τους φοιτητές να έχουν πρόσβασή σε πληροφορίες, με άμεσες απαντήσεις για ακαδημαϊκά και διοικητικά ζητήματα με χρονική διαθεσιμότητα 24 ωρών (Okonkwo & Ade-Ibijola, 2021). Παράλληλα, τα chatbots μπορούν να χρησιμοποιηθούν και να λειτουργήσουν ως συμπληρωματικά εργαλεία μάθησης, όπως καθοδήγηση στη μελέτη, γεγονός που ενισχύει την εξατομίκευση της εκπαιδευτικής εμπειρίας (Winkler & Soellner, 2018). Ωστόσο, η ενσωμάτωση αυτών των τεχνολογιών συνοδεύεται με πολλαπλές προκλήσεις . Μια βασική πρόκληση είναι η ακρίβεια των απαντήσεων, καθώς τα γλωσσικά μοντέλα δεν είναι αλάνθαστα και μπορούν να εμφανίσουν παραπλανητικές απαντήσεις. Μια επιπλέον πρόκληση είναι η προστασία προσωπικών δεδομένων και στη συμμόρφωση με το GDPR, καθώς τα συστήματα αυτά ενδέχεται να επεξεργαστούν ευαίσθητα στοιχεία φοιτητών (Floridi et al., 2018). Συνεπώς, υπάρχουν προκλήσεις που σχετίζονται με την συμμόρφωση και αποδοχή των χρηστών, καθώς υπάρχουν φοιτητές και διδάσκοντες οι οποίοι εκφράζουν επιφυλάξεις και την δυσαρέσκεια τους σχετικά με την παιδαγωγική αξία των chatbos (Zawacki-Richter et al., 2019). Τέλος, ενώ τα chatbots αποτελούν καινοτόμο εργαλείο για πανεπιστήμια, η επιτυχής αξιοποίηση τους απαιτεί ισορροπία αναμεσά στα πλεονεκτήματα και τους κινδύνους που κατέχουν.

Κεφάλαιο 3: Ανάλυση και Σχεδίαση Συστήματος

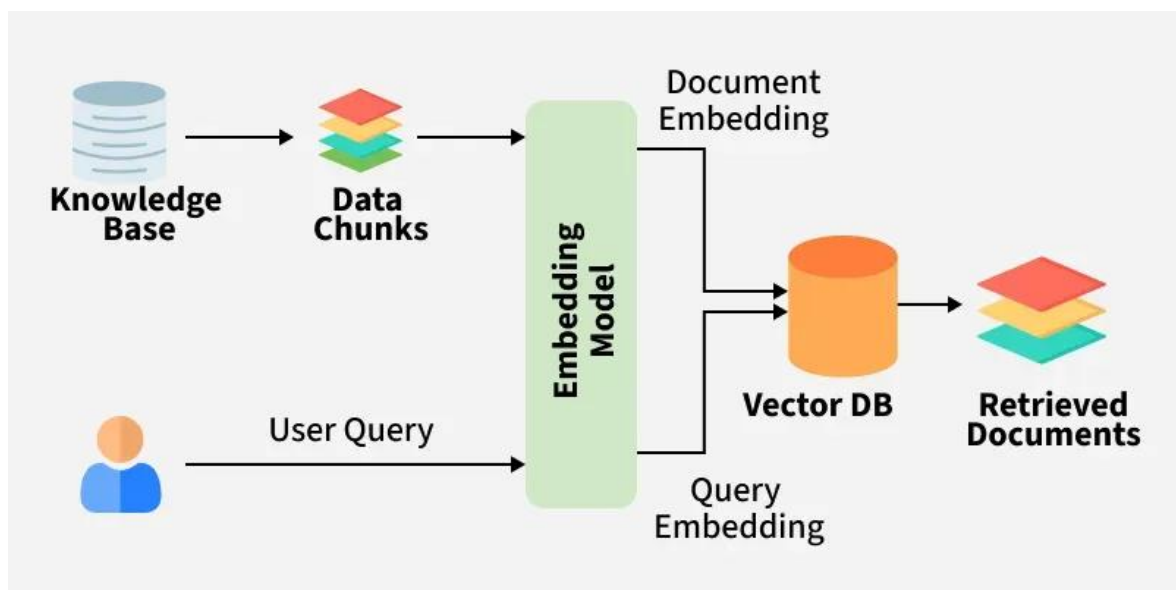
3.1: Συνοπτική περιγραφή

Ο στόχος του συστήματος είναι ένα εκπαιδευτικό chatbot για την ιστοσελίδα και τα μαθήματα του Τμήματος, που απαντά σε ερωτήσεις φοιτητών, διδασκόντων είτε και της γραμματείας αξιοποιώντας RAG (Retrieval Augmented Generation): αντλεί αποσπάσματα από το επίσημο υλικό (σελίδες ,ανακοινώσεις, έγγραφα PDF) και συνθέτει απαντήσεις με παραπομπές στα εκάστοτε URLs ή pdfs. Η αρχιτεκτονική είναι υβριδική, συνδυάζει λεκτική ανάκτηση πληροφορίας μέσω BM25 πάνω σε FTS5/SQLite και σημασιολογική με embeddings χρησιμοποιώντας το μοντέλο intfloat/multilingual-e5-large (huggingface) και αποθήκευση σε ChromaDB, ενώ η τελική απάντηση παράγεται από το ελληνικό γλωσσικό

μοντέλο LLM ilsp/Llama-Krikri-8b-instruct. Η προσωρινή διεπαφή προς τον χρήστη υλοποιείται με Gradio ChatInterface για απλό και καθαρό User Interface.

3.2: Αρχιτεκτονική συστήματος και βάση δεδομένων

Η αρχιτεκτονική του συστήματος αποτελείται από 5 διακριτά στάδια, (α) συλλογή του περιεχομένου με web scraping από τις ενότητες του ισότοπου (ανακοινώσεις, δράσεις, γραμματεία, μαθήματα, προσωπικό κ.λπ.) και λήψη των αρχείων pdf, εκ των οποίων καθαρίζονται από html σε text αρχεία και δημιουργούνται πίνακες CSV για την καλύτερη οργάνωση και ιχνηλασιμότητα των δεδομένων, (β) μετατροπή των αρχείων pdf σε text αρχεία, (γ) αποθήκευση και δεικτοδότηση στην βάση (SQLite) με δημιουργία corpus_chunks πάνω στα οποία χτίζεται FTS5 (που περιέχει BM25) ώστε να πετύχουμε την λεκτική αναζήτηση (SQLite), (δ) ανάκτηση των δεδομένων με υβριδική αναζήτηση (Microsoft_Hybrid_Search) που συνδυάζει BM25 και σημασιολογική αναζήτηση μέσω embeddings που αποθηκεύονται σε ChromaDB (ChromaDB) και (ε) σύνθεση της απάντησης από το LLM (GeeksForGeeks).



Εικόνα 2. Λειτουργία ενός RAG συστήματος.

Τα δεδομένα χωρίζονται σύμφωνα με τις ενότητες του τμήματος ανακοινώσεις, δράσεις, γραμματεία κ.λπ. Η κάθε ενότητα αποθηκεύεται αρχικά σε ξεχωριστό φάκελο, όπου υπάρχει ένας φάκελος με τα text αρχεία και ένα CSV αρχείο pages.csv, το αρχείο αυτό περιέχει χρήσιμες στήλες για την καθοδήγηση και οργάνωση των δεδομένων (ενδεικτικές στήλες: url, title, text_path, chars, doc_links, follow_links) . Τα pdf αρχεία καταγράφονται στο φάκελο pdfs, αφού γίνουν οι απαραίτητες μετατροπές δημιουργούμε έναν φάκελο

pdf_processed ώστε να διατηρείται ξεκάθαρη συσχέτιση μεταξύ ενοτήτων και εγγραφών. Παράλληλα, δημιουργούμε ένα επιπλέον CSV αρχείο που καθοδηγεί όλες τις ενότητες με τα καθαρισμένα αρχεία στο file_manifest.csv. Στη βάση δεδομένων κάθε ενότητα αντιστοιχεί σε πίνακα *_pages (π.χ. actions_pages για τις δραστηριότητες του τμήματος), τα pdfs αποθηκεύονται επίσης με τρόπο ώστε να διατηρείτε η συσχέτιση μεταξύ εγγράφων. Για το RAG δημιουργούνται corpus_docs ένα ανά σελίδα ή pdf κείμενο και corpus_chunks όπου τεμαχίζονται ανά περίπου 1000 χαρακτήρες και overlap περίπου 180 χαρακτήρες για συνοχή (chunking Microsoft, n.d.). Πάνω στα corpus_chunk χτίζεται FTS5 ευρετήριο (fts_corpus) που παρέχει BM25 για την λεκτική αναζήτηση. Παράλληλα τα ίδια corpus_chunk κωδικοποιούνται σε embeddings με intfloat/multilingual-e5-large και αποθηκεύονται σε ChromaDB (persistent collection) (ChromaDB, persistent collection). Η τελική κατάταξη προκύπτει με Reciprocal Rank Fusion (RRF) ώστε να συνδυάζονται ισορροπημένα οι λεξικές και σημασιολογικές ενδείξεις (RRF Microsoft). Τα top-K αποσπάσματα περνούν ως context στο LLM, το οποίο παράγει μια σύντομη και τεκμηριωμένη απάντηση με παραπομπές από τις εκάστοτε πηγές.

3.3: Λειτουργικές και μη Λειτουργικές απαιτήσεις

Λειτουργικές απαιτήσεις:

- Συνομιλία σε φυσική γλώσσα: ερωτήσεις/απαντήσεις στα Ελληνικά και Αγγλικά όπου χρειάζεται με διατήρηση σύντομου ιστορικού με την δυνατότητα αποθήκευσης της συνομιλίας.
- RAG ανάκτηση: υβριδική αναζήτηση (λεξική + σημασιολογική) και επιλογή των top-k αποσπασμάτων.
- Παραπομπές: κάθε απάντηση συνοδεύεται από τουλάχιστον ένα url ώστε ο χρήστης να μπορεί να επαληθεύει την απάντηση.
- Διαχείριση αβεβαιότητας: στη περίπτωση έλλειψης πηγής ή χαμηλής συνάφειας, το σύστημα εμφανίζει σαφές μήνυμα.
- Ρυθμίσεις από UI: παραμετροποίηση k, n_lex, n_sem, w_sem, temperature.

Μη λειτουργικές απαιτήσεις:

- Ακρίβεια: $\geq 70\%$ χρηστική ορθότητα σε σερ απαντήσεων με 30 ερωτήσεις.
- Χρόνος απόκρισης: 6-15 δευτερόλεπτα end-to-end (αναζήτηση+LLM) σε GPU περιβάλλον.

- Προσβασιμότητα: UI φιλικό προς το χρήστη.

