

Ατομική Διπλωματική Εργασία

**ΕΚΤΕΛΕΣΗ ΕΡΩΤΗΣΕΩΝ ΣΕ ΜΕΓΑΛΑ ΓΛΩΣΣΙΚΑ ΜΟΝΤΕΛΑ (LLMs)
ΜΕ ΜΗΧΑΝΙΣΜΟΥΣ ΠΡΟΣΤΑΣΙΑΣ ΤΗΣ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΧΡΗΣΤΩΝ**

Δήμητρα Παφίτου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εκτέλεση Ερωτήσεων σε Μεγάλα Γλωσσικά Μοντέλα (LLMs) με Μηχανισμούς Προστασίας της Ιδιωτικότητας Χρηστών

Δήμητρα Παφίτου

Επιβλέπων Καθηγητής

Δρ. Δημήτρης Ζεϊναλιπούρ

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2025

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένεια μου για την συνεχή υποστήριξη που μου παρείχαν καθ' όλη την διάρκεια σπουδών μου. Καθημερινά με ωθούσαν να συνεχίζω και να πιέζω τον εαυτό μου ώστε να πετυχαίνω τον κάθε μου στόχο.

Ακόμη, θέλω να ευχαριστήσω τους φίλους μου και τους συμφοιτητές μου οι οποίοι έκαναν τα φοιτητικά μου χρόνια καλύτερα και αξέχαστα.

Επίσης, θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Δημήτρη Ζεϊναλιπούρ για την εμπιστοσύνη του, και την αδιάκοπη βοήθεια που μου πρόσφερε τον τελευταίο χρόνο για την ολοκλήρωση της ατομικής διπλωματικής μου εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω τον συμφοιτητή μου Κωνσταντίν Κρασοβίτσκι ο οποίος μέσω της συνεργασίας μας στην διπλωματική εργασία μου με βοήθησε να ξεπεράσω δυσκολίες στην υλοποίηση.

Περίληψη

Η σημερινή εποχή χαρακτηρίζεται από την χρήση και την ανάπτυξη της Τεχνητής Νοημοσύνης και συγκεκριμένα της Παραγωγικής Τεχνητής Νοημοσύνης (Generative Artificial Intelligence). Με την ραγδαία αυτή ανάπτυξη οι άνθρωποι αναζητούν τρόπους να προσαρμόσουν την Τεχνητή Νοημοσύνη στα μέτρα του. Ο κύριος τρόπος με τον οποίο ένα Γλωσσικό Μοντέλο μπορεί να εξειδικευτεί στα δεδομένα του κάθε ατόμου είναι η βελτιστοποίησή (fine tuning) του, ωστόσο, η μέθοδος αυτή δεν είναι ιδανική για γρήγορη εκπαίδευση ενός μοντέλου από απλούς χρήστες. Έτσι, βρέθηκε μια νέα μέθοδος, η Retrieval Augmented Generation (RAG) η οποία σε πραγματικό χρόνο διαβάζει τα δεδομένα του χρήστη και τα χρησιμοποιεί για να παραγάγει απαντήσεις που τα αφορούν. Παράλληλα, με την αδιαμφισβήτητη βοήθεια που προσφέρει στους ανθρώπους η Τεχνητή Νοημοσύνη, διατυπώνονται και ανησυχίες σχετικά με την χρήση της και τις επιπτώσεις που επιφέρει ως προς την έκθεση των προσωπικών δεδομένων.

Αυτή η διπλωματική εργασία στοχεύει στην υλοποίηση ενός συστήματος το οποίο ενώ αξιοποιεί Μεγάλα Γλωσσικά Μοντέλα, διατηρεί την προστασία των χρηστών και των δεδομένων τους με μεθόδους ανωνυμίας. Το σύστημα PrivateLLM με την χρήση βάσεων οι οποίες αποθηκεύουν δεδομένα στον ιδιωτικό φυλλομετρητή ιστού (web browser) του χρήστη, εκτελεί την μέθοδο RAG και στέλνει δεδομένα σε Μεγάλα Γλωσσικά Μοντέλα ώστε ο χρήστης να μπορεί να αλληλοεπιδράσει με δικά του έγγραφα. Κατά τη διάρκεια της διαδικασίας αυτής, τα δεδομένα του χρήστη είναι προστατευμένα τόσο λόγω της τοπικής αποθήκευσής τους, η οποία δεν φεύγει από τον ιδιωτικό φυλλομετρητή ιστού του χρήστη, όσο και με την δυνατότητα παραπλάνησης του γλωσσικού μοντέλου. Τέλος, πραγματοποιήθηκε αξιολόγηση του συστήματος, σε μετρικές χρόνου, ακρίβειας, και σύγκρισης του με άλλα παρόμοια συστήματα.

Περιεχόμενα

Ευχαριστίες	iii
Περίληψη	iv
Περιεχόμενα	v
Κεφάλαιο 1	1
Εισαγωγή.....	1
1.1 Κίνητρο.....	1
1.2 Πρόβλημα.....	4
1.3 Συνεισφορά.....	5
1.4 Συνοπτική Ανασκόπηση Εργασίας	6
Κεφάλαιο 2.....	7
Επιστημονικό Υπόβαθρο και Σχετική Εργασία	7
2.1 Επιστημονικό Υπόβαθρο	7
2.1.1 Τεχνητή Νοημοσύνη	7
2.1.2 Μεγάλα Γλωσσικά Μοντέλα (LLMs)	8
2.1.3 Μεγάλα Γλωσσικά Μοντέλα Ανοικτού Κώδικα	9
2.1.4 Εκπαίδευση και Βελτιστοποίηση.....	10
2.1.5 Retrieval – Augmented Generation (RAG)	12
2.1.6 Βελτιστοποίηση vs. RAG.....	12
2.1.7 Διανυσματικές Βάσεις.....	14
2.1.8 IndexedDB	16
2.2 Σχετική Εργασία	18
2.2.1 Βιομηχανικές Εφαρμογές.....	18
2.2.2 Ακαδημαϊκές Εργασίες	21
2.2.2.1 SaulLM-7B.....	21
2.2.2.2 DISC-LawLLM	22
2.2.2.3 Fine Tuning Llama 3	25
2.2.2.4 Document Drafting with LLM	26
2.2.2.5 Law-Neo	27
Κεφάλαιο 3.....	29
Μηχανισμοί για Συμπερίληψη Προσωπικών Δεδομένων σε LLMs.....	29
3.1 Εκπαίδευση με Γεωγραφικά Δεδομένα	29
3.2 Βήματα για Βελτιστοποίηση.....	32
3.3 Αποθήκευση στη ChromaDB.....	38
3.4 Ανάκτηση Πληροφορίας με RAG	45

3.5 Αξιολόγηση Μεθόδου RAG.....	48
3.6 Προβλήματα με RAG.....	57
Κεφάλαιο 4.....	59
Αρχιτεκτονική Συστήματος	59
4.1 Επισκόπηση Αρχιτεκτονικής	59
4.2 Αποθήκευση & Επεξεργασία Εγγράφων	61
4.2.1 Φόρτωση Αρχείων.....	61
4.2.2 Αποθήκευση Διανυσμάτων	64
4.2.3 Προβολή Εγγράφων.....	66
4.2.4 Διαχείριση Εγγράφων.....	69
4.3 Ανάλυση & Απάντηση Ερωτήσεων	70
4.3.1 Εύρεση Συμφραζόμενων.....	70
4.3.2 Επικοινωνία με Γλωσσικά Μοντέλα.....	74
4.3.3 Επεξεργασία Ερωτήματος.....	80
4.3.4 Μέθοδοι Προστασίας Ιδιωτικότητας & Υλοποίηση Κ-Ανωνυμίας.....	87
4.4 Διαχείριση Συνεδριών Χρήστη.....	93
Κεφάλαιο 5.....	97
Αξιολόγηση, Συγκρίσεις και Προοπτικές Βελτίωσης.....	97
5.1 Αξιολόγηση	97
5.1.1 Απόδοση Χρόνου	97
5.1.2 Ακρίβεια Απαντήσεων	100
5.1.3 Επιρροή Κ-Ανωνυμίας στην Ακρίβεια	104
5.2 Συγκρίσεις με Παρόμοια Συστήματα	108
5.3 Περιορισμοί.....	109
5.4 Μελλοντικές Εργασίες	110
Κεφάλαιο 6.....	112
Συμπεράσματα.....	112
6.1 Τελικά Συμπεράσματα	112
Βιβλιογραφία	113

Κεφάλαιο 1

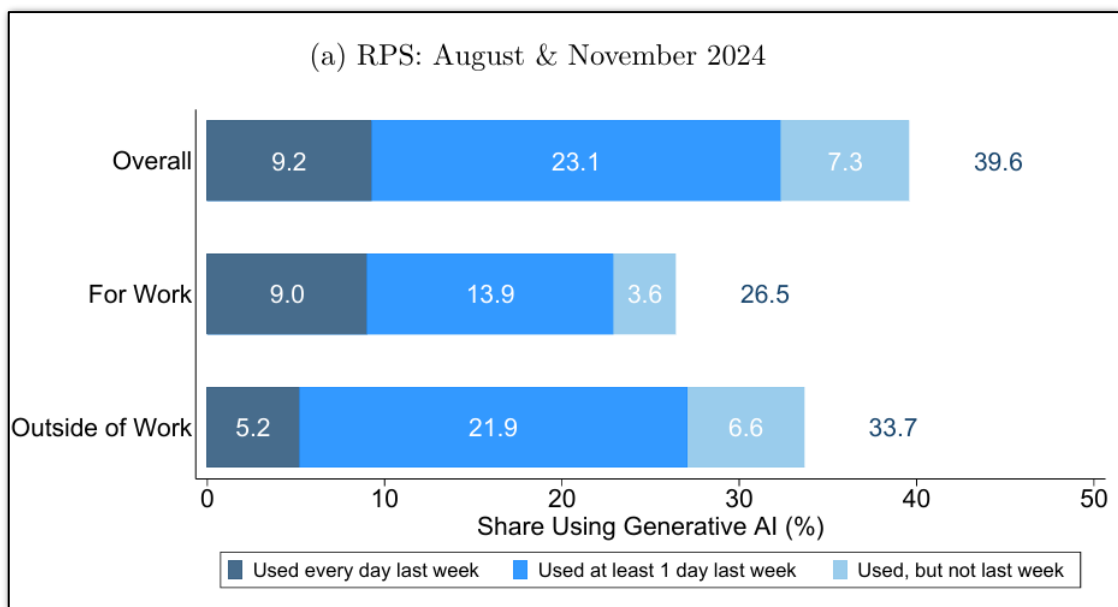
Εισαγωγή

<i>1.1 Κίνητρο.....</i>	<i>1</i>
<i>1.2 Πρόβλημα.....</i>	<i>4</i>
<i>1.3 Συνεισφορά.....</i>	<i>5</i>
<i>1.4 Συνοπτική Ανασκόπηση Εργασίας</i>	<i>6</i>

1.1 Κίνητρο

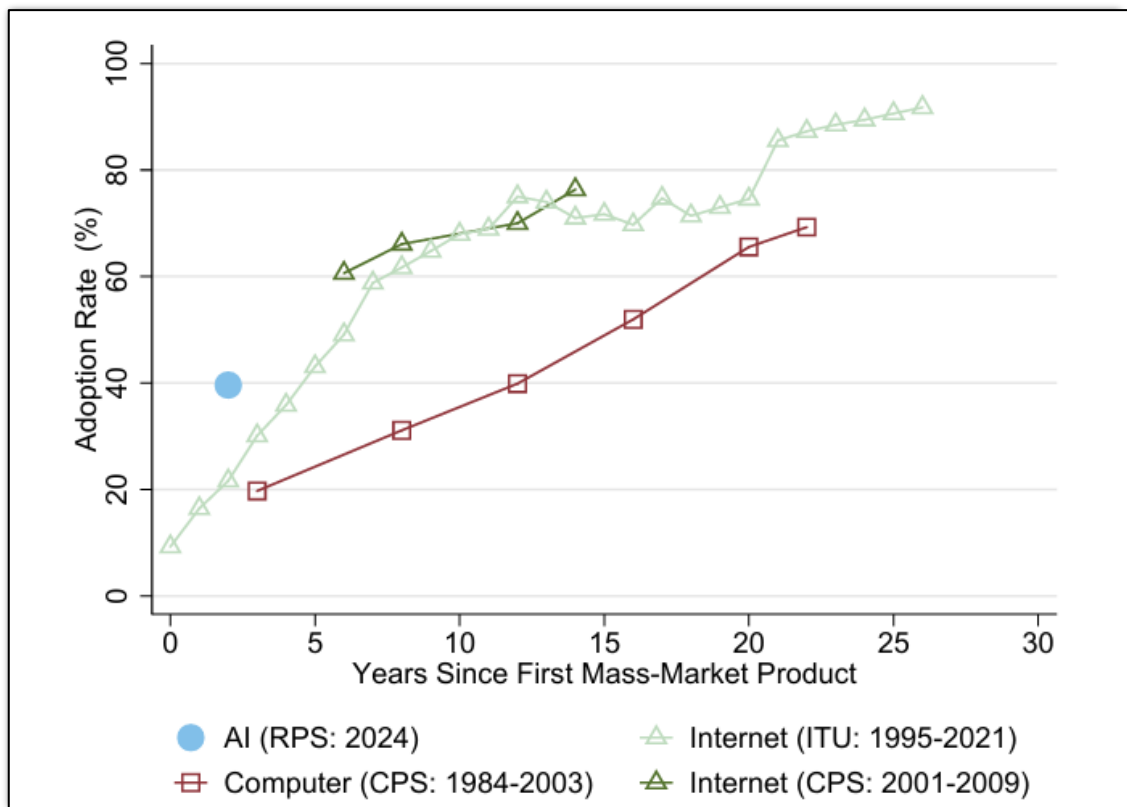
Τα τελευταία χρόνια η παγκόσμια τάση για την χρήση της Τεχνητής Νοημοσύνης έχει αυξηθεί ραγδαία. Η ραγδαία αυτή χρήση της Τεχνητής Νοημοσύνης και συγκεκριμένα της Παραγωγικής Τεχνητής Νοημοσύνης (Generative Artificial Intelligence) άρχισε με την εμφάνιση του Μεγάλου Γλωσσικού Μοντέλου ChatGPT τον Νοέμβριο του 2022.