Κεφάλαιο 4: Υλοποίηση

4.1: Web scraping: Πώς συλλέγω τα δεδομένα (modules & helpers)

Στόχος και αρχές σχεδίασης:

Το web scraping μετατρέπει την ιστοσελίδα του τμήματος σε εύκολα αναζητήσιμα αρχεία με πλήρη ιστορικό. Αυτό τηρείτε σύμφωνα με τις εξής αρχές: (α) αναπαραγωγιμότητα δηλαδή όλα τα αρχεία έχουν ίδια δομή εξόδου ανά εκτέλεση. (β) ιχνηλασιμότητα, δημιουργείτε ένα CSV αρχείο ανά ενότητα. (γ) ανθεκτικότητα διότι υπάρχει κοινό περιβάλλον περιήγησης, χειρισμός expanders και lazy-loading, έλεγχοι για κενές σελίδες. (δ) καθαρότητα περιεχομένου μέσω της αφαίρεσης θορύβου.

Ροή εκτέλεσης:

Το app.py λειτουργεί ως σημείο εκκίνησης του συστήματος: φορτώνει τα αρχεία YAML ανά ενότητα (π.χ. students.yaml, actions.yaml κ.λπ.), δημιουργεί ένα κοινό headless Chrome και για κάθε ενότητα καλεί τη συνάρτηση run_sections() του αντίστοιχου module (π.χ. students.py, staff.py κ.λπ.). Κάθε εκτέλεση παράγει έξοδο σε φάκελο της ενότητας μέσα στον τρέχοντα φάκελο run, όπου αποθηκεύονται το pages.csv με τα μεταδεδομένα και αρχεία pages/<sha256>.txt με καθαρό κείμενο. Όπου απαιτείται δημιουργούνται εξειδικευμένα αρχεία csv (π.χ. announcements.csv, items.csv, lessons_plan.csv)

Υποδομή προγράμματος πλοήγησης (Selenium/Chrome):

Το browser.py δίνει ενιαία συμπεριφορά κατά την εκτέλεση του αρχείου. Λειτουργεί χωρίς γραφικό περιβάλλον (headless), καθορισμένο user-agent(Chrome/Mozilla), γλώσσα, χρονικά όρια (timeouts) και μέγεθος παραθύρου. Παρέχει βοηθητικές συναρτήσεις όπως goto(), wait_css(), wait_xpath για αναμονές, scroll_lazyload() για φόρτωση δυναμικού περιεχομένου, expand_common() για άνοιγμα επέκταση σε παρεμφερές συνδέσμους κ.α. Με αυτό το τρόπο όλα τα υποσυστήματα έχουν σταθερή και προβλέψιμη συμπεριφορά, ενώ μειώνονται οι τυχαίες αστοχίες κατά την εκτέλεση.

Εξαγωγή από Document Object Model σε κείμενο:

Το `extract.py` αναλαμβάνει τον καθαρισμό και την απόδοση του περιεχομένου. Με την `ExtractConfig` ορίζουμε ποιο είναι το κύριο περιεχόμενο (`keep_selector`), ποια τμήματα αφαιρούνται (`drop_selector`), αν θα διατηρηθούν πίνακες (`keep_tables`). Η `sanitize_dom()` απομακρύνει τα στοιχεία προς απόκρυψη (π.χ. κεφαλίδα, μενού πλοήγησης, υποσέλιδο, πλαϊνές στήλες κ.α.), η `extract_main_node()` εντοπίζει το «πλουσιότερο» κόμβο με βάση το μήκος του κειμένου και η `inner_text()` αποδίδει το περιεχόμενο των βασικών στοιχείων (όπως επικεφαλίδες h1-h4, παράγραφοι, λίστες, πίνακες κ.α.) σε καθαρό κείμενο. Αν το `keep_tables=true`, οι πίνακες μετατρέπονται σε απλό Markdown. Η `collect_attachments_and_links()` συγκεντρώνει απόλυτους συνδέσμους εγγράφων (pdf, doc, xls, ppt κ.λπ.) και συνδέσμους προς άλλες σελίδες html. Η `extract_content()` επιστρέφει της στήλες για το `pages.csv` και τα αρχεία text στην `pages/*.txt`.

Κοινό schema εξόδου:

Για κάθε ενότητα, παράγεται ένα `pages.csv` με στήλες `url`, `title`, `text_path`, `chars`, `text_hash`, `doc_links`, `follow_links` και ένα αρχείο κειμένου (`text`) ανά σελίδα. Με αυτό το ενιαίο σχήμα, οι επόμενες επεξεργασίες γίνονται ομοιόμορφες όπως η φόρτωση σε βάση δεδομένων, αναζήτηση πλήρους κειμένου FTS5, υπολογισμός των embeddings.

Scoring συνδέσμων και περιορισμοί:

Η ανίχνευση είναι αυστηρά οριοθετημένη ώστε να μένουν στο πεδίο του τμήματος: με τα `allow_prefixes` επιτρέπουμε μόνο συγκεκριμένα μονοπάτια, ενώ με τα `deny_patterns` αποκλείουμε ανεπιθύμητα μονοπάτια. Παραμένουμε στον ίδιο τομέα, με εξαίρεση ειδικές αφετηρίες για την γραμματεία. Τέλος, εφαρμόζουμε ελέγχους για τις σελίδες με κενό περιεχόμενο μέσω λίστας ενδεικτικών φράσεων («Η σελίδα δεν βρέθηκε» κ.α.). Όλοι αυτοί οι περιορισμοί καταγράφονται στα YAML αρχεία ανά ενότητα, ώστε κατά την εκτέλεση να παραμένει καθαρή και πλήρως εντός του ορισμένου πεδίο το περιεχόμενο που ανακτητέ (GeeksForGeeks_yaml, n.d.).

Ενότητες – κοινή λογική & ιδιαιτερότητες:

Όλες οι ενότητες μοιράζονται την ίδια ροή, διαβάζουν τις ρυθμίσεις τους, ανιχνεύουν μόνα τα επιτρεπτά μονοπάτια, φορτώνουν την ιστοσελίδα με κοινό περιβάλλον περιήγησης, καθαρίζουν το DOM ώστε να εξάγουν κείμενο και συνδέσμους. Κάθε ενότητα παράγει συγκεκριμένα αποτελέσματα για την διασφάλιση της ίδιας δομής εξόδου, ιχνηλασιμότητα

και σταθερή συμπεριφορά του έργου. Κάποιες ενότητες, πέρα από το αρχείο pages.csv και τα text αρχεία δημιουργούν κάποια επιπλέον βοηθητικά αρχεία.

Ενότητα ανακοινώσεων: announcements.csv, το αρχείο αυτό περιέχει πληροφορίες όπως ημερομηνίες, κατηγορίες, συνημμένα.

Ενότητα δράσεις: items.csv, το αρχείο είναι παρόμοιο με το αυτό των ανακοινώσεων.

Ενότητα μάθημα: δημιουργούνται 2 επιπλέον αρχεία το lessons_plan.csv όπου περιγράφονται τα μαθήματα ανά εξάμηνο και εκπαιδευτικό πρόγραμμα σπουδών και το lessons_details.csv όπου υπάρχει πλήρης περιγραφή ανά μάθημα.

Λήψη συνημμένων PDF:

Ως post-processing βήμα, ο download_attachments.py σαρώνει όλα τα pages.csv των ενοτήτων, εντοπίζει τα doc_links και κατεβάζει όλα τα αρχεία που έχουν κατάληξη .pdf. Κατά την εκτέλεση, για να λειτουργεί σαν κανονικός browser, στέλνει αναγνωριστικό user-agent και σελίδα προέλευσης (Referer), λαμβάνει αρχικά τα cookies, αν αποτύχει ξαναδοκιμάζει και ελέγχει επαναληπτικές διευθύνσεις (http $\leftrightarrow$ https, με ή χωρίς www) για απευθείας συνδέσμους. Κάθε αρχείο αποθηκεύεται προσωρινά, υπολογίζεται το αποτύπωμα sha-256 και στη συνέχεια μετονομάζεται σε <sha>.pdf. Τα αποτελέσματα καταγράφονται στο file_manifest.csv με πεδία url, file_path, sha_256, content_type, bytes, source_section, source_page_url, status, error_detail, first_seen_at, filename_hint. Το manifest λειτουργεί ως γέφυρα προς το επόμενο στάδιο που επεξεργαζόμαστε τα pdf αρχεία και για την συσχέτιση μεταξύ των σελίδων - αρχείων κατά την εισαγωγή στην βάση.

Γιατί αυτή η προσέγγιση δουλεύει:

Διαχωρίζει τη λογική από τις ρυθμίσεις και επιβάλλει ενιαία έξοδο. Πρώτον, η συντήρηση γίνεται μέσω YAML, δηλαδή αλλάζουμε selectors χωρίς να τροποποιούμε κώδικα Python. Δεύτερον, το κοινό σχήμα εξόδου επιτρέπει ομοιόμορφη εισαγωγή στην βάση, αναζήτηση πλήρους κειμένου (FTS5) και υπολογισμό των embeddings χωρίς ιδιαίτερους μετασχηματισμούς. Τρίτον, τα τεκμήρια βελτιώνουν την ανάκτηση τόσο σε κλασικές μεθόδους (BM25) όσο και σε νευρωνικές (dense retrieval) (Cohere, n.d.). Τέταρτον, διατήρηση πλήρη ιχνηλασιμότητα δεδομένων (doc_links $\rightarrow$ downloader $\rightarrow$ file_manifest.csv $\rightarrow$ PDF $\rightarrow$ Text $\rightarrow$ DB), ώστε κάθε αποτέλεσμα να μπορεί να αποδοθεί στην πηγή του. Τέλος, το σύστημα αντέχει σε εύθραστα περιβάλλοντα διεπαφής χάρης στον

χειρισμό expanders και lazy_loading και, όπου απαιτείται, στην εκτέλεση JS(JavaScript) (π.χ. στις ανακοινώσεις).

4.2: Επεξεργασία pdf στο Google Colab

Σε αυτό το στάδιο μετατρέπουμε όλα τα αρχεία PDF σε καθαρά αρχεία κειμένου text, παράγουμε μεταδεδομένα (manifest) και εφαρμόζουμε οπτική αναγνώριση χαρακτήρων OCR για σελίδες που έχουν εικόνες. Το αποτέλεσμα είναι ενιαίο, ομοιόμορφο σύνολο κειμένων έτοιμο για εισαγωγή στη βάση. Η διαδικασία εκτελείται στο Google Colabs (L4 GPU) για ταχύτερη επεξεργασία, ιδίως στο σκέλος του OCR μοντέλων.

Εκτέλεση και ρυθμίσεις:

Το πρόγραμμα είναι το 00_pdf_processed.py και δέχεται βασικά ορίσματα, --input_dir (φάκελος με τα pdf αρχεία), --output_dir (φάκελος αποτελεσμάτων), --ocr_model_id (OCR μοντέλο), καθώς και --dpi, --min_char, --min_greek_ratio για τη καλύτερη ποιότητα(quality gates). Στο ξεκίνημα δημιουργείτε η ρύθμιση RunConfig και γίνεται μια φορά η αρχικοποίηση του OCR μοντέλου ώστε να αποφεύγεται τα επαναλαμβανόμενα κόστη φόρτωσης μοντέλου. Στο τέλος, γράφεται συγκετρωτικό manifest.csv με τα μεταδεδομένα όλων των pdf αρχείων.

Για κάθε pdf αρχείο ακολουθείτε μια αρχή: πρώτα δοκιμάζεται PyMuPDF text extraction, αν η σελίδα είναι image-only (δηλαδή δεν έχει κάποιο κείμενο), περνάμε σε OCR. Πιο συγκεκριμένα:

Βήμα 1 (PyMuPDF): Χρησιμοποιούμε page.get_text("text"), σε περίπτωση που επιστρέφει μικρό κείμενο, περνάμε σε εναλλακτική ανά block extraction όπου ταξινομούμε κατά (x,y) και ενώνουμε το κείμενο ανά block, έπειτα εφαρμόζεται clean_text(), για τους απαραίτητους καθαρισμούς.

Βήμα 2 (OCR): Αν η σελίδα δεν περιέχει κείμενο, τότε τη μετατρέπουμε σε εικόνα και εφαρμόζουμε OCR με το μοντέλο της Qwen2.5-VL (μέσω transformers) (Souvik Mandal and Ashish Talewar and Paras Ahuja and Prathamesh Juvatkar, n.d.). Χρησιμοποιούμε παραστατικό όριο tokens με βάση την ανάλυση της εικόνας (περίπου 900-1200). Το αποτέλεσμα, περνάει σε clean_text() και καταγράφουμε πόσες σελίδες επεξεργάστηκε.

Μετά από κάθε σελίδα, εφόσον υπάρχει Cuda, γίνεται `torch.cuda.empty_cache()` για να διατηρείτε σταθερή η μνήμη. Στο τέλος του εγγράφου, συντίθεται `full_text` με σαφή διαχωριστικά σελίδων (`--Σελίδα[i]--`).

Quality gates και φίλτρα:

Στο παραγόμενο `full_text` υπολογίζονται δείκτες ποιότητας. `Greek_ratio` ποσοστό ελληνικών χαρακτήρων, είναι χρήσιμο για να διασφαλίσει αρχεία που έχουν θόρυβο είτε λανθασμένο OCR. `Min_chars` ελάχιστος αριθμός για να θεωρηθεί αξιοποιήσιμο κείμενο. Αν κάποιος δείκτης δείξει χαμηλότερο αριθμό από το όριο, ορίζεται μια στήλη στο `manifest`: `skip_from_db = True`. Επιπλέον, υπολογίζεται δείκτης `quality score` για συγκριτική αξιολόγηση.

Outputs ανά pdf:

Για κάθε αρχείο παράγονται, Text: `output_dir/txt/<sha256>.txt`, με σελιδοδείκτες. Metadata(json): `output_dir/meta/<sha256>.json`, που περιλαμβάνει `file`, `pages`, `is_schedule`, `ocr_pages`, `text_chars`, `Greek_ratio`, `quality_score`, `skip_from_db`. Manifest.csv: `output_dir/manifest.csv` συνοψίζει τα μεταδεδομένα και εκτυπώνει μικρή αναφορά (όπως πόσα επεξεργάστηκαν, πόσα ήταν image – only κ.λπ.).

Γέφυρα προς την βάση δεδομένων:

Στο στάδιο της δημιουργίας της βάσης ενοποιούμε τα αποτελέσματα των προηγούμενων βημάτων. Διαβάζουμε τόσο το `file_manifest` από την λήψη συνημμένων όσο και το `output_dir/manifest.csv` από την επεξεργασία. Με αυτά καταχωρούμε όλα τα pdf αρχεία στο πίνακα `pdf_files` και για κάθε σελίδα του ιστότοπου δημιουργούμε τη σύνδεση της με τα αντίστοιχα pdf στον πίνακα `page_pdf_links`. Αφού ολοκληρωθεί η συσχέτιση σελίδας – pdf, παράγουμε τα `corpus_docs` (έγγραφα) και `corpus_chunks` (τμήματα κειμένου) που θα χρησιμοποιηθούν στην αναζήτηση.

4.3: Δημιουργία Βάσης Δεδομένων – Σχήμα, εισαγωγή, chunking και FTS5

Το `build_ditbot_database.py` ενοποιεί τα outputs του web scrapping και pdf processed σε μια SQLite. Εισάγει από τα `pages.csv` όλων των ενοτήτων στους πίνακες `*_pages` (π.χ. `actions_pages`, `students_pages` κ.λπ.) και φτιάχνει ενιαίο `pages_registry`. Απο τα `anouncements.csv`, `items.csv` και `lessons_plan.csv` γεμίζει τους αντίστοιχους πίνακες.

Επιπλέον, δημιουργείται ένας πίνακας `instructor_courses_fact` όπου χρησιμοποιείται για την άμεση συσχέτιση μαθημάτων και καθηγητών. Τα pdf φορτώνονται από τα αρχεία `file_manifest.csv` - `manifest.csv` στο πίνακα `pdf_files` και συνδέονται με σελίδες μέσω `pages_pdf_links`. Κατόπιν δημιουργούνται `corpus_docs` (σελίδα ή pdf) και `corpus_chunks` με τεμαχισμό περίπου 1000 χαρακτήρων και overlap περίπου 180 χαρακτήρων. Πάνω στα chunks χτίζεται FTS5 (`fts_corpus`) για BM25 λεξική αναζήτηση. Παράλληλα, σε δεύτερο βήμα, υπολογίζονται τα embeddings για τα ίδια chunks με το μοντέλο `intfloat/multilingual-e5-large` (με χρήση GPU) μέσω `SentenceTransformer`, σε συστάδες των 2000. Πιο συγκεκριμένα, τα embeddings παράγονται για κάθε `corpus_chunk` και εγγράφονται στην ChromaDB σε παρτίδες 2000 εγγραφών ανα συναλλαγή (`upsert`). Η επιλογή του μεγέθους της παρτίδας επιλέχθηκε ως συμβιβασμός απόδοσης και σταθερότητας για να μειώνει τις κλήσεις προς τον vector store, χωρίς να επιβαρύνει υπερβολικά την μνήμη ή να προκαλέσει timeouts. Τα embeddings αποθηκεύονται σε ChromaDB με `ID = chunk_id` και πλούσιο metadata (`doc_id`, `section`, `best_url`, `offsets`, `text_hash`) ώστε τα αποτελέσματα να φαίνονται άμεσα με τις πηγές. Η ομοιομορφία που εφαρμόζεται στο chunking για FTS5 και για τα embeddings εξασφαλίζει ότι δύο κανάλια επιστρέφουν συγκρίσιμες μονάδες κειμένου. Όλα τα paths είναι σχετικά και τα κείμενα - αρχεία ταυτοποιούνται με hashes για idempotency. Το αποτέλεσμα είναι ένα ενιαίο, ιχνηλάσιμο RAG corpus έτοιμο για υβριδική ανάκτηση: (α) λεξικό κανάλι αναζήτηση πάνω στη SQLite και (β) σημασιολογικό κανάλι (ChromaDB + e5-large). Στο επόμενο στάδιο 4.4 τα δύο κανάλια συγχωνεύονται με `Reciprocal Rank Fusion`, ώστε η τελική ανάκτηση να συνδυάζει λέξεις κλειδιά για την ακρίβεια και εννοιολογική κάλυψη.

κατάταξη. Αν το όριο των tokens το απαιτεί, κάνουμε trimming και αφαιρούμε διπλότυπα. Σε περίπτωση που τα ευρήματα είναι οριακά, το σύστημα σηματοδοτεί χαμηλή εμπιστοσύνη ώστε η παραγωγή να παραμείνει συντηρητική.

Βάρη/Παράμετροι της υβριδικής ανάκτησης:

n_lex: Πόσα κορυφαία αποτελέσματα κρατάμε από το λεξικό κανάλι (BM25/FTS5) πριν τη συγχώνευση, Όσο μεγαλύτερο n_lex, τόσο μεγαλύτερη κάλυψη έχουμε σε λέξεις κλειδιά αλλά πιθανόν περισσότερο θόρυβο.

n_sem: Πόσα κορυφαία αποτελέσματα κρατάμε από το σημασιολογικό κανάλι (embeddings/Chroma) πριν την συγχώνευση. Όσο μεγαλύτερο n_sem, καλύτερη εννοιολογική κάλυψη αλλά αυξάνει το κόστος ελέγχου.

w_sem: Βάρος του σημασιολογικού καναλιού κατά τη συγχώνευση RRF. Υψηλότερος δείκτης, δίνει προτεραιότητα στην νοηματική ομοιότητα έναντι των ακριβών όρων.

k: Τελικός αριθμός αποσπασμάτων που θα περάσουν στο prompt. Μικρότερα k, μπορεί να χάνει πλαίσιο αλλά με μεγαλύτερο k μπορεί να γεμίσει άσκοπα το prompt.

rrf_k: Σταθερά εξομάλυνσης της RRF στον τύπο $(1 / rrf_k + rank)$. Μεγαλύτερο rrf_k μαλακώνει τις διαφορές μεταξύ θέσεων και δίνει πιο ισορροπημένη συγχώνευση.

neighbor_radius: Πόσοι γείτονες (chunks) του ίδιου εγγράφου θα προστεθούν γύρω από τη κάθε εύρεση. Αυξάνοντας το δείκτη παίρνεις περισσότερο πλαίσιο και καταναλώνετε περισσότερα tokens.

diversity_cap_per_doc: Μέγιστος αριθμός chunks ανά έγγραφο. Αποτρέπει να κυριαρχήσει μία πηγή στο context.

Παραγωγή (prompt → LLM):

Το prompt οργανώνεται σε τρία μέρη: System , Context, User. Το System, ορίζει κανόνες συμπεριφοράς, απαντά αποκλειστικά βάσει του παρεχόμενου context, δηλώνει αβεβαιότητα όταν οι πηγές δεν επαρκούν και παραθέτει σαφείς παραπομπές. Το Context, περιλαμβάνει τα επιλεγμένα αποσπάσματα σε σειρά κατάταξης μαζί με μεταδεδομένα και urls για επαλήθευση. Το User, είναι το αρχικό ερώτημα του χρήστη που τηρείται αυτούσιο ώστε το μοντέλο να στοχεύσει ακριβώς στο ζητούμενο. Αφού γίνουν οι απαραίτητοι έλεγχοι,

εφαρμόζεται token budgeting, αν ξεπεράσει το όριο tokens, συντομεύονται τα αποσπάσματα ή αφαιρούνται τα χαμηλότερα στην κατάταξη αναζήτησης. Το μοντέλο παραγωγής είναι το `ilsp/Llama-KriKri-8b-Instruct` (Dimitris Roussis and Leon Voukoutis and Georgios Paraskevanopoulos and Sokratis Sofianopoulos and Prokopis Prokopidis and Vassilis Papavasileiou and Athanasios Katsamanis and Stelios Piperidis and Vassilis Katsouros, n.d.), στην έξοδο γίνεται καθαρισμός τυχόν θορύβων και τυποποίηση των παραπομπών.

Αβεβαιότητα και ασφάλεια:

Όταν το recall είναι χαμηλό, το σύστημα επιστρέφει επιφυλακτικό μήνυμα ή παραθέτει μόνο σχετικές πηγές. Δεν παράγει εικασίες εκτος του διαθέσιμου context.

Οφέλη:

Η RRF συνδυάζει την ακρίβεια της BM25 και τη νοηματική κάλυψη των embeddings, το πειθαρχημένο prompt οδηγεί σε σύντομες και επαληθεύσιμες απαντήσεις. Τέλος, τα ρυθμιζόμενα βάρη επιτρέπουν να προσαρμόζεται η συνεισφορά κάθε καναλιού στις ανάγκες του εκάστοτε ερωτήματος.