Οι Alexander Bick , Adam Blandin, και David J. Deming σε ένα άρθρο τους συνδύασαν δεδομένα από διάφορες έρευνες και εξήγαγαν αποτελέσματα σχετικά με την χρήση του Generative AI από τον γενικό πληθυσμό [2]. Οι έρευνες έδειξαν ότι η χρήση του Generative AI πραγματοποιείται τόσο στα πλαίσια εργασίας όσο και στην καθημερινότητα του πληθυσμού (βλ. Σχήμα 1.1). Το συμπέρασμα ήταν ότι 39.6% του πληθυσμού ηλικιών 18-64 δηλώνουν ότι χρησιμοποιούν Generative AI στην καθημερινότητα τους [2].



Σχήμα 1.1: Ποσοστό Ενηλίκων που Χρησιμοποιούν Παραγωγική Τεχνητή Νοημοσύνη [2]

Στην ίδια έρευνα [2], αναδείχτηκε επίσης ότι ο πληθυσμός προσαρμόστηκε πιο γρήγορα στην χρήση της Παραγωγικής Τεχνητής Νοημοσύνης από οποιαδήποτε άλλη τεχνολογία του παρελθόντος. Δηλαδή, ο γενικός πληθυσμός δέχτηκε με παραπάνω ευκολία την άφιξη του GenAI από την άφιξη των ηλεκτρονικών υπολογιστών και του διαδικτύου (βλ. Σχήμα 1.2).



Σχήμα 1.2: "Εξελικτική Πορεία Υιοθέτησης Υπολογιστών, Διαδικτύου και Παραγωγικής Τεχνητής Νοημοσύνης [2]

Πέρα από τον γενικό πληθυσμό, παρατηρήθηκε επίσης ραγδαία αύξηση στις επιστημονικές έρευνες και μελέτες οι οποίες αφορούν Μεγάλα Γλωσσικά Μοντέλα. Συγκεκριμένα, μια έρευνα που διεξάχθηκε από τους Zhiqiu Xia, Lang Zhu, και Bingzhe Li, αναλύθηκαν 168,331 επιστημονικές μελέτες μεταξύ των χρονιών 2019 και 2024 [69]. Από αυτές τις μελέτες, εντόπισαν για κάθε χρονιά πόσες αφορούσαν την χρήση Μεγάλων Γλωσσικών Μοντέλων. Τα αποτελέσματα που εξάχθηκαν φανερώνουν την αυξημένη τάση σε μελέτες που αφορούν τα Large Language Models (LLMs) (βλ. Πίνακα 1.1).

Year	2019	2020	2021	2022	2023	2024
Number	503	1105	1595	2027	3854	7109

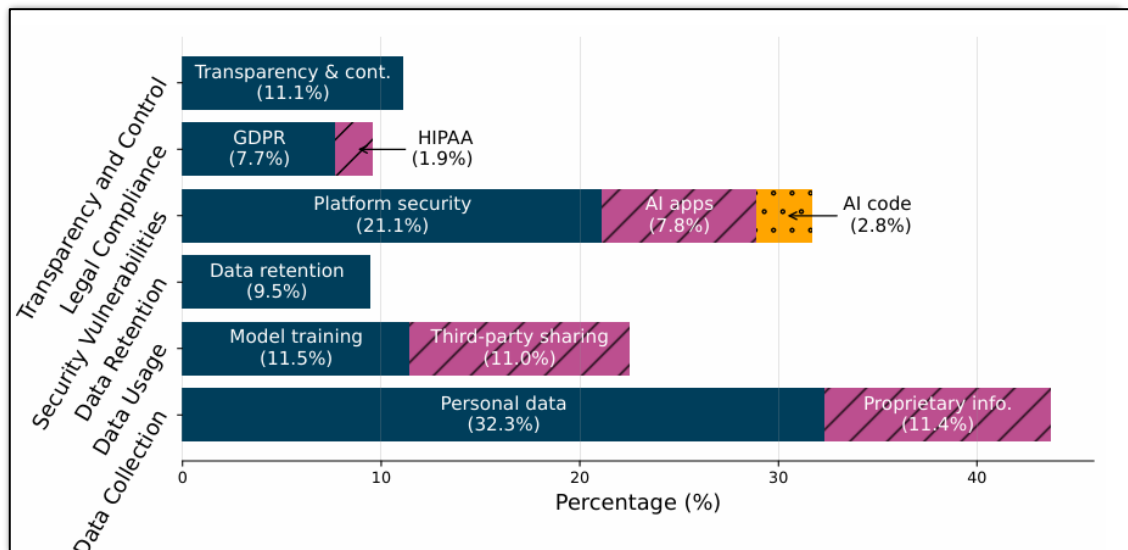
Πίνακας 1.1: Αριθμοί Μελετών που Αφορούν Μεγάλα Γλωσσικά Μοντέλα ανά Χρονιά [69]

Συμπερασματικά, το κίνητρο για την διεξαγωγή της συγκεκριμένης διπλωματικής εργασίας είναι η ραγδαία ανάπτυξη της τάσης για χρήση της Παραγωγικής Τεχνητής Νοημοσύνης, τόσο από τον γενικό πληθυσμό όσο και από επιστήμονες.

1.2 Πρόβλημα

Με την ραγδαία ανάπτυξη της χρήσης της Τεχνητής Νοημοσύνης, ακόλουθο είναι και η δημιουργία προβληματισμών που την αφορούν. Χαρακτηριστικό παράδειγμα το οποίο επηρέασε τους χρήστες και αύξησε τις ανησυχίες για τα δεδομένα τους αποτελεί το περιστατικό τον Ιούνιο του 2023 όπου δεδομένα από εκατό χιλιάδες λογαριασμούς χρηστών του ChatGPT βρέθηκαν να πωλούνται στο σκοτεινό διαδίκτυο (dark web) [64].

Επιπρόσθετα, η δυνατότητα των Μεγάλων Γλωσσικών Μοντέλων να εκπαιδεύονται από μόνα τους δίνει την ευκαιρία να αποθηκεύουν στην μνήμη τους δεδομένα από τους χρήστες, τα οποία μπορούν στην συνέχεια να εμφανίσουν ως αποτέλεσμα στην απάντηση ερωτήσεων άλλων χρηστών. Πέρα από αυτού υπάρχουν πολλές ακόμη ανησυχίες από τους χρήστες όσον αφορά την ασφάλεια στη χρήση Μεγάλων Γλωσσικών Μοντέλων (βλ. Σχήμα 1.3) [25].



Σχήμα 1.3: Επικράτηση Ανησυχιών των Χρηστών Σχετικά με την Ασφάλεια και την Ιδιωτικότητα [25]

1.3 Συνεισφορά

Η παρούσα διπλωματική εργασία παρουσιάζει τρόπους με τους οποίους η χρήση Μεγάλων Γλωσσικών Μοντέλων και Παραγωγικής Τεχνητής Νοημοσύνης στοχεύει στο να γίνει πιο ασφαλής προς τον χρήστη και τα προσωπικά του δεδομένα. Η συνεισφορά της διπλωματικής εργασίας και του συστήματος που υλοποιήθηκε εντοπίζεται τόσο στην καλύτερη διαχείριση και διαφάνεια προς τα δεδομένα που χρησιμοποιούνται κατά την επικοινωνία με τα γλωσσικά μοντέλα, όσο και στην καλύτερη ασφάλεια των δεδομένων του χρήστη. Ουσιαστικά στοχεύει στην προστασία των δεδομένων του χρήστη από τα Μεγάλα Γλωσσικά Μοντέλα.

Το σύστημα αποτελεί μια ιστοσελίδα στην οποία οι χρήστες μπορούν να ανεβάσουν έγγραφα και να επικοινωνήσουν με Μεγάλα Γλωσσικά Μοντέλα. Τα έγγραφα που ανεβάζουν οι χρήστες χρησιμοποιούνται ως συμφραζόμενα για τον εμπλουτισμό της απάντησης του μοντέλου με βάση αυτά. Επιπλέον, ο χρήστης μπορεί να διαχειριστεί τα έγγραφα του, καθώς και τα δεδομένα που εξάγονται από αυτά. Ακόμη, δίνεται η δυνατότητα στον χρήστη να επιλέξει με ποιο γλωσσικό μοντέλο θα επικοινωνήσει.

Συγκεκριμένα, το σύστημα λειτουργεί εξ ολοκλήρου στην πλευρά του χρήστη, δηλαδή ο κώδικας τρέχει μόνο στον ιδιωτικό φυλλομετρητή του χρήστη, χωρίς την ανάγκη για εξωτερικό διακομιστή. Όσον αφορά τα δεδομένα τα οποία ανεβάζει ο χρήστης, αποθηκεύονται στην τοπική βάση IndexedDB, η οποία λειτουργεί στον φυλλομετρητή ιστού του χρήστη [29]. Επιπρόσθετα, το σύστημα χρησιμοποιεί τεχνικές ανωνυμίας, όπως η κ-ανωνυμία, για την προστασία τόσο της ερώτησης του χρήστη, όσο και των δεδομένων που αποστέλλονται στο γλωσσικό μοντέλο.

Με αυτό τον τρόπο, το σύστημα συμβάλλει στην χρήση της Παραγωγικής Τεχνητής Νοημοσύνης, ενώ ταυτόχρονα προσφέρει τρόπους προστασίας των δεδομένων των χρηστών.

1.4 Συνοπτική Ανασκόπηση Εργασίας

Η παρούσα ατομική διπλωματική εργασία αποτελείται από 6 κεφάλαια.

Στο Κεφάλαιο 1 παρουσιάζεται το κίνητρο και το πρόβλημα τα οποία οδήγησαν στην μελέτη και την υλοποίηση του συστήματος. Επίσης, παρουσιάζεται μια σύντομη επισκόπηση της εργασίας.

Ακολούθως, στο Κεφάλαιο 2 αναλύονται έννοιες οι οποίες χρησιμοποιούνται ως υπόβαθρο για την διπλωματική εργασία. Ακόμη, παρουσιάζονται παρόμοια συστήματα στον χώρο, αλλά και αντίστοιχες επιστημονικές μελέτες και συστήματα.

Στη συνέχεια, στο Κεφάλαιο 3 εξηγούνται τα αρχικά βήματα και οι πρώτες μέθοδοι με τις οποίες έγιναν προσπάθειες για την αντιμετώπιση του προβλήματος το οποίο αναλύθηκε πιο πάνω.

Στο Κεφάλαιο 4 γίνεται λεπτομερής παρουσίαση της αρχιτεκτονικής του συστήματος. Σε αυτό το κεφάλαιο επεξηγούνται οι βασικές λειτουργίες και ο τρόπος με τον οποίο υλοποιήθηκαν.

Έπειτα, στο Κεφάλαιο 5 πραγματοποιείται αξιολόγηση του συστήματος με βάση την απόδοση του και την ακρίβεια των απαντήσεων του. Επίσης, συγκρίνεται με άλλα παρόμοια συστήματα.

Τέλος, στο Κεφάλαιο 6 συνοψίζονται τα συμπεράσματα της εργασίας.