4.5: Εκτέλεση & User Interface – Gradio app, ρυθμίσεις, ροή αιτήματος

Το `app_gradio.py` λειτουργεί ως κέλυφος που ενώνει τα στάδια προετοιμασίας και παραγωγής: φορτώνει τους πόρους (βάση δεδομένων, embeddings, μοντέλο), προβάλλει ένα απλό Chat UI και κατευθύνει το αίτημα από την είσοδο μέχρι την τελική απάντηση. Κατά την εκκίνηση γίνεται ανάγνωση των ορισμάτων (διαδρομή της βάσης, φάκελος Chroma, όνομα συλλογής, μοντέλο για embeddings και LLM), δημιουργείται ένα αντικείμενο ρυθμίσεων που ακολουθεί η αρχικοποίηση δύο βασικών συνιστωσών: του HybridSearcher (απο το `hybrid_search.py`), που ανοίγει τη βάση, συνδέεται στο persistent συλλογή της Chroma και φορτώνει το encoder `e5-large` για κωδικοποίηση queries, και του LLM pipeline (απο το `generator.py`), που φωρτονει το LLM με τον αντίστοιχο tokenizer. Στη συνέχεια χτίζεται το περιβάλλον Gradio: ένα πλαίσιο συνομιλίας με πεδίο εισαγωγής και μια πλευρική στήλη με sliders για τις παραμέτρους ανάκτησης (`k`, `n_lex`, `n_sem` κ.λπ.) και παραγωγής (`max_tokens`, `temperature`).

Η βασική ροή ενεργοποιείται κάθε φορά που ο χρήστης πατά Send. Το callback respond() προσθέτει το μήνυμα στο ιστορικό και καθαρίζει ελαφρά το input. Έπειτα καλεί το HybridSearcher με τις τρέχουσες τιμές των sliders. Εκεί, εκτελείται η υβριδική αναζήτηση (βλ. 4.4), επιστρέφοντας μια λίστα από επιλεγμένα chunk_id. Με τα επιλεγμένα αποτελέσματα, το respond() συνθέτει το context που θα τροφοδοτήσει το μοντέλο. Για κάθε chunk_id ανακτά από τη βάση το κείμενο του και τα μεταδεδομένα του και παράγει σύντομα αποσπασματα (βλ. 4.4). Στο επόμενο βήμα, το respond() χτίζει το prompt σε τρία τμήματα (System => Context => User). Το System κλειδώνει τους κανόνες, το Context συγκεντρώνει τα αποσπάσματα που επιλέχθηκαν από την RRF κατάταξη με τις αντίστοιχες πηγές. Το User περιέχει αυτούσια το ερώτημα. Τέλος γίνεται έλεγχος αν το σύνολο των κανόνων και του context ξεπερνούν το όριο για να γίνουν οι απαραίτητες αλλαγές (βλ. 4.4). Η παραγωγή γίνεται μέσω του LLM pipeline του [generator.py](#), το οποίο έχει ήδη φορτωθεί κατά την εκκίνηση. Η κλήση generate() λαμβάνει το τελικό prompt και τις ενεργές ρυθμίσεις εξόδου (max tokens, temperature) και επιστρέφει το καθαρό κείμενο της απάντησης. Στη συνέχεια, εφαρμόζεται ελαφρύ post-processing ώστε τα αποτελέσματα να είναι άμεσα αναγνώσιμα (μείωση θορύβου). Μαζί με το κείμενο, προβάλλονται οι παραπομπές, έτσι ο χρήστης μπορεί να επαληθεύσει κάθε ισχυρισμό ακολουθώντας τον σύνδεσμο στην πρωτογενή σελίδα ή στο αντίστοιχο pdf (βλ. 4.3).

Η διαχείριση δομημένων ερωτημάτων (π.χ. “ποιος διδάσκει το X”, “σε ποιο εξάμηνο το Y”) γίνεται πριν την υβριδική διαδικασία: το response() ζητά από τον HybridSearcher ανίχνευση αν στο ερώτημα υπάρχουν λέξεις όπως “μάθημα, εξάμηνο, διδάσκει”, όπου εντοπιστεί, εκτελούνται άμεσες SQL κλήσεις στους σχετικούς πίνακες της βάσης (π.χ. instructor_courses_fact, lessons_plan) και σχηματίζει structured context χωρίς RAG. Η επιλογή αυτή μειώνει τον χρόνο απόκρισης και τον κίνδυνο ασάφειας, αφού η απάντηση προκύπτει από τους επίσημους πίνακες αντί για αποσπάσματα. Αν αυτή η διαδικασία αποτύχει, τότε λειτουργεί με την υβριδική αναζήτηση χρησιμοποιώντας RAG.

Κατά την συνεδρία, το Gradio διατηρεί το ιστορικό διαλόγου και τις τελευταίες ρυθμίσεις έτσι ώστε κάθε νέο μήνυμα να συνεχίζει ομαλά τη ροή (Gradio, n.d.). Παρέχεται ακόμη εξαγωγή σε Txt αρχείο, ώστε ο χρήστης να αποθηκεύσει άμεσα τη συζήτηση με τις παραπομπές της.

Σε επίπεδο ανθεκτικότητας και χειρισμού αβεβαιότητας, όταν το recall είναι χαμηλό ή τα αποτελέσματα ασθενή, το σύστημα προτιμά συντηρητική έξοδο με σύντομο

προειδοποιητικό μήνυμα (βλ. 4.4). Αν αποτύχει κάποια εξάρτηση, επιστρέφει ευγενές μήνυμα και διατηρείται ότι έχει ήδη ανακτηθεί, ώστε ο χρήστης να έχει πρακτική αξία ακόμη και σε σενάρια σφάλματος.

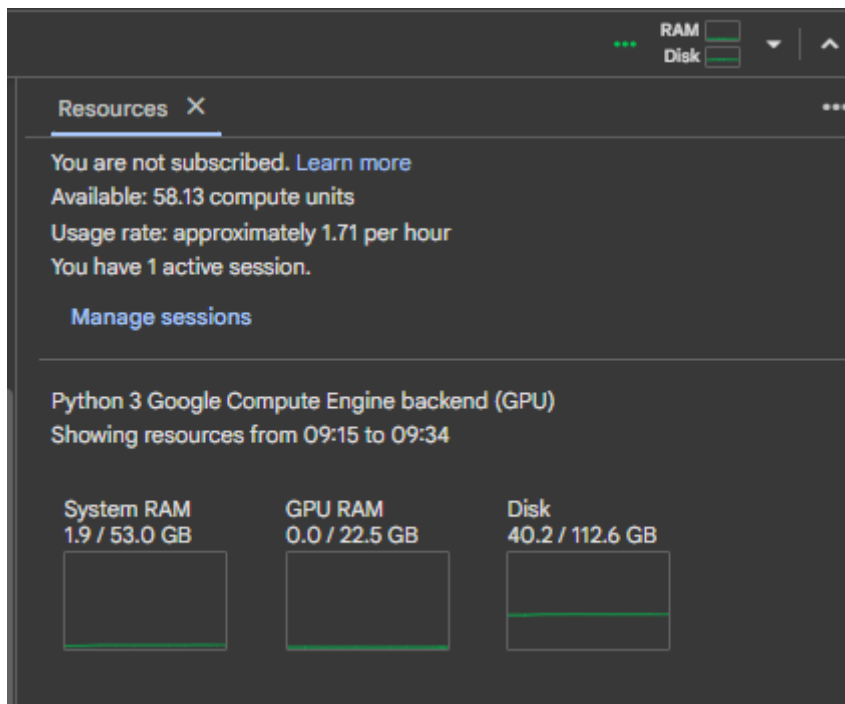
Τέλος, αξίζει να επισημανθεί ότι το `app_gradio.py` συγκεντρώνει την ευθύνη της διεπαφής και του προσανατολισμού κλήσεων, ενώ η ουσία της αναζήτησης και παραγωγής παραμένει στους άλλους κώδικες. Αυτό σημαίνει ότι μελλοντικές βελτιώσεις, όπως προσθήκη άλλου LLM, διαφορετική στρατηγική συγχώνευσης, νέα φίλτρα μπορούν να ενσωματωθούν χωρίς αλλαγές στη ροή του UI. Ομοίως, η ίδια υπηρεσία μπορεί να ενσωματωθεί σε άλλο REST endpoint διατηρώντας το ίδιο μοτίβο : δίνεις ερώτημα + ρυθμίσεις => παίρνεις απάντηση + παραπομπές. Με αυτή την οργάνωση, η εκτέλεση μέσω Gradio παραμένει προβλέψιμη, ρυθμίσιμη και επεκτάσιμη. Ο χρήστης ελέγχει τη συμπεριφορά με απλά sliders, το σύστημα φροντίζει την ποιότητα και κάθε απάντηση συνοδεύεται με τις πηγές του. Σε συνέχεια του προβλήματος που παρουσιάζεται (βλ. 1.2), αυτό ακριβώς επιδιώκει το εκπαιδευτικό chatbot: σύντομες και αξιόπιστες απαντήσεις με σαφείς παραπομπές, μέσα από ροή που σέβεται τους περιορισμούς των μοντέλων και παρέχει στον χρήστη ουσιαστικό έλεγχο.

4.6: Παραδείγματα και Εκτέλεση του έργου

Ως τεκμηρίωση της end-to-end ροής, από την συλλογή περιεχομένου (web scraping) και τη μετατροπή των PDF σε κείμενο, έως την ενοποίηση σε βάση, τη δημιουργία embeddings, την υβριδική ανάκτηση και το περιβάλλον διεπαφής – Παρατίθενται ενδεικτικά στιγμιότυπά από Google Colab και Gradio.

Εκτέλεση ροής:

1. Πόροι Colab: επιβεβαίωση διαθέσιμης RAM : 53 GB – GPU 22.5 GB – Disk 112.6 GB.



Εικόνα 4. Πόροι Colab (L4 GPU).

2. Εγκατάσταση requirements και φόρτωση κλειδιού:

```

Load models, install requirements.txt, load key

[1] ✓ 0s !unzip -q /content/Final_Project_ChatBot.zip -d /content/

[2] ✓ 10s !pip install -q "urllib3<2.4.0" \
        selenium lxml cssselect beautifulsoup4 PyYAML requests \
        pymupdf pillow numpy pandas tqdm

----- 128.4/128.4 kB 6.3 MB/s eta 0:00:00
----- 9.4/9.4 MB 131.3 MB/s eta 0:00:00
----- 24.1/24.1 MB 118.7 MB/s eta 0:00:00
----- 512.7/512.7 kB 45.4 MB/s eta 0:00:00

[3] ✓ 0s from dotenv import load_dotenv
import os
load_dotenv()
HF_TOKEN= os.getenv("HF_TOKEN")
  
```

Εικόνα 5. Φόρτωση εξαρτήσεων.

3. Web scraping: εκτέλεση του app.py με sections & configs για πλοήγηση, εξαγωγή κειμένου και συλλογή συνδέσμων (βλ. 4.1).

```
Web scraping of dit.uoi.gr

Run the webscraper

[4] # 2) go to your project folder
    ✓ 0s !cd /content/Final_Project_ChatBot/WebScraper

[5] # 3) run the scraper
    ✓ 19m !python /content/Final_Project_ChatBot/WebScraper/app.py \
        --root_out /content/Final_Project_ChatBot/ChatBot \
        --run_name scraped_data \
        --headless \
        --sections students,professors,secretariat,actions,announcements,lessons,fixed,staff \
        --config_dir /content/Final_Project_ChatBot/WebScraper/configs \
        # --download_attachments \
        # --files_dest /content/Final_Project_ChatBot/ChatBot/scraped_data/pdfs

[nav] https://www.dit.uoi.gr/staff/profile?id=44
      extracted chars=1685 kept_nodes=97 dropped_nodes=46 attachments=1
[staff] visiting 26: https://www.dit.uoi.gr/staff/profile?id=48
[nav] https://www.dit.uoi.gr/staff/profile?id=48
      extracted chars=265 kept_nodes=46 dropped_nodes=46 attachments=0
[staff] visiting 27: https://www.dit.uoi.gr/staff/profile?id=41
[nav] https://www.dit.uoi.gr/staff/profile?id=41
      extracted chars=1700 kept_nodes=117 dropped_nodes=46 attachments=1
[staff] visiting 28: https://www.dit.uoi.gr/staff/profile?id=45
[nav] https://www.dit.uoi.gr/staff/profile?id=45
      extracted chars=2196 kept_nodes=124 dropped_nodes=46 attachments=1
[staff] visiting 29: https://www.dit.uoi.gr/staff/profile?id=47
[nav] https://www.dit.uoi.gr/staff/profile?id=47
      extracted chars=11420 kept_nodes=227 dropped_nodes=46 attachments=1
[staff] visiting 30: https://www.dit.uoi.gr/staff/profile?id=42
[nav] https://www.dit.uoi.gr/staff/profile?id=42
      extracted chars=267 kept_nodes=46 dropped_nodes=45 attachments=0
[staff] visiting 31: https://www.dit.uoi.gr/staff/profile?id=46
[nav] https://www.dit.uoi.gr/staff/profile?id=46
      extracted chars=3858 kept_nodes=162 dropped_nodes=46 attachments=0
[staff] visiting 32: https://www.dit.uoi.gr/staff/profile?id=43
[nav] https://www.dit.uoi.gr/staff/profile?id=43
      extracted chars=994 kept_nodes=100 dropped_nodes=46 attachments=0
[staff] visiting 33: https://www.dit.uoi.gr/staff/ctsl.dit.uoi.gr/
[nav] https://www.dit.uoi.gr/staff/ctsl.dit.uoi.gr/
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/ctsl.dit.uoi.gr/
[staff] visiting 34: https://www.dit.uoi.gr/staff/chgogos.github.io
[nav] https://www.dit.uoi.gr/staff/chgogos.github.io
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/chgogos.github.io
[staff] visiting 35: https://www.dit.uoi.gr/staff/www.asl.dit.uoi.gr
[nav] https://www.dit.uoi.gr/staff/www.asl.dit.uoi.gr
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/www.asl.dit.uoi.gr
[staff] visiting 36: https://www.dit.uoi.gr/staff/kic.uoi.gr
[nav] https://www.dit.uoi.gr/staff/kic.uoi.gr
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/kic.uoi.gr
[staff] visiting 37: https://www.dit.uoi.gr/staff/scholar.google.com/citations?user=Jrpqo_QAAAAJ&hl=el
[nav] https://www.dit.uoi.gr/staff/scholar.google.com/citations?user=Jrpqo_QAAAAJ&hl=el
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/scholar.google.com/citations?user=Jrpqo_QAAAAJ&hl=el
[staff] visiting 38: https://www.dit.uoi.gr/staff/lessons.php?id=2 target=
[nav] https://www.dit.uoi.gr/staff/lessons.php?id=2 target=
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/lessons.php?id=2 target=
[staff] visiting 39: https://www.dit.uoi.gr/staff/lessons.php?id=20 target=
[nav] https://www.dit.uoi.gr/staff/lessons.php?id=20 target=
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/lessons.php?id=20 target=
[staff] visiting 40: https://www.dit.uoi.gr/staff/lessons.php?id=45 target=
[nav] https://www.dit.uoi.gr/staff/lessons.php?id=45 target=
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/lessons.php?id=45 target=
[staff] visiting 41: https://www.dit.uoi.gr/staff/dit.uoi.gr
[nav] https://www.dit.uoi.gr/staff/dit.uoi.gr
      [skip] 404-like/not-found: https://www.dit.uoi.gr/staff/dit.uoi.gr
[staff] visiting 42: https://www.dit.uoi.gr/staff/profile/?id=45
[nav] https://www.dit.uoi.gr/staff/profile/?id=45
      extracted chars=2196 kept_nodes=124 dropped_nodes=46 attachments=1
[browser] closed
[staff] ✓ done at 2025-10-13 06:37:33
[browser] closed

[aggregate] all sections completed → /content/Final_Project_ChatBot/ChatBot/scraped_data
```

Εικόνα 6. Εξαγωγή των δεδομένων από την ιστοσελίδα του τμήματος.

4. Λήψη συνημμένων: εκτέλεση του download_attachments.py αποθήκευση των pdf και δημιουργία file_manifest.csv (βλ. 4.1)


```
download the pdf files.

[6]
!python /content/Final_Project_ChatBot/WebScraper/tools/download_attachments.py \
--root_out /content/Final_Project_ChatBot/ChatBot/scraped_data/ \
--dest_dir /content/Final_Project_ChatBot/ChatBot/scraped_data/pdfs

[0388/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/da1_aitisi-ypopsifiotitas-did- f.pdf -> ok
[0389/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/da2_-aitisi-ananeosis-eggrafis-did- f.docx -> ok
[0390/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/da2_-aitisi-ananeosis-eggrafis-did- f.pdf -> ok
[0391/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/da3_aitisi-anagoreysis-did- f.docx -> ok
[0392/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/da3_aitisi-anagoreysis-did- f.pdf -> ok
[0393/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a3a-enarxi-diplomatikis-ergasias.docx -> ok
[0394/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a3a-enarxi-diplomatikis-ergasias.pdf -> ok
[0395/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a3v-praktiko-diplomatikis-ergasias.docx -> ok
[0396/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a3v-praktiko-diplomatikis-ergasias.pdf -> ok
[0397/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2022/11/a9v-sygentr-katagrafi-paraponon-foititon-pl.docx -> ok
[0398/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2022/11/a9v-sygentr-katagrafi-paraponon-foititon-pl.pdf -> ok
[0399/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a10a-entypo katagrafis synantisis-symvovlos spoydon.docx -> ok
[0400/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a10a-entypo katagrafis synantisis-symvovlos spoydon.pdf -> ok
[0401/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a10v-synoptiko-entypo synantiseon-symvovlos spoydon.docx -> ok
[0402/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a10v-synoptiko-entypo synantiseon-symvovlos spoydon.pdf -> ok
[0403/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/12/v10-vevaiosi-symmetochis-se-exetaseis-1.docx -> ok
[0404/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/12/v10-vevaiosi-symmetochis-se-exetaseis-1.pdf -> ok
[0405/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/me2-praktiko-exetasis-diplomatikis-ergasias-pms- f.docx -> ok
[0406/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/me2-praktiko-exetasis-diplomatikis-ergasias-pms- f.pdf -> ok
[0407/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a5-entypo-gia-ekkremitites-me-foititikes-katoikies.docx -> ok
[0408/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a5-entypo-gia-ekkremitites-me-foititikes-katoikies.pdf -> ok
[0409/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a6-entypo-gia-ekkremitites-me-vivliothiki.docx -> ok
[0410/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2019/11/a6-entypo-gia-ekkremitites-me-vivliothiki.pdf -> ok
[0411/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/de1-orismos-trimeloy-symvovleytikis-epitropis-did- f.docx -> ok
[0412/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/de1-orismos-trimeloy-symvovleytikis-epitropis-did- f.pdf -> ok
[0413/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/de2-orismos-eptameloy-exetastikis-epitropis-did- f.docx -> ok
[0414/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/de2-orismos-eptameloy-exetastikis-epitropis-did- f.pdf -> ok
[0415/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/me1-orismos-trimeloy-exetastikis-epitropis-pms- f.docx -> ok
[0416/0465] secretariat https://oldsite.uyi.gr/wp-content/uploads/2020/01/me1-orismos-trimeloy-exetastikis-epitropis-pms- f.pdf -> ok
[0440/0465] fixed http://83.212.170.184/files/Proedr Diat Epag Dik.pdf -> ok
[0441/0465] fixed https://www.dit.uyi.gr/files/fek_metadidaktoriki.pdf -> ok
[0442/0465] fixed https://www.dit.uyi.gr/files/gnorimia.pdf -> ok
[0443/0465] fixed https://www.dit.uyi.gr/files/b7_kanonismos_politikis_piottas2022.pdf -> ok
[0446/0465] fixed https://www.dit.uyi.gr/files/2017pps_tmpte.pdf -> ok
[0447/0465] fixed https://www.dit.uyi.gr/files/2017tmpte.pdf -> ok
[0448/0465] fixed https://www.dit.uyi.gr/files/2018pps_tmpte.pdf -> ok
[0449/0465] fixed https://www.dit.uyi.gr/files/2018tmpte.pdf -> ok
[0450/0465] fixed https://www.dit.uyi.gr/files/2019pps_tmpte.pdf -> ok
[0451/0465] fixed https://www.dit.uyi.gr/files/2019tmpte.pdf -> ok
[0452/0465] fixed https://www.dit.uyi.gr/files/2020_pps_tmpte.pdf -> ok
[0453/0465] fixed https://www.dit.uyi.gr/files/2020_pps_tpkt.pdf -> ok
[0454/0465] fixed https://www.dit.uyi.gr/files/2020_tpkt.pdf -> ok
[0455/0465] fixed https://www.dit.uyi.gr/files/ethae2022_dit.pdf -> ok
[0456/0465] fixed https://www.dit.uyi.gr/files/ethae2022_pms.pdf -> ok
[0457/0465] fixed https://www.dit.uyi.gr/files/ethae2022_te1.pdf -> ok
[0458/0465] fixed https://www.dit.uyi.gr/files/stoxothesia_1920.pdf -> ok
[0459/0465] fixed https://www.dit.uyi.gr/files/stoxothesia_2021.pdf -> ok
[0460/0465] fixed https://www.dit.uyi.gr/files/stoxothesia_2122.pdf -> ok
[0461/0465] fixed https://www.dit.uyi.gr/files/epixirisiako_sxedio_tetraetias_202226.pdf -> ok
[0462/0465] fixed https://www.dit.uyi.gr/files/axiologiseis2021earv2.pdf -> ok
[0463/0465] fixed https://www.dit.uyi.gr/files/axiologiseis2021xeimv2.pdf -> ok
[0464/0465] fixed https://www.dit.uyi.gr/files/axiologiseis2122xeimv2.pdf -> ok
[0465/0465] fixed https://www.dit.uyi.gr/files/%CE%A3%CF%84%CE%B8%CF%87%CE%B5%CE%AF%CE%B1_%CE%9F%CF%81%CE%B3%CE%AC%CE%B0%CF%89%CF%

Scanned doc_links: 465 | manifest rows: 331 | ok total: 330 | error total: 1
Manifest: /content/Final_Project_ChatBot/ChatBot/scraped_data/file_manifest.csv
Files dir: /content/Final_Project_ChatBot/ChatBot/scraped_data/pdfs
```

Εικόνα 7. Λήψη των αρχείων.