Κεφάλαιο 2

Επιστημονικό Υπόβαθρο και Σχετική Εργασία

2.1 Επιστημονικό Υπόβαθρο	7
2.1.1 Τεχνητή Νοημοσύνη	7
2.1.2 Μεγάλα Γλωσσικά Μοντέλα (LLMs)	8
2.1.3 Μεγάλα Γλωσσικά Μοντέλα Ανοικτού Κώδικα	9
2.1.4 Εκπαίδευση και Βελτιστοποίηση	10
2.1.5 Retrieval – Augmented Generation (RAG)	12
2.1.6 Βελτιστοποίηση vs. RAG	12
2.1.7 Διανυσματικές Βάσεις	14
2.1.8 IndexedDB	16
2.2 Σχετική Εργασία	18
2.2.1 Βιομηχανικές Εφαρμογές	18
2.2.2 Ακαδημαϊκές Εργασίες	21
2.2.2.1 SaulLM-7B.....	21
2.2.2.2 DISC-LawLLM.....	22
2.2.2.3 Fine Tuning Llama 3.....	25
2.2.2.4 Document Drafting with LLM	26
2.2.2.5 Law-Neo.....	27

2.1 Επιστημονικό Υπόβαθρο

2.1.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη είναι μια έννοια που υπήρχε από την δεκαετία του '50 με τον John McCarthy να είναι ο πρώτος που θα αναφερθεί σε τεχνητή νοημοσύνη με το όνομα που έχει σήμερα [6]. Η τεχνητή νοημοσύνη αφορά κυρίως τη δημιουργία προγραμμάτων τα οποία παρουσιάζουν χαρακτηριστικά μιας «έξυπνης» συμπεριφοράς [36]. Παραδείγματα μιας τέτοιας συμπεριφοράς ενός προγράμματος είναι η ικανότητα να παίζει επιτραπέζια παιχνίδια, να αναγνωρίσει αντικείμενα σε εικόνες, αλλά και να βρίσκει μοτίβα σε κείμενα. Ακόμη μια έννοια που είναι άμεσα συνδεδεμένη με την τεχνητή νοημοσύνη είναι αυτή της μηχανικής μάθησης. Η μηχανική μάθηση είναι ο επικρατέστερος τρόπος ενίσχυσης της τεχνητής νοημοσύνης. Συγκεκριμένα, είναι η ικανότητα ενός

προγράμματος να ενισχύει τις γνώσεις του μέσω επηρείας με την χρήση αλγορίθμων που βοηθούν στην επίτευξη του στόχου αυτού.

Σήμερα, η τεχνητή νοημοσύνη είναι παγκοσμίως γνωστή για τη χρήση της σε chatbots, όπως το ChatGPT. Τέτοια μοντέλα τεχνητής νοημοσύνης είναι ικανά να αναγνωρίζουν κείμενο και να παραγάγουν απάντηση. Συγκεκριμένα, ο χρήστης μπορεί να ρωτήσει ή να πει ό,τι θέλει σε ένα τέτοιο μοντέλο και το μοντέλο είναι ικανό να απαντήσει πίσω. Η ικανότητα του να απαντά πίσω προέρχεται από τους αλγόριθμους μηχανικής μάθησης που αναφέρονται πιο πάνω.

Σε γενικές γραμμές ένας αλγόριθμος μηχανικής μάθησης είναι η διαδικασία αναγνώρισης μοτίβων σε κείμενο. Δηλαδή, ένα μοντέλο τρέχει τον συγκεκριμένο αλγόριθμο και έχει ως αποτέλεσμα να μπορεί να προβλέψει την επόμενη λέξη. Ουσιαστικά, η απάντηση που παραγάγει το chatbot είναι αποτέλεσμα συρραφής πολλών κειμένων τα οποία «διάβασε» το μοντέλο και η δημιουργία μιας απάντησης με βάση αυτά. Όσο παραπάνω διάλογος γίνεται με ένα τέτοιο μοντέλο, τόση παραπάνω εμπειρία αποκτά και άρα οι απαντήσεις του γίνονται πιο ακριβείς και κοντά στην πραγματικότητα.

2.1.2 Μεγάλα Γλωσσικά Μοντέλα (LLMs)

Ένα chatbot, δηλαδή τεχνητή νοημοσύνη εστιασμένη στον διάλογο και την παραγωγή λόγου, χαρακτηρίζεται επίσης και ως ένα Μεγάλο Γλωσσικό Μοντέλο (Large Language Model - LLM). Ένα τέτοιο μοντέλο είναι ικανό να αναγνωρίζει και να επεξεργάζεται φυσική γλώσσα, δηλαδή γλώσσα που γράφουμε και μιλούμε στην καθημερινότητά μας. Αρχικά, τα large language models ονομάζονταν pre-trained language models. Αυτά τα μοντέλα είχαν μοναδικό σκοπό να μπορούν να αναγνωρίζουν φυσική γλώσσα. Έπειτα όσο πιο κοντά στην επεξεργασία της φυσικής γλώσσας ήταν τα μοντέλα αυτά, τόσο μεγάλωνε και ο αριθμός των δεδομένων που είχαν. Αυτό είχε ως αποτέλεσμα στην μετονομασία τους σε Large Language Models [13] .

Η διαδικασία που οδήγησε στην επέκταση των δεδομένων και των παραμέτρων τις οποίες χρησιμοποιούν τα μοντέλα για να επεξεργαστούν τη φυσική γλώσσα ονομάζεται fine tuning. Το fine tuning επιτρέπει στα μοντέλα να παραγάγουν πιο ακριβείς και

λεπτομερείς απαντήσεις οι οποίες ανταποκρίνονται στις ανάγκες των ανθρώπων. Το fine tuning πραγματοποιείται με την χρήση ειδικά διαμορφωμένων δεδομένων έτσι ώστε να επεκτείνουν την ικανότητα των μοντέλων να επεξεργαστούν φυσική γλώσσα.

Τέτοια δεδομένα, datasets, μπορεί να έχουν την μορφή διαλόγου, όπου το μοντέλο μπορεί να μάθει διάφορες μορφές input αλλά και τι output πρέπει να παραγάγει σε κάθε περίπτωση. Επίσης, τα δεδομένα μπορούν να είναι διαμορφωμένα στα θέλω του προγραμματιστή που θα κάνει fine tune το μοντέλο. Για παράδειγμα, υπάρχει η δυνατότητα να εκπαιδεύσει το μοντέλο να παραγάγει πιο αγενείς απαντήσεις, ή να απαντά παριστάνοντας μια ειδικότητα. Όλες αυτές οι προσαρμογές του μοντέλου εξαρτώνται από τα δεδομένα με τα οποία δημιουργήθηκε και αναπτύχθηκε το Large Language Model που χρησιμοποιεί.

Στα πρώτα στάδια της δημιουργίας των Large Language Models τα μοντέλα που υπήρχαν για την αναγνώριση και επεξεργασία φυσικής γλώσσας ήταν πολύ απλά. Τέτοια μοντέλα αποτελούν τα recurrent neural networks και τα long-short term memory networks, τα οποία λειτουργούσαν σειριακά, δηλαδή επεξεργάζονταν κάθε token ένα προς ένα. Ένα token είναι ο τρόπος που σπάζει ένα κείμενο έτσι ώστε να είναι πιο εύκολη η πρόβλεψη του επόμενου χαρακτήρα ή λέξης από το μοντέλο. Tokens αποτελούν ολόκληρες λέξεις, συλλαβές, ή ακόμη και σημεία στίξης.

Τα σειριακά μοντέλα όμως, δεν ήταν ικανά για την επεξεργασία μεγάλων δεδομένων. Έτσι δημιουργήθηκε μια νέα μέθοδος, η Transformer. Αυτή ήταν ικανή να αναγνωρίσει τις σχέσεις μεταξύ των tokens ταυτόχρονα, αντί ένα προς ένα. Ουσιαστικά επιτεύχθηκε μια παραλληλοποίηση σε ότι αφορά την εύρεση σχέσεων μεταξύ των tokens στα κείμενα.

2.1.3 Μεγάλα Γλωσσικά Μοντέλα Ανοικτού Κώδικα

Τα Μεγάλα Γλωσσικά Μοντέλα ανοικτού κώδικα είναι μοντέλα στα οποία ο καθένας έχει την δυνατότητα να επεξεργαστεί και να εκπαιδεύσει με δικά του δεδομένα. Μοντέλα όπως το ChatGPT είναι close source, και άρα δεν μπορεί ο καθένας να ενσωματώσει δικά του δεδομένα σε αυτά. Παρόλο που ένα close source μοντέλο είναι πιο ανθεκτικό σε

λανθασμένη γνώση, είναι πιο δύσκολο να επεκταθεί και να χρησιμοποιηθεί για συγκεκριμένες ανάγκες που έχει κάποιος χρήστης.

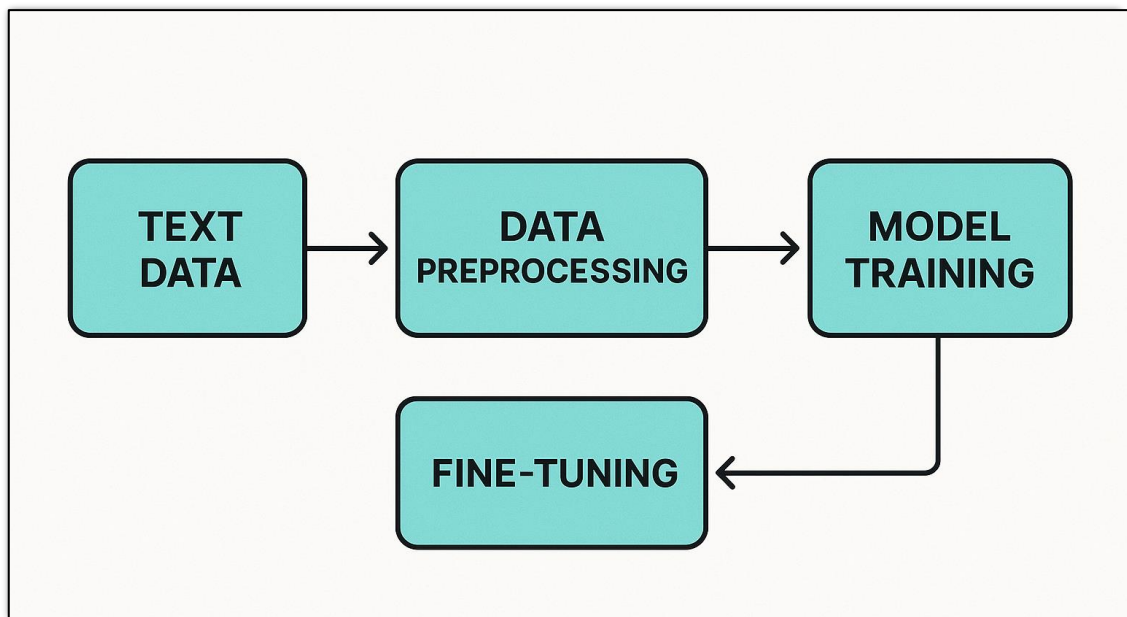
Λόγω του πιο πάνω, εμφανίστηκε η ανάγκη για μοντέλα όπου ο κάθε χρήστης θα μπορούσε να χρησιμοποιήσει για προσωπικούς σκοπούς και την ενσωμάτωση δεδομένων που ο ίδιος επιθυμεί. Αυτή όμως η δυνατότητα έφερε νέα εμπόδια σε ότι αφορά την εκπαίδευση ενός μοντέλου. Είναι γνωστό ότι για να εκπαιδευτεί ένα μοντέλο χρειάζονται πολλοί υπολογιστικοί πόροι. Οι πόροι αυτοί όμως δεν είναι διαθέσιμοι σε ένα απλό προγραμματιστή ή μια μικρή επιχείρηση.

Αυτό το πρόβλημα λύθηκε με την εμφάνιση διάφορων μεθόδων για εκπαίδευση μοντέλων οι οποίες δεν απαιτούσαν τεράστιους ή ακριβούς υπολογιστικούς πόρους. Παράδειγμα τέτοιων μεθόδων αποτελεί η LoRA, ή αλλιώς Low-Rank Adaptation, μέθοδος η οποία κατά την διάρκεια του fine tuning χρησιμοποιεί μικρά κομμάτια των δεδομένων που δόθηκαν [19].

2.1.4 Εκπαίδευση και Βελτιστοποίηση

Η εκπαίδευση είναι η τεχνική που χρησιμοποιείται για να εκπαιδευτεί ένα μοντέλο στην αναγνώριση, επεξεργασία και παραγωγή φυσικής γλώσσας. Η εκπαίδευση ενός Μεγάλου Γλωσσικού Μοντέλου χωρίζεται σε τρία κύρια στάδια (βλ. Σχήμα 2.1). Το πρώτο στάδιο αφορά τα δεδομένα που θα χρησιμοποιηθούν κατά την εκπαίδευση του μοντέλου, δηλαδή η συλλογή και η επεξεργασία των δεδομένων.