5. Λήψη επιπλέον βιβλιοθηκών απαραίτητες για την εκτέλεση των μοντέλων.

```
Preprocess of pdf files into text.

[7] 41s
!pip install -q --upgrade pip
!pip install -q chromadb sentence-transformers fastapi uvicorn gradio sqlite-utils numpy pandas python-dotenv pymupdf tiktoken rapidfuzz unicode tqdm
# για το μοντέλο E5 & το llama (μέσω transformers/accelerate)
!pip install -q transformers accelerate torch torchvision torchaudio
!pip install -q bitsandbytes accelerate
!apt-get -y install ghostscript python3-tk

Preparing to unpack .../08-libijs-0.35-0.35-15build2_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-15build2) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../09-libjbig2dec0_0.19-3build2_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.19-3build2) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../10-libgs9_9.55.0-dfsg1-0ubuntu5.13_amd64.deb ...
Unpacking libgs9:amd64 (9.55.0-dfsg1-0ubuntu5.13) ...
Selecting previously unselected package ghostscript.
Preparing to unpack .../11-ghostscript_9.55.0-dfsg1-0ubuntu5.13_amd64.deb ...
Unpacking ghostscript (9.55.0-dfsg1-0ubuntu5.13) ...
Selecting previously unselected package python3-tk:amd64.
Preparing to unpack .../12-python3-tk_3.10.8-1-22.04_amd64.deb ...
Unpacking python3-tk:amd64 (3.10.8-1-22.04) ...
Setting up tk8.6-blt2.5 (2.5.3+dfsg-4.1build2) ...
Setting up fonts-noto-mono (20201225-1build1) ...
Setting up libijs-0.35:amd64 (0.35-15build2) ...
Setting up blt (2.5.3+dfsg-4.1build2) ...
Setting up python3-tk:amd64 (3.10.8-1-22.04) ...
Setting up fonts-urw-base35 (20200910-1) ...
Setting up poppler-data (0.4.11-1) ...
Setting up libjbig2dec0:amd64 (0.19-3build2) ...
Setting up libidn12:amd64 (1.38-4ubuntu1) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Setting up libgs9-common (9.55.0-dfsg1-0ubuntu5.13) ...
Setting up libgs9:amd64 (9.55.0-dfsg1-0ubuntu5.13) ...
Setting up ghostscript (9.55.0-dfsg1-0ubuntu5.13) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_openc1.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero_v2.so.0 is not a symbolic link

Processing triggers for man-db (2.10.2-1) ...
```

Εικόνα 8. Φόρτωση εξαρτήσεων για τα γλωσσικά μοντέλα.

6. Preprocess των αρχείων pdf: εκτέλεση 00_pdf_processors.py (βλ. 4.2).

Preprocess of pdf files into text.

```
[7] 41s !pip install -q --upgrade pip
!pip install -q chromadb sentence-transformers fastapi uvicorn gradio sqlite-utils numpy pandas pyth
# για το μοντέλο ES & το llama (μείω transformers/accelerate)
!pip install -q transformers accelerate torch torchvision torchaudio
!pip install -q bitsandbytes accelerate
!apt-get -y install ghostscript python3-tk
```

Show hidden output

```
[8] 38m !python /content/Final_Project_ChatBot/WebScraper/00_pdf_processors.py \
--input_dir /content/Final_Project_ChatBot/ChatBot/scraped_data/pdfs \
--output_dir /content/Final_Project_ChatBot/ChatBot/scraped_data/pdf_processed
```

model-00001-of-00002.safetensors: 61% 3.03G/5.00G [00:39:00:45, 43.1MB/s]

model-00001-of-00002.safetensors: 63% 3.13G/5.00G [00:39:00:21, 85.4MB/s]

model-00001-of-00002.safetensors: 65% 3.26G/5.00G [00:39:00:11, 151MB/s]

model-00001-of-00002.safetensors: 66% 3.32G/5.00G [00:39:00:08, 189MB/s]

model-00001-of-00002.safetensors: 69% 3.44G/5.00G [00:39:00:05, 281MB/s]

model-00001-of-00002.safetensors: 70% 3.52G/5.00G [00:40:00:07, 199MB/s]

model-00001-of-00002.safetensors: 72% 3.60G/5.00G [00:43:00:20, 67.3MB/s]

model-00001-of-00002.safetensors: 75% 3.75G/5.00G [00:43:00:11, 113MB/s]

model-00001-of-00002.safetensors: 77% 3.83G/5.00G [00:43:00:08, 139MB/s]

model-00001-of-00002.safetensors: 78% 3.89G/5.00G [00:43:00:06, 169MB/s]

model-00001-of-00002.safetensors: 81% 4.03G/5.00G [00:43:00:03, 258MB/s]

model-00001-of-00002.safetensors: 82% 4.11G/5.00G [00:44:00:02, 302MB/s]

model-00001-of-00002.safetensors: 84% 4.19G/5.00G [00:45:00:04, 186MB/s]

model-00001-of-00002.safetensors: 85% 4.25G/5.00G [00:47:00:10, 73.3MB/s]

model-00001-of-00002.safetensors: 87% 4.33G/5.00G [00:47:00:06, 99.9MB/s]

model-00001-of-00002.safetensors: 90% 4.52G/5.00G [00:47:00:02, 181MB/s]

model-00001-of-00002.safetensors: 92% 4.58G/5.00G [00:47:00:01, 209MB/s]

model-00001-of-00002.safetensors: 93% 4.65G/5.00G [00:48:00:01, 203MB/s]

model-00001-of-00002.safetensors: 94% 4.72G/5.00G [00:48:00:01, 180MB/s]

model-00001-of-00002.safetensors: 95% 4.76G/5.00G [00:49:00:01, 168MB/s]

model-00001-of-00002.safetensors: 96% 4.80G/5.00G [00:49:00:01, 185MB/s]

model-00001-of-00002.safetensors: 97% 4.85G/5.00G [00:49:00:00, 206MB/s]

model-00001-of-00002.safetensors: 98% 4.90G/5.00G [00:50:00:00, 104MB/s]

model-00001-of-00002.safetensors: 100% 5.00G/5.00G [00:51:00:00, 96.4MB/s]

Fetching 2 files: 100% 2/2 [00:52:00:00, 26.04s/it]

Loading checkpoint shards: 100% 2/2 [00:01:00:00, 1.04it/s]

generation_config.json: 100% 214/214 [00:00:00:00, 2.45MB/s]

Processing PDFs: 28% 77/275 [00:31:01:00, 3.27it/s]The following generation flags are not valid at

Processing PDFs: 100% 275/275 [37:05:00:00, 8.09s/it]

Processed 275 PDFs - kept (not skip from db): 259, schedule-like: 0

Outputs at: /content/Final_Project_ChatBot/ChatBot/scraped_data/pdf_processed

You are not subscribed. Learn more

Available: 56.71 compute units

Usage rate: approximately 1.71 per hour

You have 1 active session.

Manage sessions

Python 3 Google Compute Engine backend (GPU)

Showing resources from 09:15 to 10:29

System RAM	GPU RAM	Disk
1.7 / 53.0 GB	0.0 / 22.5 GB	54.7 / 112.6 GB

Εικόνα 9. Επεξεργασία αρχείων pdf.

7. Δημιουργία σχήματος Βάσης Δεδομένων: εκτέλεση build_ditbot_database.py (βλ.4.3).

▼ Create a schema for the database

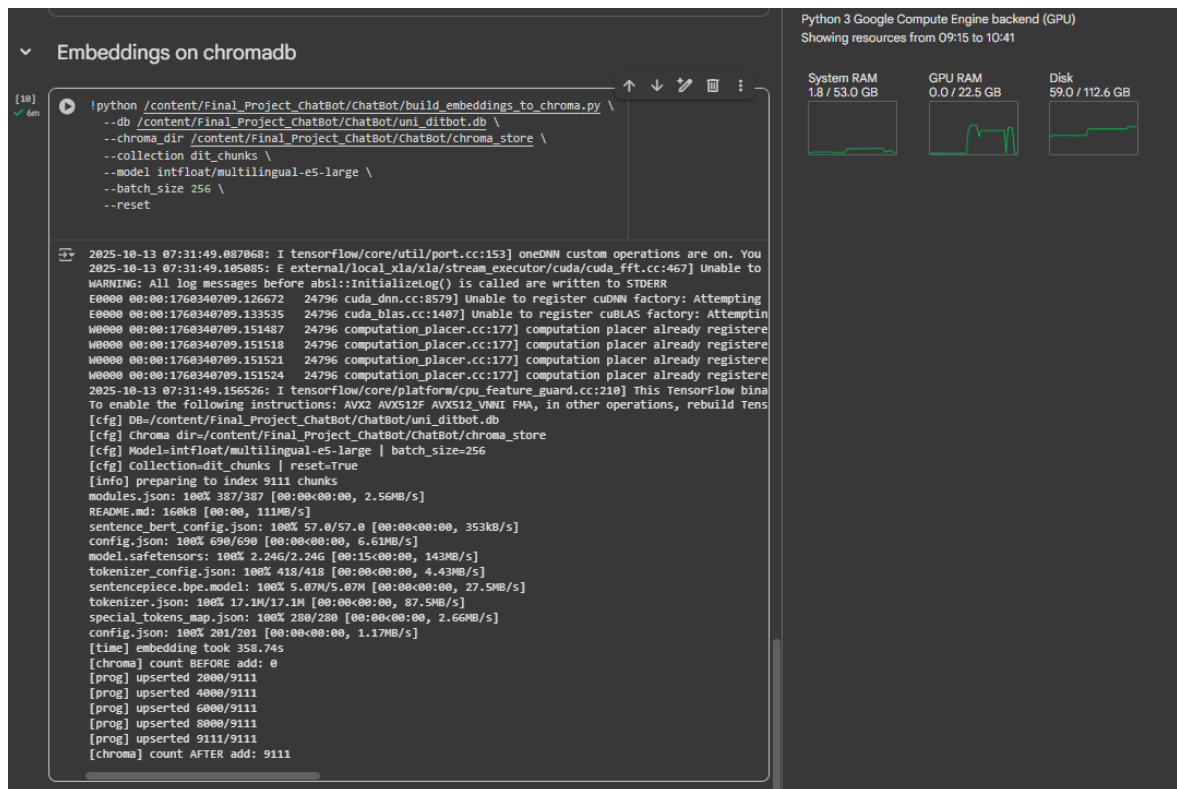
[9]
✓ 1s

```
!python /content/Final_Project_ChatBot/CreateDatabaseSQLite/build_ditbot_database.py \  
--db /content/Final_Project_ChatBot/ChatBot/uni_ditbot.db \  
--run_dir /content/Final_Project_ChatBot/ChatBot/scraped_data
```

```
[cfg] RUN_DIR=/content/Final_Project_ChatBot/ChatBot/scraped_data  
[cfg] DB_PATH=/content/Final_Project_ChatBot/ChatBot/uni_ditbot.db  
[pages] fixed: inserted 26  
[pages] secretariat: inserted 7  
[pages] students: inserted 20  
[pages] professors: inserted 16  
[pages] staff: inserted 33  
[pages] lessons: inserted 116  
[pages] actions: inserted 59  
[pages] announcements: inserted 266  
[items] actions inserted 59  
[items] announcements inserted 266  
[lessons] plan inserted 128  
[registry] rebuilt rows=543  
[pdf] pdf_files inserted 330  
[pdf] page+pdf links 330 (examined 331)  
[corpus] page_docs: 539  
[corpus] pdf_docs : 275  
[corpus] chunks : 9111  
[timing] pages chunking: 0.14s | pdfs chunking: 0.70s | total: 0.85s  
[facts] inserted=52 (from staff_links=52)  
[done] docs=814 chunks=9111 fts_rows=9111 integrity_ok=True
```

Εικόνα 10. Δημιουργία schema της βάσης.