Τα δεδομένα πρέπει να είναι μεγάλα και κατανοητά προς το μοντέλο έτσι ώστε να είναι δυνατό να παραγάγει πιο ακριβή και σωστά αποτελέσματα. Τέτοια δεδομένα μπορούν να είναι κείμενα από βιβλία, καθημερινοί διάλογοι, ή ακόμη και επιστημονικά δεδομένα που αφορούν συγκεκριμένους κλάδους. Αφού μαζευτούν τα απαραίτητα δεδομένα, πρέπει να επεξεργαστούν έτσι ώστε να είναι κατανοητά και ιδανικά για την εκπαίδευση του μοντέλου.



Σχήμα 2.1: Ροή Εκπαίδευσης Μεγάλου Γλωσσικού Μοντέλου

Η επεξεργασία αφορά κυρίως την ποιότητα των δεδομένων. Ένα κείμενο πρέπει να έχει σωστή γραμματική σύνταξη έτσι ώστε και το ίδιο το μοντέλο να παραγάγει πιο ποιοτικά αποτελέσματα. Επίσης, τα δεδομένα που θα πάρει το μοντέλο πρέπει να είναι αμερόληπτα και αντικειμενικά.

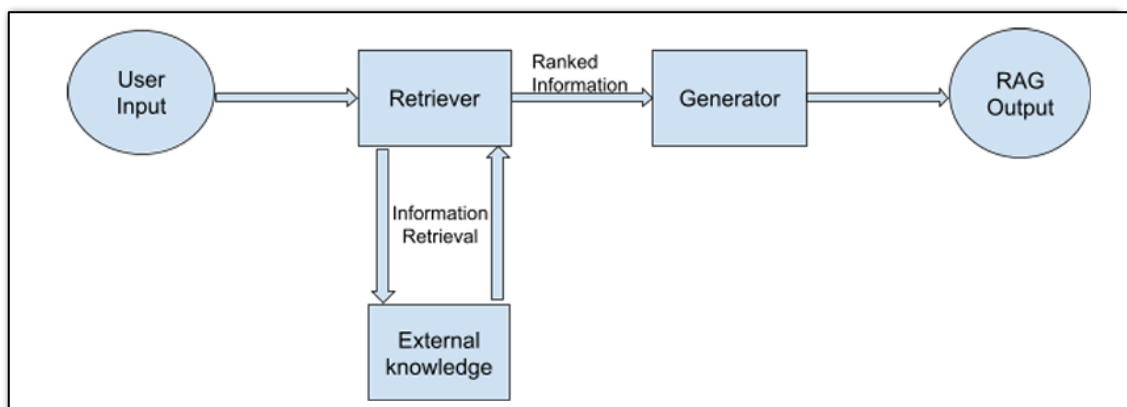
Το επόμενο βήμα στην εκπαίδευση ενός μοντέλου είναι ο καθορισμός της αρχιτεκτονικής του. Η αρχιτεκτονική που χρησιμοποιούν τα περισσότερα μοντέλα είναι η Transformer, η οποία αναφέρθηκε πιο πάνω. Το τελευταίο στάδιο είναι αυτό της βελτιστοποίησης (fine tuning).

Υπάρχουν αρκετές τεχνικές fine tuning. Η πρώτη είναι το επιβλεπόμενο (supervised) fine tuning το οποίο επιτυγχάνεται με την επέκταση του υπάρχοντος dataset προσθέτοντας δεδομένα που αφορούν συγκεκριμένο σκοπό. Έπειτα υπάρχει το ευθυγραμμιζόμενο (alignment) tuning το οποίο βασίζεται σε ανθρώπινη ανατροφοδότηση, έτσι ώστε να υπάρχει έλεγχος των αποτελεσμάτων ως προς την χρησιμότητα, την ειλικρίνεια και την αντικειμενικότητα των αποτελεσμάτων. Τέλος υπάρχει tuning που είναι αποδοτικό προς τις παραμέτρους, το οποίο στοχεύει στην επεξεργασία μικρού μέρους των προϋπαρχόντων δεδομένων, κρατώντας το μεγαλύτερο μέρος απαράλλακτο έτσι ώστε να επιτευχθεί μείωση κόστους [66].

2.1.5 Retrieval – Augmented Generation (RAG)

Η Retrieval – Augmented Generation (RAG) είναι μια αρχιτεκτονική η οποία συνδυάζει δύο στοιχεία. Το πρώτο αποτελεί ένα μηχανισμό ανάκτησης πληροφοριών από εξωτερικές πηγές πέρα των δικών του δεδομένων. Το δεύτερο στοιχείο είναι η χρήση των ανακτημένων αυτών πληροφοριών για την παραγωγή απάντησης σε μορφή φυσικής γλώσσας (βλ. Σχήμα 2.2).

Για να ανακτήσει δεδομένα, χρησιμοποιεί dense vectors. Τέτοιες δομές είναι ικανές να αναγνωρίσουν σχέσεις μεταξύ των δεδομένων, κάνοντας πιο εύκολη την κατανόηση των πληροφοριών από το μοντέλο. Στη συνέχεια, για να παραχθεί η απάντηση, το μοντέλο περνάει τις πληροφορίες από ένα αλγόριθμο αναπαραγωγής λόγου, ο οποίος είναι βασισμένος στην αρχιτεκτονική transformer. Έτσι το μοντέλο πλέον έχει παραγάγει μια απάντηση η οποία είναι βασισμένη σε δεδομένα τα οποία δεν υπήρχαν στο αρχικό του dataset [49].



Σχήμα 2.2: Βασική Ροή του Συστήματος RAG [49]

2.1.6 Βελτιστοποίηση vs. RAG

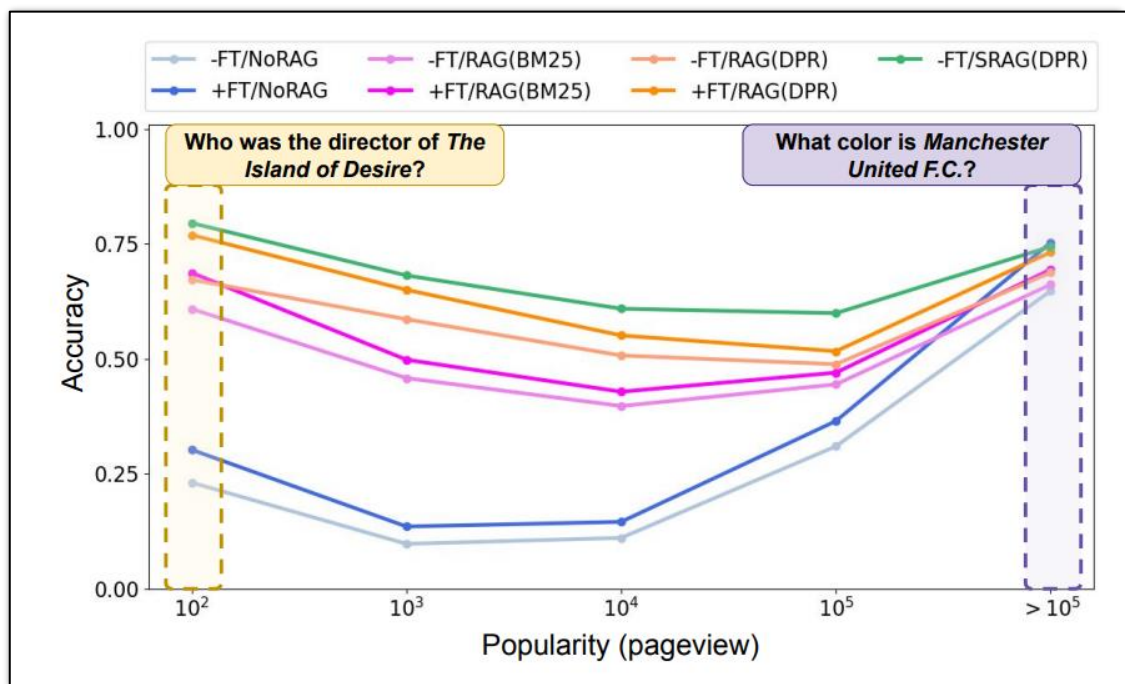
Όπως αναφέρθηκε πιο πάνω, οι δύο τεχνικές, fine tuning και RAG, χρησιμοποιούνται για τον εμπλουτισμό της γνώσης ενός έτοιμου Large Language Model. Μεταξύ τους οι τεχνικές αυτές έχουν διαφορές όχι μόνο ως προς τον τρόπο λειτουργίας ο οποίος αναλύθηκε πιο πάνω, αλλά και ως προς την απόδοσή τους. Η κάθε τεχνική είναι

αποτελεσματική σε διαφορετικό τομέα όσον αφορά την εκπαίδευση και την εκμάθηση του μοντέλου.

Με βάση έρευνες που διεξάχθηκαν, το fine tuning αυξάνει την απόδοση του μοντέλου. Αυτό ισχύει καθώς το fine tuning εκμεταλλεύεται προϋπάρχουσες γνώσεις οι οποίες είναι ήδη γνωστές στο μοντέλο και το βοηθά να εστιάζει πάνω σε αυτές.

Από την άλλη, η τεχνική RAG είναι σημαντικά πιο αποδοτική από το fine tuning σε ότι αφορά την εκπαίδευση του μοντέλου σε πολύ συγκεκριμένη και λιγότερο γνωστή γνώση.

Στην έρευνά τους, οι H. Soudani, E. Kanoulas, και F. Hasibi μελέτησαν την ακρίβεια στις απαντήσεις ενός μοντέλου το οποίο εκπαίδευσαν με RAG και fine tuning. Χρησιμοποίησαν τις δύο μεθόδους ξεχωριστά αλλά και μαζί. Μετά από την ανάλυση του κάθε συνδυασμού ήταν φανερό ότι η RAG ήταν πολύ πιο αποδοτική μέθοδος από το fine tuning (βλ. Σχήμα 2.3) [14].



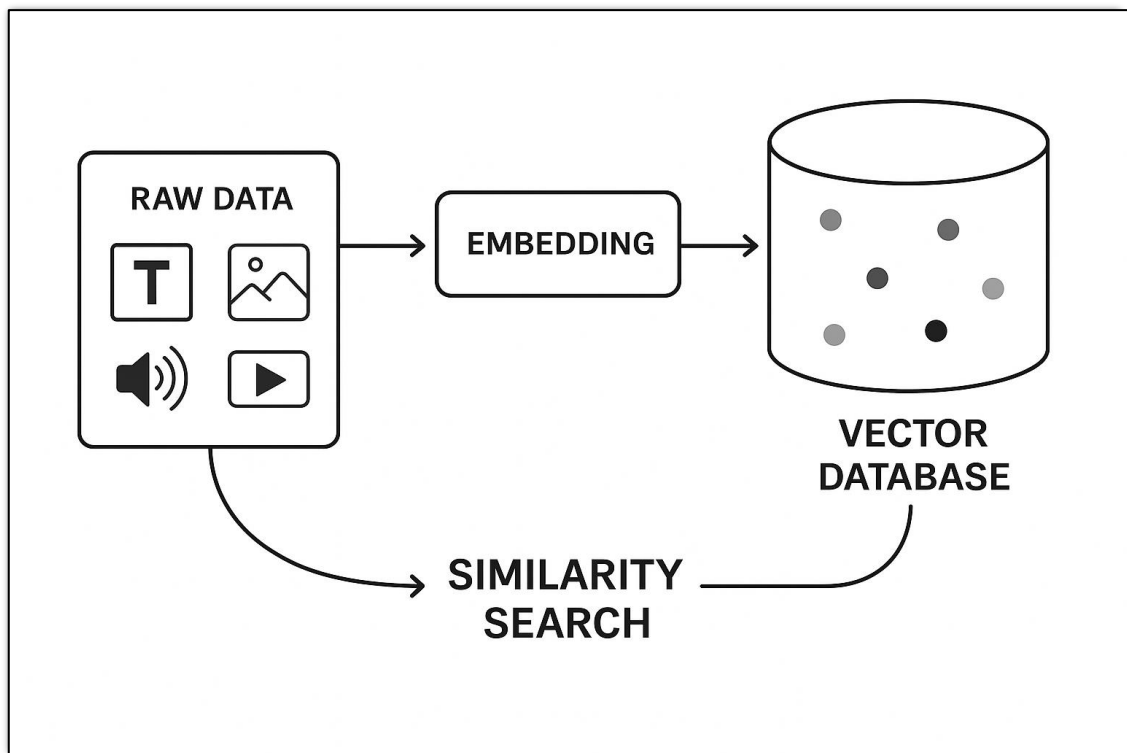
Σχήμα 2.3: Σύγκριση RAG και της μεθόδου fine-tuning ως προς την απόδοση του StableLM2 [14]

2.1.7 Διανυσματικές Βάσεις

Οι διανυσματικές βάσεις δεδομένων είναι ένα είδος Βάσεων Δεδομένων οι οποίες αποθηκεύουν τα δεδομένα ως πολυδιάστατα διανύσματα. Αυτά τα διανύσματα αντιπροσωπεύουν με μαθηματικό τρόπο τα χαρακτηριστικά των δεδομένων που αποθηκεύονται.