- Υπολογισμός embeddings και αποθήκευση σε ChromaDB: εκτέλεση `build_embeddings_to_chroma.py` (βλ.4.3).



Εικόνα 11. Embeddings σε ChromaDB.

9. Έλεγχος hybrid search: εκτέλεση `hybrid_search.py`, επιστροφή των κορυφαίων αποτελεσμάτων (βλ. 4.4).

```
Backend hybrid search

[11]
✓ 18s
!python /content/Final_Project_ChatBot/ChatBot/hybrid_search.py \
--db /content/Final_Project_ChatBot/ChatBot/uni_ditbot.db \
--chroma_dir /content/Final_Project_ChatBot/ChatBot/chroma_store \
--collection dit_chunks \
--model intfloat/multilingual-e5-large \
--query "Ποιος διδάσκει Τεχνητή Νοημοσύνη;" \
--section_hint lessons \
--k 12 --n_lex 120 --n_sem 120 --w_sem 0.7 --neighbor_radius 1

2025-10-13 07:44:15.733424: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You
2025-10-13 07:44:15.751351: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1760341455.772732 28078 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attemptin
E0000 00:00:1760341455.779235 28078 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attemptin
W0000 00:00:1760341455.796104 28078 computation_placer.cc:177] computation placer already registere
W0000 00:00:1760341455.796135 28078 computation_placer.cc:177] computation placer already registere
W0000 00:00:1760341455.796138 28078 computation_placer.cc:177] computation placer already registere
W0000 00:00:1760341455.796141 28078 computation_placer.cc:177] computation placer already registere
2025-10-13 07:44:15.801082: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow bina
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild Tens
[timing] {'mode': 'hybrid', 'fts': 0.0018622875213623047, 'chroma': 0.413240909576416, 'fuse+expand':
1. ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=139]
2. ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=139]
3. ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=139]
4. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=100]
5. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=100]
6. ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=139]
7. ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=139]
8. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=22]
9. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=22]
10. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=22]
11. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=100]
12. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (lessons/page) [https://www.dit.uoi.gr/lessons/?id=100]
```

Εικόνα 12. Υβριδική ανάκτηση.

10. Εκκίνηση Gradio: εκτέλεση του app_gradio.py (βλ.4.5).

Run The chatbot using Gradio

[12]

```
!python /content/Final_Project_ChatBot/ChatBot/app_gradio.py \
--db /content/Final_Project_ChatBot/ChatBot/uni_ditbot.db \
--chroma_dir /content/Final_Project_ChatBot/ChatBot/chroma_store \
--collection dit_chunks \
--emb_model intfloat/multilingual-e5-large \
--llm_name llsp/Llama-Krikri-8B-Instruct \
--share
```

... model-00001-of-00004.safetensors: 87% 4.38G/4.91G [06:43<00:06, 99.9MB/s]

model-00001-of-00004.safetensors: 89% 4.36G/4.91G [06:44<00:05, 101MB/s]

model-00001-of-00004.safetensors: 90% 4.43G/4.91G [06:45<00:05, 93.5MB/s]

model-00001-of-00004.safetensors: 92% 4.50G/4.91G [06:47<00:07, 53.9MB/s]

model-00002-of-00004.safetensors: 99% 4.85G/4.92G [06:57<00:04, 14.7MB/s]

model-00001-of-00004.safetensors: 92% 4.50G/4.91G [07:07<00:07, 53.9MB/s]

model-00001-of-00004.safetensors: 93% 4.56G/4.91G [07:11<00:41, 8.52MB/s]

model-00001-of-00004.safetensors: 93% 4.56G/4.91G [07:27<00:41, 8.52MB/s]

model-00001-of-00004.safetensors: 94% 4.63G/4.91G [08:47<02:24, 1.96MB/s]

model-00002-of-00004.safetensors: 100% 4.92G/4.92G [08:47<00:00, 9.31MB/s]

model-00001-of-00004.safetensors: 95% 4.65G/4.91G [08:47<02:06, 2.13MB/s]

model-00001-of-00004.safetensors: 96% 4.71G/4.91G [08:48<01:01, 3.29MB/s]

model-00001-of-00004.safetensors: 100% 4.91G/4.91G [08:48<00:00, 9.30MB/s]

Fetching 4 files: 100% 4/4 [08:48<00:00, 132.14s/it]

Loading checkpoint shards: 100% 4/4 [08:13<00:00, 3.26s/it]

generation_config.json: 100% 184/184 [08:00<00:00, 2.04MB/s]

Device set to use cuda:0

/content/Final_Project_ChatBot/ChatBot/app_gradio.py:85: UserWarning: You have not specified a value for chat = gr.Chatbot(label="Συζήτηση", height=480)

* Running on local URL: <http://127.0.0.1:7860>

* Running on public URL: <https://72241b7a77919eba62.gradio.live>

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run "gradio deploy"

Usage rate: approximately 1.71 per hour
You have 1 active session.
[Manage sessions](#)

Python 3 Google Compute Engine backend (GPU)
Showing resources from 09:15 to 10:56

System RAM
4.3 / 53.0 GB

GPU RAM
18.7 / 22.5 GB

Disk
74.5 / 112.6 GB

Εικόνα 13. Εκκίνηση του chatbot.

11. Εξαγωγή διαλόγου σε αρχείο.

Save chat to TXT

Λήψη αρχείου

chat_debug_20251013_084136.txt 8.4 KB ↓

Εικόνα 14. Αποθήκευση διαλόγου και εξαγωγή.

28

Παραδείγματα διαλόγου:

Συζήτηση

Πού και πώς μπορώ να ζητήσω πιστοποιητικό φοίτησης;

Για να ζητήσεις πιστοποιητικό φοίτησης από το Τμήμα Πληροφορικής & Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων, πρέπει να ακολουθήσεις τα εξής βήματα:

1. Σύνδεση στο σύστημα: Επισκέψου την ιστοσελίδα <https://classweb.uoi.gr/> και σύνδεσε με τα ακαδημαϊκά σου στοιχεία (Όνομα Χρήστη + Κωδικό Πρόσβασης).
2. Επιλογή κατηγορίας: Από το προφίλ σου, πήγαινε στην κατηγορία "Πιστοποιητικά".
3. Υποβολή αιτήματος: Επίλεξε "Αίτημα Πιστοποιητικού" και διάλεξε το είδος του πιστοποιητικού που επιθυμείς.
4. Παρακολούθηση: Η αίτησή σου θα εμφανίζεται ως εκκρεμής μέχρι να την επεξεργαστεί η Γραμματεία.
5. Λήψη πιστοποιητικού: Όταν η Γραμματεία ολοκληρώσει την επεξεργασία, θα μπορείς να κατεβάσεις το πιστοποιητικό σου πατώντας "Download".

Πηγές: https://www.dit.uoi.gr/files/odigies_bebaiosi_spoudon_2022.pdf, https://www.dit.uoi.gr/files/%CE%9219.%CE%95%CF%83%CF%89%CF%84%CE%B5%CF%81%CE%B9%CE%BA%CF%8C%CF%82_%CE%9A%CE%B1%CE%BD%CE%BF%CE%BD%CE%B9%CF%83%CE%BC%CF%8C%CF%82_%CE%9B%CE%B5%CE%B9%CF%84%CE%BF%CF%85%CF%81%CE%B3%CE%AF%CE%B1%CF%82_%CE%BD%CE%AD%CE%BF%CF%85_%CE%A0%CE%A0%CE%A3_v2.0.pdf, https://www.dit.uoi.gr/files/kanonismos_katexeresi2024.pdf, https://www.dit.uoi.gr/news.php?sa=view_new&id=8696

Μήνυμα

Γράψε εδώ και πάτα Enter...

Ρυθμίσεις

k12n_lex (FTS)120n_sem (Chroma)120Βάρος semantic0,7Γείτονες ±radius1

42020200102

Section φίλτρο (προαιρετικό)Φίλτρο έτους (π.χ. 2025)

max_new_tokens896temperature0,1

128204801

SendSave chat to TXTΛήψη αρχείου

Εικόνα 15. Ερώτηση 1.

Ποια είναι τα στοιχεία επικοινωνίας και οι ώρες γραφείου του Γιαννακέα;

Ο Γιαννακέας Νικόλαος είναι Αναπληρωτής Καθηγητής στο Τμήμα Πληροφορικής & Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων. Τα στοιχεία επικοινωνίας του είναι:
Email: giannakeas@uoi.gr
Τηλέφωνο: 26810 50519
Ώρες συνεργασίας με φοιτητές: Δευτέρα 12:00-14:00, Παρασκευή 08:00-10:00

Πηγές:
[staff/page] Γιαννακέας Νικόλαος | <https://www.dit.uoi.gr/staff/profile?id=5>

Μήνυμα

Γράψε εδώ και πάτα Enter...

Ρυθμίσεις

k

12

↕

4

20

n_lex (FTS)

120

↕

20

200

n_sem (Chroma)

120

↕

20

200

Βάρος semantic

0,7

↕

0

1

Γείτονες ±radius

1

↕

0

2

Section φίλτρο (προαιρετικό)

Φίλτρο έτους (π.χ. 2025)

max_new_tokens

896

↕

128

2048

temperature

0,1

↕

0

1

Send

Save chat to TXT

Λήψη αρχείου

Εικόνα 16. Ερώτηση 2.

Συζήτηση

Ποιός διδάσκει μαθηματική ανάλυση;

Η διδάσκουσα του μαθήματος Μαθηματική Ανάλυση είναι η Γεωργία Φουτσιτζή. Πηγές: [announcements/page]

Μήνυμα

Γράψε εδώ και πάτα Enter...

Ρυθμίσεις

k

12

⌵

n_lex (FTS)

120

⌵

n_sem (Chroma)

120

⌵

Βάρος semantic

0,7

⌵

Γείτονες ±radius

1

⌵

4

20

20

200

20

200

0

1

0

2

Section φίλτρο (προαιρετικό)

Φίλτρο έτους (π.χ. 2025)

max_new_tokens

768

⌵

temperature

0,1

⌵

128

2048

0

1

Send

Save chat to TXT

Λήψη αρχείου

Εικόνα 18. Ερώτηση 4.

Συζήτηση

Πώς κάνω δηλώσεις μαθημάτων

Για να κάνετε δηλώσεις μαθημάτων, πρέπει να ακολουθήσετε τα εξής βήματα:

1. Συνδεθείτε στην πλατφόρμα του Φοιτητολογίου: <https://classweb.uoi.gr> χρησιμοποιώντας τον ιδρυματικό σας λογαριασμό (Username, password).
2. Η περίοδος δηλώσεων είναι συγκεκριμένη και ανακοινώνεται από το Τμήμα. Για παράδειγμα, για το εαρινό εξάμηνο 2024-2025, η δήλωση έγινε από 17/3/25 έως 23/3/25 (πηγή: <http://www.dit.uoi.gr/files/pdf17325.pdf>).
3. Δηλώστε τα μαθήματα που επιθυμείτε να παρακολουθήσετε και να εξεταστείτε στο τέλος του εξαμήνου. Για το εαρινό εξάμηνο, μπορείτε να δηλώσετε όσα μαθήματα θέλετε των εαρινών εξαμήνων (πηγή: <http://www.dit.uoi.gr/files/pdf17325.pdf>).
4. Ιδιαίτερη προσοχή απαιτείται στην τήρηση των προϋποθέσεων δήλωσης μαθημάτων:
 - ο Δήλωση όλων των προαπαιτούμενων μαθημάτων (προαπαιτούμενα)
 - ο Μη δήλωση μαθημάτων μεγαλύτερου εξαμήνου από το τρέχον (εκτός ειδικών περιπτώσεων)
 - ο Τήρηση του ορίου ECTS ανά εξάμηνο (30 για 1ο-2ο, 42 για 3ο-6ο, 48 για 7ο και άνω)
 - ο Για πρακτική άσκηση ή πτυχιακή εργασία, πρέπει να πληρούνται οι αντίστοιχες προϋποθέσεις και να δηλωθούν ως μαθήματα (πηγές: <http://www.dit.uoi.gr/files/pdf173252.pdf>, https://www.dit.uoi.gr/files/v15_dit_os.pdf)
5. Σε περίπτωση προβλημάτων, επικοινωνήστε με τη Γραμματεία στο grampt@uoi.gr (πηγή: <http://www.dit.uoi.gr/files/pdf173252.pdf>).

Μήνυμα

Γράψτε εδώ και πάτα Enter...

Ρυθμίσεις

k

12

↕

n_lex (FTS)

120

↕

n_sem (Chroma)

120

↕

Βάρος semantic

0,7

↕

Γείτονες ±radius

1

↕

4

20

20

200

20

200

0

1

0

2

Section φίλτρο (προαιρετικό)

Φίλτρο έτους (π.χ. 2025)

max_new_tokens

768

↕

temperature

0,1

↕

128

2048

0

1

Send

Save chat to TXT

Λήψη αρχείου

Εικόνα 19. Ερώτηση 5.

Σύμφωνα με τα παραπάνω παραδείγματα, τεκμηριώνεται πρακτικά το pipeline: ομοιόμορφη συλλογή και καθαρισμός εγγράφων (βλ. 4.1 – 4.2), ενοποιημένη βάση και υπολογισμός embeddings (βλ.4.3), υβριδική ανάκτηση με RRF και φίλτρα (βλ.4.4) και παρουσίαση απαντήσεων στο Gradio με παραπομπές και δυνατότητα εξαγωγής.

Κεφάλαιο 5 – Συμπεράσματα και προτάσεις βελτίωσης

5.1: Περιορισμοί και βελτιώσεις

Η λύση βασίστηκε εξ' ολοκλήρου σε δωρεάν εργαλεία-μοντέλα και υποδομές (με εξαίρεση το google colabs), κάτι που έθεσε φυσικά όρια σε latency, context window και γλωσσική ρευστότητα. Υπάρχουν εμπορικά LLMs και επαγγελματικού rerankers/vector stores που πιθανότατα θα βελτιώναν την ακρίβεια και ταχύτητα (Vellum.ai, n.d.). Τα PDFs χαμηλής ποιότητας παραμένουν μια πρόκληση, ωστόσο το OCR μοντέλο βελτιώνει, αλλά δεν λύνει πλήρως τα προβλήματα. Επιπλέον, το chunking επηρεάζει άμεσα την ποιότητα διότι μικρά

chunks κόβουν τα συμφραζόμενα και τα μεγάλα “φουσκώνουν” το prompt. Παρά τους περιορισμούς, η υλοποίηση δείχνει ότι με προσεκτική μηχανική RAG μπορείς να πετύχεις χρήσιμες απαντήσεις σε δωρεάν περιβάλλον.

5.2: Μελλοντικές ιδέες

Πιθανές κατευθύνσεις θα μπορούσε να ήταν η συνεχή ενημέρωση των δεδομένων και ανίχνευση αλλαγών, ώστε να εισάγονται μόνο νεότερα ή τροποποιημένα στοιχεία. Θα μπορούσε επίσης να προστεθεί ενσωμάτωση των email της γραμματεία, ώστε να καλύπτονται πιο έγκαιρα οι πληροφορίες. Η αξιολόγηση με feedback χρηστών θα μπορούσε να τροφοδοτεί αυτόματη βελτίωση παραμέτρων και prompts. Τέλος, το fine-tuning θα μπορούσε να εξελιχθεί από παραδείγματα σε LoRa/QLoRa με συστηματική μέτρηση ακρίβειας και καθυστέρησης.

5.3 Συμπεράσματα

Εν κατακλείδι, το έργο έδειξε ότι ένα προσεκτικά σχεδιασμένο RAG μπορεί να μετατρέψει έναν χαοτικό ιστότοπο σε αξιόπιστες απαντήσεις. Το όφελος δεν είναι μόνο τεχνικό, αλλά φοιτητές και προσωπικό θα μπορούν να βρίσκουν γρηγορότερα τη σωστή πληροφορία, με σαφείς παραπομπές. Υπάρχουν βελτιώσεις να γίνουν, ιδίως σε OCR, chunking και αξιολόγηση, αλλά η κατεύθυνση είναι ξεκάθαρη.

Βιβλιογραφία

Baldi, P., & Sadowski, P. (2016). *A Theory of Local Learning, the Learning Channel, and the Optimality of Backpropagation*. <https://doi.org/10.1016/j.neunet.2016.07.006>

ChromaDB. (n.d.-a). *ChromaDB*. Retrieved October 9, 2025, from <https://docs.trychroma.com/docs/embeddings/embedding-functions>

ChromaDB. (n.d.-b). *Persistent Client*. Retrieved October 9, 2025, from <https://docs.trychroma.com/docs/run-chroma/persistent-client>

- Cohere. (n.d.). *Dense Retrieval*. Retrieved October 9, 2025, from <https://cohere.com/llmu/dense-retrieval>
- Dimitris Roussis and Leon Voukoutis and Georgios Paraskevopoulos and Sokratis Sofianopoulos and Prokopis Prokopidis and Vassilis Papavasileiou and Athanasios Katsamanis and Stelios Piperidis and Vassilis Katsouros. (n.d.). *Llama-Krikri-8B-Instruct*. Retrieved October 9, 2025, from <https://huggingface.co/ilsp/Llama-Krikri-8B-Instruct>
- Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018). AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. *Minds and Machines*, 28(4), 689–707. <https://doi.org/10.1007/s11023-018-9482-5>
- GeeksForGeeks. (n.d.). *Retrieval-Augmented Generation*. Retrieved October 9, 2025, from <https://www.geeksforgeeks.org/nlp/what-is-retrieval-augmented-generation-rag/>
- GeeksForGeeks_yaml. (n.d.). *How Can YAML Be Utilized In Ansible Playbooks For Automation And Orchestration ?* Retrieved October 9, 2025, from <https://www.geeksforgeeks.org/devops/how-can-yaml-be-utilized-in-ansible-playbooks-for-automation-and-orchestration/>
- Gradio. (n.d.). *Gradio App*. Retrieved October 9, 2025, from <https://www.gradio.app/docs/gradio/interface>
- huggingface. (n.d.). *Multilingual-E5-large*. Retrieved October 9, 2025, from <https://huggingface.co/intfloat/multilingual-e5-large?>
- Joseph Weizenbaum. (1966). *ELIZA—a computer program for the study of natural language communication between man and machine*. <https://dl.acm.org/doi/10.1145/365153.365168>
- Jurafsky, D., & Martin, J. H. (n.d.). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models Third Edition draft*.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (n.d.). *The Stanford CoreNLP Natural Language Processing Toolkit*.
- Microsoft, chunking. (n.d.). *Chunk large documents for vector search solutions in Azure AI Search*. Retrieved October 9, 2025, from <https://learn.microsoft.com/en-us/azure/search/vector-search-how-to-chunk-documents>

- Microsoft, R. (n.d.). *Relevance scoring in hybrid search using Reciprocal Rank Fusion (RRF)*. Retrieved October 9, 2025, from <https://learn.microsoft.com/en-us/azure/search/hybrid-search-ranking>
- Microsoft_Hybrid_Search. (n.d.). *Hybrid search using vectors and full text in Azure AI Search*. Retrieved October 9, 2025, from <https://learn.microsoft.com/en-us/azure/search/hybrid-search-overview>
- Moors, K., Sorée, B., & Magnus, W. (2015). *Modeling surface roughness scattering in metallic nanowires*. <https://doi.org/10.1063/1.4931573>
- Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. In *Computers and Education: Artificial Intelligence* (Vol. 2). Elsevier B.V. <https://doi.org/10.1016/j.caeai.2021.100033>
- Souvik Mandal and Ashish Talewar and Paras Ahuja and Prathamesh Juvatkar. (n.d.). *Nanonets-OCR-s*. Retrieved October 9, 2025, from <https://huggingface.co/nanonets/Nanonets-OCR-s>
- SQLite. (n.d.). *SQLite FTS5 Extension*. Retrieved October 9, 2025, from https://www.sqlite.org/fts5.html#the_bm25_function
- Stuart J. Russel, & Peter Norvig. (2022). *Artificial Intelligence A Modern Approach Fourth Edition*. <http://aima.cs.berkeley.edu/>
- Vaidyam, A. N., Wisniewski, H., Halamka, J. D., Kashavan, M. S., & Torous, J. B. (2019). Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape. In *Canadian Journal of Psychiatry* (Vol. 64, Issue 7, pp. 456–464). SAGE Publications Inc. <https://doi.org/10.1177/0706743719828977>
- Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023). *Attention Is All You Need*.
- Vellum.ai. (n.d.). *LLM Ranking*. Retrieved October 9, 2025, from https://www.vellum.ai/llm-leaderboard?utm_source=google&utm_medium=organic
- Watson, J., Goel, A. K., & Polepeddi, L. (n.d.). *A Virtual Teaching Assistant for Online Education*. <https://www.class-central.com/report/mooc-stats-2016/>
- Winkler, R., & Soellner, M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. *Academy of Management Proceedings*, 2018(1), 15903. <https://doi.org/10.5465/ambpp.2018.15903abstract>

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? In *International Journal of Educational Technology in Higher Education* (Vol. 16, Issue 1). Springer Netherlands. <https://doi.org/10.1186/s41239-019-0171-0>