Τα διανύσματα συνήθως παράγονται μέσω της εφαρμογής κάποιας συνάρτησης διανυσματικής ενσωμάτωσης (embedding function) στα αρχικά δεδομένα, ώστε να αποκτήσουν μια διανυσματική αναπαράσταση κατάλληλη για επεξεργασία από υπολογιστικά συστήματα (βλ. Σχήμα 2.4).

Συγκρίνοντας τις διανυσματικές βάσεις δεδομένων με τις πιο παραδοσιακές βάσεις, έχουν πολλά πλεονεκτήματα, ειδικά όταν χρησιμοποιούνται για μηχανική μάθηση. Αρχικά, ο τρόπος που παίρνουν δεδομένα είναι γρήγορος και ακριβής. Σε αντίθεση με τις παραδοσιακές βάσεις που τα αποτελέσματα επιστρέφονται όταν υπάρχει ακριβής απάντηση, στις διανυσματικές βάσεις τα αποτελέσματα επιστρέφονται όταν υπάρχει ομοιότητα μεταξύ της ερώτησης και της απάντησης. Αυτό είναι ιδιαίτερα χρήσιμο όταν ο χρήστης επιθυμεί να χρησιμοποιήσει δεδομένα για σκοπούς επεξεργασίας φυσικής γλώσσας.



Σχήμα 2.4: Ροή μετατροπής δεδομένων σε διανύσματα

Ακόμη, οι διανυσματικές βάσεις υποστηρίζουν αδόμητα δεδομένα, όπως εικόνες και βίντεο, τα οποία μετατρέπουν σε διανύσματα. Επιπρόσθετα, οι διανυσματικές βάσεις έχουν τη δυνατότητα να επεξεργαστούν μεγάλα δεδομένα σε πραγματικό χρόνο. Αυτή η λειτουργία είναι απαραίτητη για εφαρμογές τεχνητής νοημοσύνης.

Η σύγκριση και η εύρεση όμοιων δεδομένων μέσα σε μια διανυσματική βάση γίνεται με την χρήση της συνημιτονοειδούς ομοιότητας (cosine similarity) (βλ. Σχήμα 2.5). Η εξίσωση αυτή χρησιμοποιείται για την σύγκριση δύο διανυσμάτων ως προς την κατεύθυνση τους με παρόμοιο τρόπο όπως και στην γραμμική άλγεβρα. Δύο παρόμοια διανύσματα έχουν μικρή απόκλιση και άρα η γωνιά που σχηματίζεται αναμεσά τους είναι μικρή. Από τα μαθηματικά ισχύει ότι όσο πιο μικρή είναι μια γωνιά που σχηματίζουν δύο διανύσματα, τόσο μεγαλύτερο είναι και το συνημίτονό της, δεδομένου ότι η γωνιά κυμαίνεται από 0 μέχρι 90 μοίρες. Αυτή η φόρμουλα δίνει απάντηση σε δεκαδικό αριθμό. Ο αριθμός αυτός όσο πιο κοντά στη μονάδα είναι, σημαίνει ότι τόσο πιο όμοια είναι τα δεδομένα που συγκρίνονται [65].

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Σχήμα 2.5: Εξίσωση Συνημιτονοειδούς Ομοιότητας

2.1.8 IndexedDB

Η IndexedDB είναι μια NoSQL βάση δεδομένων η οποία στηρίζεται στην HTML5 και είναι ικανή να αποθηκεύει διαφόρων ειδών δεδομένα. Η βάση αυτή επιτρέπει την αποθήκευση δεδομένων από την πλευρά του χρήστη αντί του προγραμματιστή.

Η λογική που χρησιμοποιεί η βάση αυτή είναι εκείνη του κλειδιού – τιμής. Επίσης, στο πιο κάτω μέρος της, χρησιμοποιεί B-Trees για την αποθήκευση των δεδομένων στο σύστημα αρχείων. Λόγω της χρήσης του μοντέλου κλειδιού – τιμής, τα δεδομένα μπορούν εύκολα να ανακτηθούν με την χρήση του κλειδιού τους, ή με την χρήση ενός άλλου δείκτη που τα αντιπροσωπεύει.

Επιπρόσθετα, η IndexedDB υποστηρίζεται από διάφορους περιηγητές ιστού χωρίς να απαιτείται η εγκατάσταση επιπρόσθετων στοιχείων λογισμικού.

Η IndexedDB έχει πολλά θετικά χαρακτηριστικά. Ένα κύριο χαρακτηριστικό είναι ότι είναι τοπική για κάθε χρήστη. Δεν είναι μόνο τοπική στη μηχανή του χρήστη αλλά είναι τοπική και στην κάθε εφαρμογή. Αυτό σημαίνει ότι κάθε ιστότοπος που δημιουργεί μια IndexedDB, έχει τη δική του βάση με τα δικά του δεδομένα τα οποία δεν επικοινωνούν με τα δεδομένα άλλου ιστότοπου.

Ακόμη ένα χαρακτηριστικό της IndexedDB είναι ότι είναι NoSQL βάση δεδομένων. Δηλαδή, δεν αποθηκεύει δεδομένα με τον παραδοσιακό τρόπο σε πίνακες. Όπως

προαναφέρθηκε, αποθηκεύει δεδομένα με βάση τα κλειδιά τους. Έτσι, είναι δυνατή η αποθήκευση αδόμητων δεδομένων. Ουσιαστικά αυτή η βάση δεδομένων είναι αντικειμενοστραφής και όχι σχεσιακή.

Οι τρεις βασικές λειτουργίες που αφορούν την βάση είναι το άνοιγμά της, δηλαδή η δημιουργία σύνδεσης με τη βάση, η δημιουργία του αποθηκευτικού χώρου, και το κλείσιμό της. Η δημιουργία αποθηκευτικού χώρου γίνεται μόνο την πρώτη φορά που θα ανοιχτεί η βάση.

Πέρα από τις λειτουργίες που αφορούν συγκεκριμένα την βάση, υπάρχουν και τέσσερις βασικές λειτουργίες που αφορούν τα δεδομένα που είναι αποθηκευμένα σε αυτή. Αφού η IndexedDB είναι NoSQL βάση δεδομένων, δεν χαρακτηρίζεται από τις κλασσικές λειτουργίες μιας σχεσιακής βάσης.

Η πρώτη λειτουργία είναι αυτή της δημιουργίας, η οποία γίνεται με τη χρήση της συνάρτησης `add()`. Με αυτή, δημιουργείται ένα αντικείμενο το οποίο χαρακτηρίζεται από ένα μοναδικό κλειδί. Είναι αντίστοιχη με την εντολή “INSERT” σε μια σχεσιακή βάση.

Επιπρόσθετα, υπάρχει η λειτουργία της ανάγνωσης, η οποία πραγματοποιείται με τη συνάρτηση `get()`. Για την εκτέλεση της λειτουργίας αυτής, απαιτείται η παραχώρηση του κλειδιού ή του δείκτη που χαρακτηρίζει το αντικείμενο που θέλουμε να επιστραφεί. Η αντίστοιχη λειτουργία σε μια σχεσιακή βάση είναι αυτή του “SELECT”.

Ακόμη, υπάρχει η δυνατότητα αναβάθμισης και επεξεργασίας των δεδομένων. Αυτό επιτυγχάνεται με τη χρήση της συνάρτησης `put()`. Η συγκεκριμένη αυτή συνάρτηση μπορεί να χρησιμοποιηθεί και στην πρόσθεση νέων δεδομένων στη βάση.

Τέλος, είναι δυνατή και η διαγραφή εγγραφών στη βάση δεδομένων με τη συνάρτηση `delete()` η οποία λειτουργεί παρόμοια με το “DELETE” στις σχεσιακές βάσεις δεδομένων [1][29].

2.2 Σχετική Εργασία

2.2.1 Βιομηχανικές Εφαρμογές

Υπάρχουν αρκετές υπηρεσίες και εταιρίες οι οποίες υλοποίησαν παρόμοια συστήματα για αναζήτηση νομικών εγγράφων. Οι περισσότερες αποτελούν μια απλή βάση δεδομένων στην οποία μπορείς να αναζητήσεις έγγραφα, χωρίς τη χρήση τεχνητής νοημοσύνης.

Έτσι, η ατομική διπλωματική εργασία ξεκίνησε με στόχο την ανάπτυξη παρόμοιου συστήματος, το οποίο αξιοποιεί την τεχνητή νοημοσύνη. Συγκεκριμένα, είχε ως στόχο την εκπαίδευση ενός Large Language Model πάνω σε νομικά έγγραφα του Πανεπιστημίου Κύπρου. Το τελικό μοντέλο θα είχε την δυνατότητα να απαντά στους χρήστες ερωτήσεις οι οποίες αφορούν νομοθεσίες του Πανεπιστημίου Κύπρου.

Ένα χαρακτηριστικό παράδειγμα ενός συστήματος χωρίς τεχνητή νοημοσύνη αποτελεί η ιστοσελίδα του Παγκύπριου Δικηγορικού Συλλόγου. Αυτή η ιστοσελίδα παρέχει δωρεάν όλα τα νομικά έγγραφα και πηγές δικαίου τόσο σε κυπριακό αλλά και σε διεθνές επίπεδο. Μαζί με τα αρχεία υπάρχουν οι χρονολογίες της κάθε απόφασης (βλ. Σχήμα 2.6).

Όσον αφορά το Πανεπιστήμιο, στην ιστοσελίδα CyLaw, υπάρχουν αρκετά έγγραφα και νομοθεσίες όπως φαίνεται πιο κάτω.

Αποτελέσματα Έρευνας: Νέα Έρευνα: Όλες οι Λέξεις: Πανεπιστήμιο Κύπρου [πανεπιστημι* and κυπρο*]	
Κατηγορίες: Αποφάσεις Ανωτάτου Δικαστηρίου (1883-2023) Αποφάσεις Νέου Ανωτάτου Συνταγματικού Δικαστηρίου Αποφάσεις Εφετρείου Αποφάσεις Νέου Ανωτάτου Δικαστηρίου	
Αυτά είναι τα κείμενα 1 - 20 από τα 2750 κείμενα που έχουν ανευρεθεί.	
1.	Πανεπιστήμιο Κύπρου v. Κωνσταντίνου κ.α. (1994) 3 ΑΑΑ 145 126510 bytes [100] Βάση Δεδομένων: Αποφάσεις Ανωτάτου Δικαστηρίου (1883-2023)
2.	ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ v. ΘΕΟΔΩΡΟΥ Π. ΑΓΓΕΛΙΔΗ, Έφεση κατά απόφασης Διοικητικού Δικαστηρίου αρ. 171/2019, 22/5/2023 48215 bytes [96] Βάση Δεδομένων: Αποφάσεις Ανωτάτου Δικαστηρίου (1883-2023)
3.	ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ v. ΙΩΣΗΦΙΔΟΥ, Έφεση κατά απόφασης Διοικητικού Δικαστηρίου Αρ. 139/2019, 20/1/2022 92763 bytes [96] Βάση Δεδομένων: Αποφάσεις Ανωτάτου Δικαστηρίου (1883-2023)
4.	ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΚΥΠΡΟΥ v. Κοινοπραξίας Φυσικών και Νομικών Προσώπων Αντωνίου κ.α., Αναθεωρητική Έφεση Αρ. 139/2012, 8/6/2018 41763 bytes [93] Βάση Δεδομένων: Αποφάσεις Ανωτάτου Δικαστηρίου (1883-2023)
5.	Πανεπιστήμιο Κύπρου v. Στεφάνου κ.α. (1994) 3 ΑΑΑ 133 67670 bytes [92] Βάση Δεδομένων: Αποφάσεις Ανωτάτου Δικαστηρίου (1883-2023)

Σχήμα 2.6: Εμφάνιση Νομικών Εγγράφων στο UCYLaw

Χρησιμοποιώντας την μπάρα αναζήτησης, οι χρήστες μπορούν να βάλουν λέξεις κλειδιά για να βρουν τα έγγραφα που αναζητούν. Όταν εμφανιστούν τα αντίστοιχα αποτελέσματα, ο χρήστης μπορεί να επιλέξει τη νομοθεσία που επιθυμεί. Υπάρχει η δυνατότητα προβολής της νομοθεσίας απευθείας στην ιστοσελίδα, αλλά και η προβολή της ως αρχείο PDF [8].

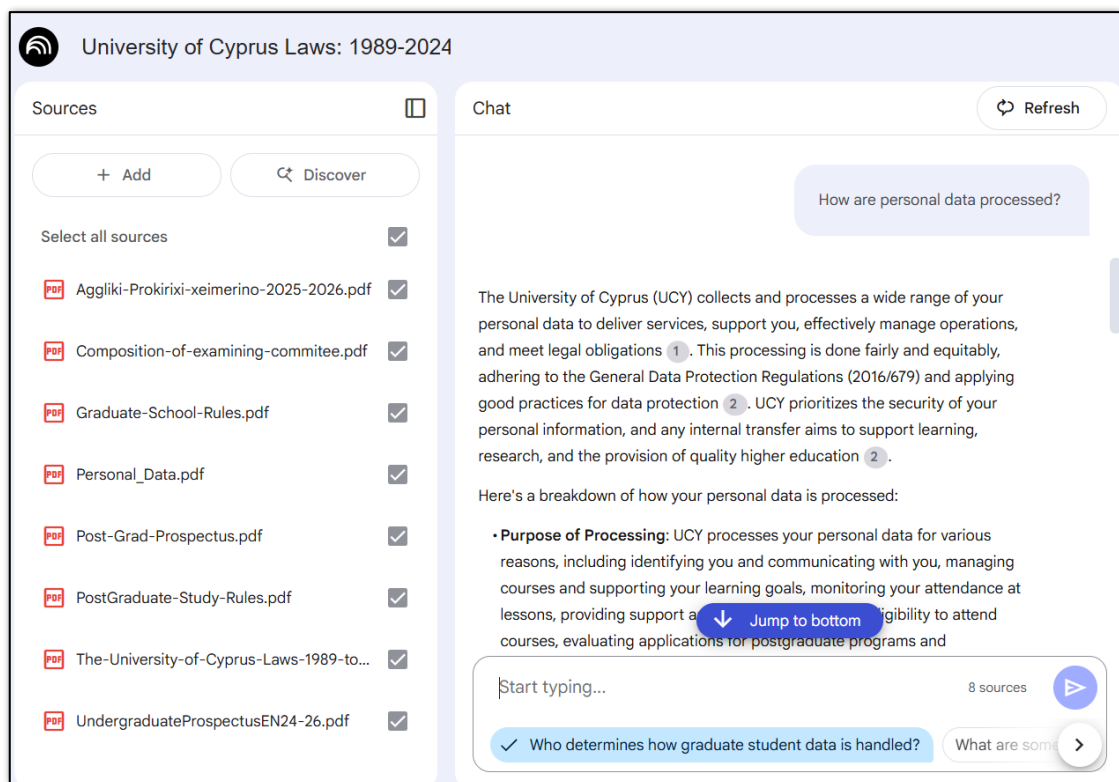
Παρόμοιες εφαρμογές υπάρχουν γενικά στο διαδίκτυο οι οποίες χρησιμοποιούν τεχνητή νοημοσύνη για να βοηθήσουν τον χρήστη να καταλάβει τους νόμους. Μια τέτοια πλατφόρμα είναι η Harvey, η οποία εξειδικεύεται στη χρήση τεχνητής νοημοσύνης για κατανόηση νομικών εγγράφων. Στηρίζεται πάνω σε διεθνή δεδομένα, αλλά έχει και υποθέσεις που αφορούν την Ευρωπαϊκή Ένωση, αλλά και συγκεκριμένων χωρών [15].

Επιπρόσθετα, υπάρχουν πολλά GPTs τα οποία μπορούν να χρησιμοποιηθούν στην πλατφόρμα του ChatGPT τα οποία κάνουν παρόμοια δουλειά με τα πιο πάνω. Δηλαδή μπορούν να παραχωρήσουν νομικές συμβουλές με βάση τις νομοθεσίες που θα τους ζητηθούν.

Ταυτόχρονα υπάρχουν και πολλές εφαρμογές οι οποίες χρησιμοποιούν την τεχνική RAG για να επεξεργάζονται αρχεία που τους δίνει ο χρήστης και να απαντά με βάση αυτά. Μια από τις πιο γνωστές εφαρμογές με αυτή τη λειτουργία αποτελεί το NotebookLM [11].

Στο NotebookLM οι χρήστες έχουν την δυνατότητα να αναρτήσουν δικά τους δεδομένα όπως διαλέξεις μαθημάτων, βιβλία, και άλλα σημαντικά έγγραφα. Έπειτα τα δεδομένα αυτά χρησιμοποιούνται από το μοντέλο για την απάντηση ερωτήσεων και τη βοήθεια στο διάβασμα (βλ. Σχήμα 2.7).

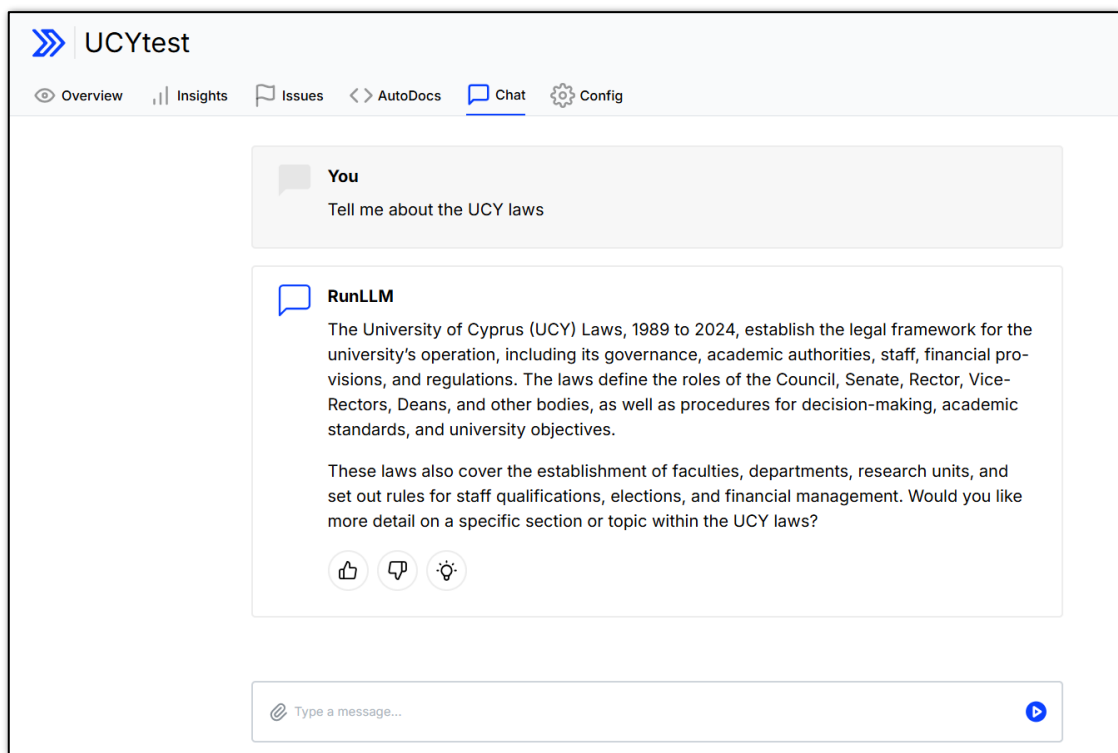
Οι χρήστες μπορούν να ρωτήσουν το σύστημα ερωτήσεις σχετικά με τα δεδομένα που έχουν αναρτήσει και το σύστημα με την σειρά του θα απαντήσει. Μια ακόμα δυνατότητα του συστήματος είναι να εμφανίσει το συγκεκριμένο έγγραφο από το οποίο πήρε την πληροφορία.



Σχήμα 2.7: Συνομιλία στο NotebookLM

Μια παρόμοια πλατφόρμα με το NotebookLM αποτελεί το RunLLM. Σε αυτή την ιστοσελίδα ο χρήστης μπορεί να αναρτήσει έγγραφα ανεξαρτήτως δομής. Μέσα από αυτά, το προϊόν μαθαίνει και είναι ικανό να απαντήσει ερωτήσεις του χρήστη που αφορούν το υλικό που ανέβασε (βλ. Σχήμα 2.8).

Για να επιτευχθεί η λειτουργία αυτή, η πλατφόρμα είναι συνδεδεμένη με πολλά γλωσσικά μοντέλα τα οποία μπορούν να εκπαιδευτούν στις ανάγκες του κάθε χρήστη. Κύρια στοιχεία της πλατφόρμας αυτής αποτελούν η γρήγορη μάθηση, οι ακριβείς απαντήσεις, και η αναζήτηση του διαδικτύου από το ίδιο το μοντέλο [48].



Σχήμα 2.8: Συνομιλία στο RunLLM

2.2.2 Ακαδημαϊκές Εργασίες

Παρόμοιες δουλειές έγιναν και σε ακαδημαϊκό επίπεδο με πολλούς επιστήμονες να δημιουργούν τα δικά τους large language models σχεδιασμένα για την κατανόηση και επεξεργασία νομικών εγγράφων.

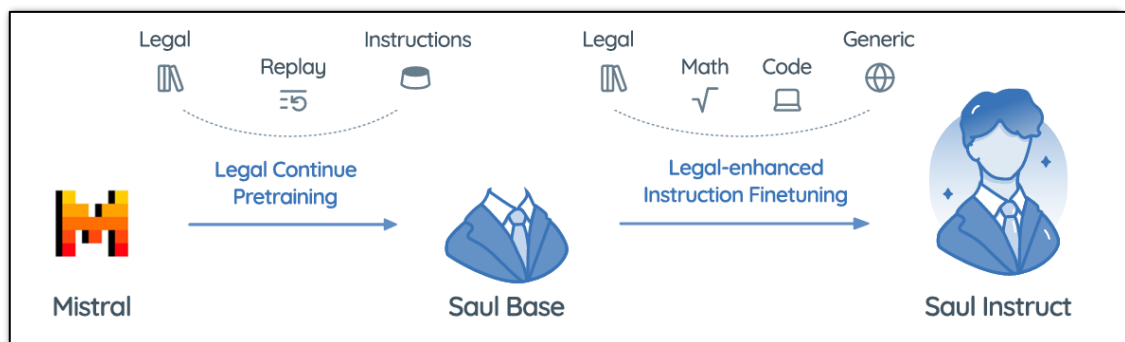
2.2.2.1 SaulLM-7B

Ένα τέτοιο παράδειγμα είναι το SaulLM-7B το οποίο είναι ένα large language model σχεδιασμένο για νομική. Για τη δημιουργία του συγκεκριμένου μοντέλου χρησιμοποιήθηκε το μοντέλο Mistral7b και έγινε fine tuned με την χρήση υψηλού επιπέδου δεδομένων τα οποία ήταν εστιασμένα πάνω σε νομοθεσίες (βλ. Σχήμα 2.9).

Για την εκπαίδευση του συγκεκριμένου μοντέλου χρησιμοποιήθηκαν δύο μέθοδοι. Η πρώτη μέθοδος ήταν πιο γενική έτσι ώστε να βελτιωθεί το μοντέλο στην κατανόηση

φυσικής γλώσσας και καλύτερη επικοινωνία με άνθρωπο. Η δεύτερη μέθοδος εστίαζε στην τροφοδότηση του μοντέλου με υλικό που αφορά την νομική.

Το υλικό αυτό ήταν στα αγγλικά και είχε κομμάτια από νομοθεσίες και υποθέσεις που αφορούσαν την Αμερική, την Αυστραλία, αλλά και την Ευρώπη. Ήταν αρκετά σημαντικό να εκπαιδευτεί το μοντέλο με δεδομένα από διάφορες χώρες, αφού οι νόμοι διαφέρουν από χώρα σε χώρα, και από οργανισμό σε οργανισμό.



Σχήμα 2.9: Διαδικασία Κατασκευής του SaulLM-7B [42]

Στο τέλος, οι ερευνητές σύγκριναν το μοντέλο που δημιούργησαν μαζί με άλλα μοντέλα ως προς την ικανότητα του να σκέφτεται και να παίρνει αποφάσεις που αφορούν την νομική. Επίσης, σύγκριναν το μοντέλο και ως προς την πολυπλοκότητα και ικανότητά του να επεξεργαστεί νέα νομικά έγγραφα που του δίνονταν.

Τα αποτελέσματα των πειραμάτων ήταν θετικά, δείχνοντας ότι πραγματικά η εκπαίδευση ενός μοντέλου με συγκεκριμένα δεδομένα, σε αυτή την περίπτωση νομικά έγγραφα και υποθέσεις, το καθιστά πιο ικανό να ανταπεξέλθει σε ερωτήσεις και αιτήματα τέτοιας φύσης [42].

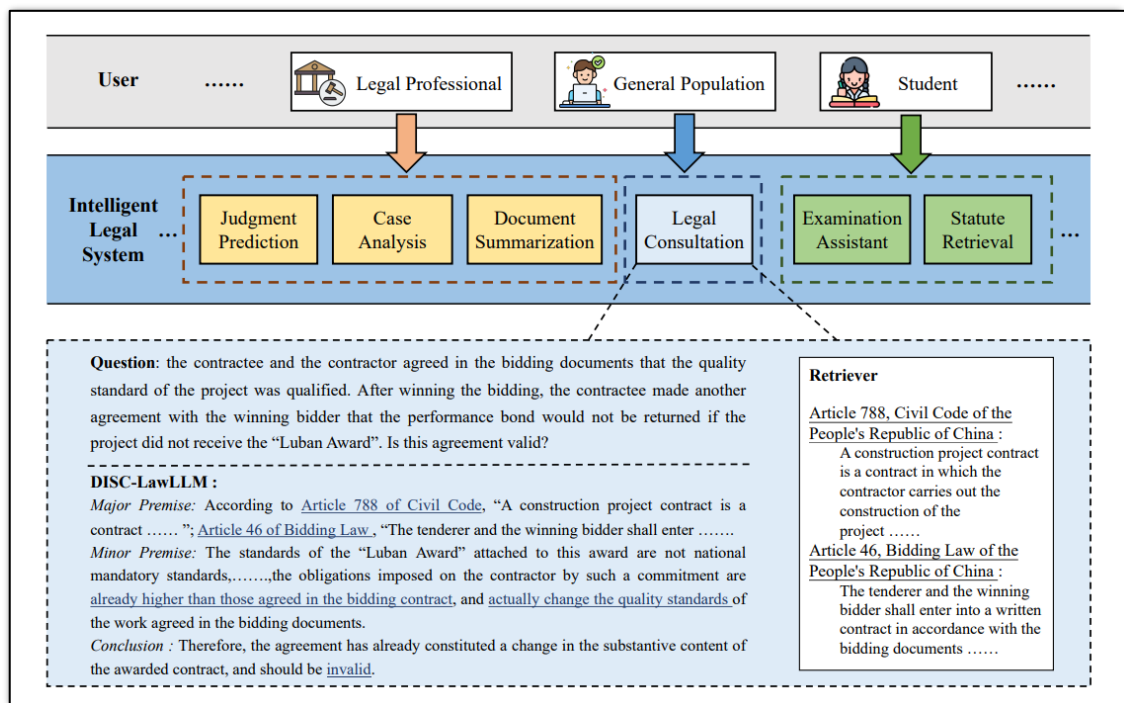
2.2.2.2 DISC-LawLLM

Σε μια άλλη μελέτη που έγινε από πανεπιστήμια στην Κίνα, ερευνητές δημιούργησαν το DISC-LawLLM, ένα έξυπνο σύστημα όπου με τη χρήση fine-tuned LLMs παρέχει νομικές υπηρεσίες στους χρήστες του. Αυτό το μοντέλο είναι διαθέσιμο στο διαδίκτυο

για εγκατάσταση και προσωπική χρήση. Το αρχικό μοντέλο που χρησιμοποιήθηκε για την ανάπτυξη αυτού του συστήματος είναι το ChineseLLM το οποίο έχει 13 δισεκατομμύρια παραμέτρους.

Για τη δημιουργία αυτού του μοντέλου έγινε χρήση τόσο fine tuning μεθόδων όσο και χρήση RAG. Το μοντέλο αυτό δημιουργήθηκε με σκοπό τη μείωση της βιβλιογραφικής δουλειάς των νομικών επαγγελματιών αλλά και τη διευκόλυνση της αναζήτησης νομοθετικών συμβουλών από απλούς πολίτες.

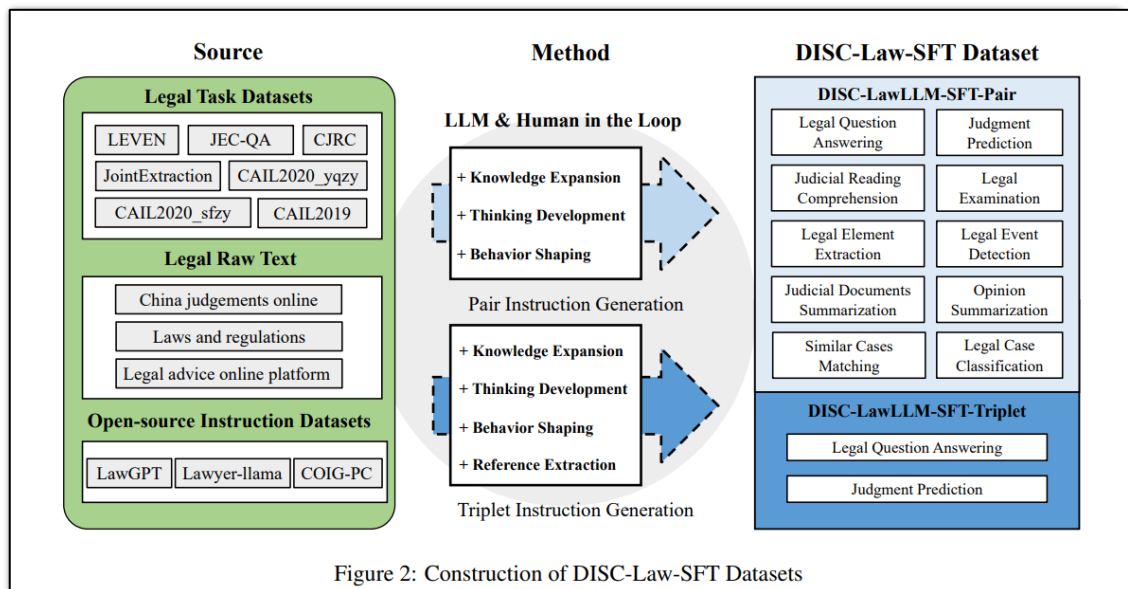
Πιο συγκεκριμένα, για νομικούς επαγγελματίες το σύστημα φτιάχτηκε με σκοπό να παρέχει προηγμένα εργαλεία για ανάκτηση νομοθεσιών, ανάλυση νομοθετικών υποθέσεων, και τη σύνοψη νομοθετικών εγγράφων. Ταυτόχρονα, το σύστημα αυτό μπορεί να λειτουργήσει και ως βοηθός στους φοιτητές νομικής, αφού μπορεί να επεκτείνει τις γνώσεις τους και να παρέχει λύσεις σε εξεταζόμενα θέματα. Όσον αφορά τον γενικό πληθυσμό, είναι ικανό να προσφέρει απλή ερμηνεία της νομοθεσίας και βοήθεια σε ότι αφορά νομικές διαφωνίες (βλ. Σχήμα 2.10).



Σχήμα 2.10: DISC-LawLLM Εξυπηρετεί Διαφορετικούς Χρήστες σε Διαφορετικά Νομικά Σενάρια [50]

Η δημιουργία ενός τέτοιου μοντέλου είχε δύο κύριες δυσκολίες, την υψηλή απαίτηση για σωστή αιτιολόγηση, και η βέλτιστη ικανότητα ανάκτησης δεδομένων και εξαγωγής σωστών συμπερασμάτων. Η πρώτη δυσκολία εμφανίζεται στο γεγονός ότι οι απαντήσεις που αφορούν τις νομικές συμβουλές πρέπει να έχουν σωστή δομή νομικού συλλογισμού. Η δεύτερη δυσκολία αφορά το γεγονός ότι για να γίνει σωστή ανάκτηση δεδομένων πρέπει να έχει βαθιά κατανόηση του αντικειμένου. Επίσης, η ανάκτηση δεδομένων γίνεται ακόμη πιο περίπλοκη με τη συνεχή επέκταση της νομικής γνώσης (βλ. Σχήμα 2.11).

Τέλος, οι ερευνητές έκαναν αξιολόγηση του μοντέλου συγκρίνοντας τις απαντήσεις του σε ερωτήσεις τόσο αντικειμενικής όσο και υποκειμενικής φύσης, σε σχέση με τις απαντήσεις άλλων μοντέλων. Για τις υποκειμενικές ερωτήσεις, παραχωρούσαν και μια απάντηση έτσι ώστε να αποφευχθεί τυχόν μεροληψία. Τα αποτελέσματα ήταν θετικά καθώς έδειξαν ότι ακόμη και σε σύγκριση με μοντέλα με περισσότερες παραμέτρους, το DISC-LawLLM είχε πολύ καλύτερες γνώσεις στην κινέζικη νομοθεσία, με αποτέλεσμα να παράγει πιο ακριβείς και αξιόπιστες απαντήσεις [50].



Σχήμα 2.11: Κατασκευή Δεδομένων για το DISC-Law-SFT [50]

2.2.2.3 Fine Tuning Llama 3

Σε μια άλλη έρευνα που διεξάχθηκε, ερευνητές χρησιμοποίησαν το μοντέλο Llama 3. Η μεγαλύτερη δουλειά για την εκπαίδευση του μοντέλου έγινε στη δημιουργία των δεδομένων των οποίων θα χρησιμοποιούνταν για αυτό το σκοπό.

Αρχικά, μάζεψαν μια ευρεία συλλογή από νομικά έγγραφα· τέτοια έγγραφα αποτελούν νομολογίες, νομοθετικές διατάξεις, και δικαιοδοσίες. Αφού έγινε η περισυλλογή, ακολούθησαν τα βήματα της εξαγωγής μόνο σημαντικού κειμένου από τα έγγραφα, της τυποποίησης της ορολογίας έτσι ώστε να υπάρχει συνοχή σε όλο το κείμενο, και τον διαχωρισμό του κειμένου σε πιο μικρά κομμάτια.

Το fine tuning του μοντέλου έγινε με ένα αλγόριθμο ο οποίος κατ' επανάληψη εκπαιδεύει το μοντέλο με κύκλους εκπαίδευσης, όπου ο κάθε κύκλος είχε σκοπό την σταδιακή βελτίωση του μοντέλου στην επεξεργασία των νομικών κειμένων (βλ. Σχήμα 2.12).

Algorithm 1 Iterative Fine-Tuning of Llama 3

```
1: Initialize hyperparameters:  $\eta$ ,  $\lambda$ , and dropout rate  $\delta$ 
2: Set maximum cycles  $N$ , current cycle  $n = 1$ 
3: while  $n \leq N$  do
4:   Train model on legal dataset with current  $\eta$ ,  $\lambda$ , and  $\delta$ 
5:   Evaluate model on validation set
6:   Calculate loss  $\mathcal{L}$  and accuracy  $\alpha$ 
7:   Adjust  $\eta$  and  $\lambda$  using backpropagation:
8:    $\eta \leftarrow \eta - \nabla_{\eta} \mathcal{L}(\eta, \lambda)$ 
9:    $\lambda \leftarrow \lambda - \nabla_{\lambda} \mathcal{L}(\eta, \lambda)$ 
10:  if  $\alpha$  improved and  $\mathcal{L}$  decreased then
11:    Fine-tune dropout rate  $\delta$  to reduce overfitting
12:  end if
13:  Apply gradient accumulation and mixed precision techniques
14:   $n \leftarrow n + 1$ 
15: end while
```

Σχήμα 2.12: Αλγόριθμος για Fine Tune του Llama 3 [37]

Μετά το τέλος της εκπαίδευσης του μοντέλου, έγιναν διάφοροι έλεγχοι πάνω στο μοντέλο. Αυτοί οι έλεγχοι είχαν σκοπό να εξετάσουν την αναγνώριση νομικών εγγράφων από το μοντέλο αλλά και την αξιολόγηση των απαντήσεων με βάση την ακρίβεια και την ορθότητα των απαντήσεων που παράγει (βλ. Πίνακα 2.1).

Metric	Pre-Tuning	Post-Tuning	Improvement
Accuracy (%)	80	93	+13%
Precision (%)	77	89	+12%
Recall (%)	70	85	+15%
F1-Score (%)	72	83	+11%
Legal Argument Recognition Accuracy (%)	65	82	+17%
Contract Element Extraction Precision (%)	68	82	+14%

Πίνακας 2.1: Μετρικές Απόδοσης Πριν και Μετά το Fine Tune [37]

Συμπερασματικά, το fine tuning βοήθησε το μοντέλο να κατανοεί τα νομικά έγγραφα και να παράγει πιο σωστές απαντήσεις. Ακόμη ένας σκοπός της έρευνας ήταν να θίξει τη σημαντικότητα της εκπαίδευσης των large language models σε συγκεκριμένα δεδομένα που αφορούν κάποιο κλάδο [37].

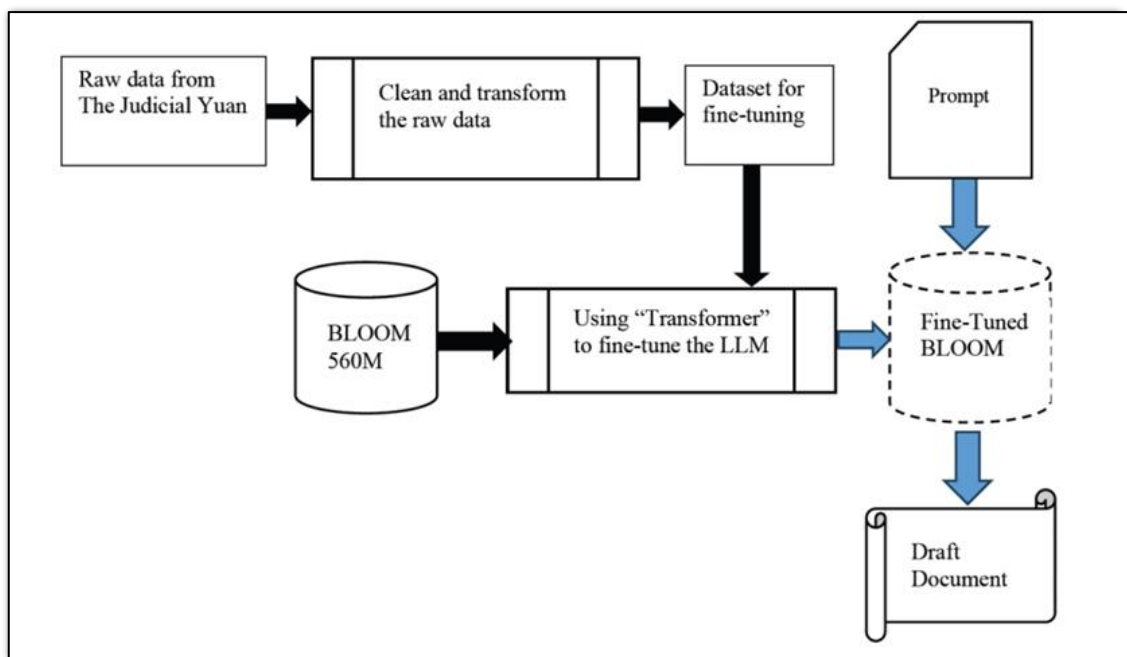
2.2.2.4 Document Drafting with LLM

Μια παρόμοια δουλειά πραγματοποίησαν και στο Εθνικό Πανεπιστήμιο της Ταϊβάν όπου χρησιμοποίησαν ένα έτοιμο large language model και το έκαναν fine tune σε νομικά έγγραφα. Τα έγγραφα που χρησιμοποίησαν εστίαζαν κυρίως σε ποινικές υποθέσεις απάτης και συγκεκριμένα περιορίστηκαν στην περιγραφή του περιστατικού τέτοιων υποθέσεων και όχι σε όλη την υπόθεση. Η απόφαση αυτή ήταν λόγω της καθαρής δομής που έχει η περιγραφή ενός περιστατικού, η οποία καθιστά τη διαδικασία της εκπαίδευσης ευκολότερη.

Στη συγκεκριμένη περίπτωση οι ερευνητές εστίασαν περισσότερο στην επεξεργασία δεδομένων αφού χώρισαν το dataset τους σε μέρη, όπου το 80% χρησιμοποιήθηκε για εκπαίδευση και το υπόλοιπο 20% για επαλήθευση και έλεγχο.

Το μοντέλο που χρησιμοποιήσαν είναι το BLOOM. Ένας από τους κύριους λόγους επιλογής αυτού του μοντέλου είναι η γνώση του σε παραδοσιακή Κινέζικη γλώσσα, η οποία είναι και η γλώσσα των νομικών εγγράφων που χρησιμοποίησαν για την εκπαίδευσή του (βλ. Σχήμα 2.13).

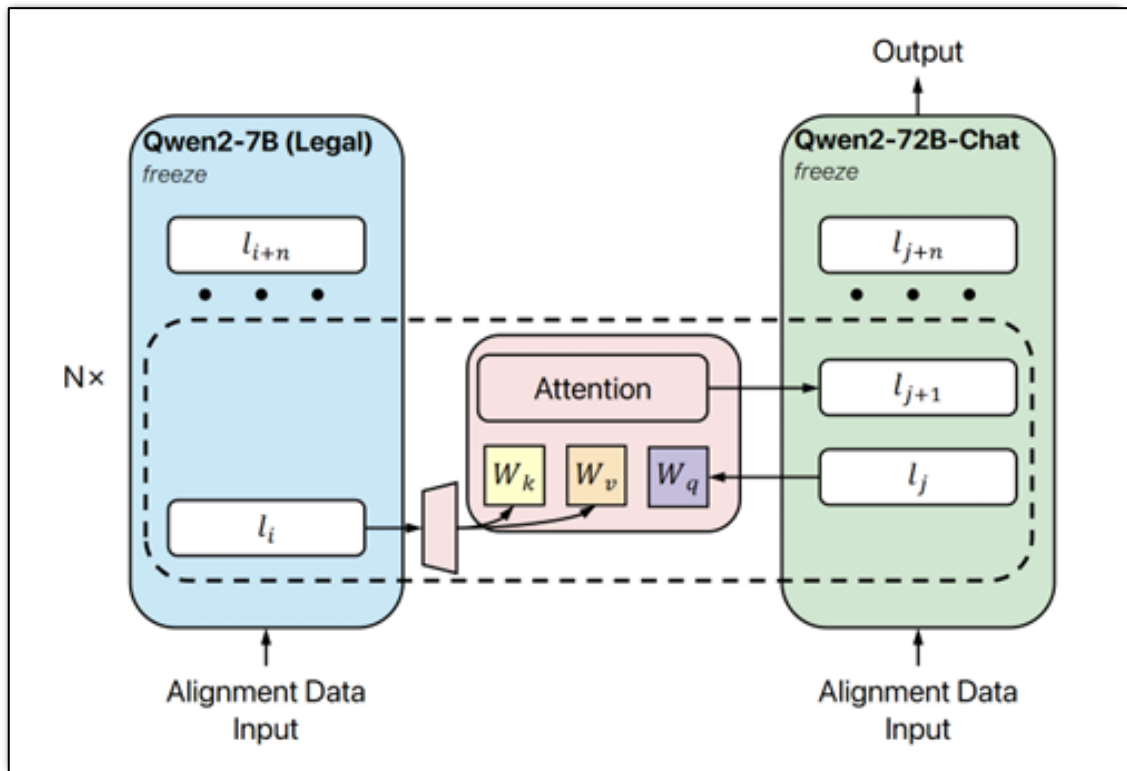
Τα αποτελέσματα της εκπαίδευσης του μοντέλου ήταν θετικά. Ακόμη, ένα κύριο πλεονέκτημα του μοντέλου που δημιουργήθηκε είναι ότι όλη η επεξεργασία δεδομένων γίνεται τοπικά και άρα δεν τίθεται ζήτημα παραβίασης ή διαρροής προσωπικών δεδομένων [5].



Σχήμα 2.13: Διαδικασία Fine Tuning και Δημιουργία Προσχέδιων [5]

2.2.2.5 Law-Neo

Ακόμη ένα παρόμοιο μοντέλο αποτελεί το Law-Neo το οποίο έχει ως βάση το μοντέλο Qwen2-72B. Η αρχιτεκτονική του συστήματος χωρίζεται σε τρία κύρια μέρη. Το βασικό μοντέλο το οποίο προαναφέρθηκε, το μοντέλο Qwen2-72B το οποίο χρησιμοποιείται ως γέφυρα, και το κομμάτι όπου πραγματοποιείται η ενοποίηση των δεδομένων (βλ. Σχήμα 2.14).



Σχήμα 2.14: Αρχιτεκτονική του Law-Neo [67]

Στο βασικό μοντέλο έγινε η ετοιμασία των δεδομένων και η εκπαίδευση του μοντέλου. Για την ετοιμασία δεδομένων χρησιμοποιήθηκαν τόσο έτοιμα δεδομένα από το διαδίκτυο, όσο και νομικά βιβλία. Για την εκπαίδευση του μοντέλου, χρησιμοποιήθηκε η γνώση που είχε πριν το μοντέλο μαζί με τα νέα δεδομένα.

Ενδιάμεσα, το μοντέλο συνδυάζει δύο τρόπους για να κάνει ενοποίηση των δεδομένων. Αρχικά, παίρνει μια απάντηση κανονικά από το βασικό μοντέλο. Έπειτα, παίρνει απαντήσεις από τα νέα δεδομένα που απέκτησε. Τέλος, συνδυάζει τα δύο και παράγει απάντηση. Στον τελικό έλεγχο που πραγματοποιήθηκε στο μοντέλο, φάνηκε ότι απαντούσε με μεγάλη ακρίβεια [67].

Κεφάλαιο 3

Μηχανισμοί για Συμπερίληψη Προσωπικών Δεδομένων σε LLMs

<i>3.1 Εκπαίδευση με Γεωγραφικά Δεδομένα.....</i>	<i>29</i>
<i>3.2 Βήματα για Βελτιστοποίηση</i>	<i>32</i>
<i>3.3 Αποθήκευση στη ChromaDB.....</i>	<i>38</i>
<i>3.4 Ανάκτηση Πληροφορίας με RAG.....</i>	<i>45</i>
<i>3.5 Αξιολόγηση Μεθόδου RAG</i>	<i>48</i>
<i>3.6 Προβλήματα με RAG.....</i>	<i>57</i>

3.1 Εκπαίδευση με Γεωγραφικά Δεδομένα

Η αρχική ιδέα διέφερε πολύ από το τελικό αποτέλεσμα. Η γενική ιδέα ήταν από την αρχή η εκπαίδευση ενός Large Language Model με δεδομένα από το διαδίκτυο. Τα δεδομένα όμως που θα χρησιμοποιούνταν για την εκπαίδευση του μοντέλου δεν ήταν πάντα νομικά ή άλλα έγγραφα που αφορούν το Πανεπιστήμιο Κύπρου.

Αρχικά η ιδέα ήταν η εκπαίδευση ενός μοντέλου με δεδομένα από το Open Street Map. Το Open Street Map είναι ένα open-source δωρεάν σύστημα χαρτών. Σε αυτό μπορεί όποιος θέλει να προσθέσει τοποθεσίες και στοιχεία για αυτές. Τα δεδομένα του Open Street Map συχνά χρησιμοποιούνται για την ανάπτυξη εφαρμογών που χρησιμοποιούν τοποθεσία ή έχουν δυνατότητα καθοδήγησης του χρήστη.

Μέσω του Open Street Map είναι δυνατή η εξόρυξη δεδομένων. Τα δεδομένα μπορούν να αποθηκευτούν σε μορφή αρχείου xml. Το xml αποτελεί ακρωνύμιο των λέξεων Extensible Markup Language και είναι ένα είδος μορφής κειμένου το οποίο χρησιμοποιείται για την περιγραφή ηλεκτρονικού περιεχομένου [20].

Απόσπασμα από xml αρχείο κατεβασμένο από το Open Street Map [41]

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.9.3 (987892 spike-
08.openstreetmap.org)" copyright="OpenStreetMap and contributors"
attribution="http://www.openstreetmap.org/copyright"
license="http://opendatacommons.org/licenses/odbl/1-0/">
  <bounds minlat="35.1435500" minlon="33.4077300" maxlat="35.1465100"
maxlon="33.4127500"/>
  <node id="33756624" visible="true" version="15" changeset="133422858"
timestamp="2023-03-07T23:38:27Z" user="SunnyIsland10" uid="17969488"
lat="35.1474342" lon="33.4157984"/>
  <node id="36881086" visible="true" version="11" changeset="133448225"
timestamp="2023-03-08T16:43:41Z" user="SunnyIsland10" uid="17969488"
lat="35.1464242" lon="33.4098643">
    <tag k="highway" v="crossing"/>
  </node>
  <node id="36881088" visible="true" version="11" changeset="133448225"
timestamp="2023-03-08T16:43:41Z" user="SunnyIsland10" uid="17969488"
lat="35.1463739" lon="33.4099129"/>
  <node id="36881666" visible="true" version="18" changeset="144669070"
timestamp="2023-12-01T17:07:52Z" user="Layft Greater" uid="13796022"
lat="35.1446072" lon="33.4073435">

  <way id="24273073" visible="true" version="13" changeset="150625347"
timestamp="2024-04-28T15:56:28Z" user="nave88" uid="21342013">
    <nd ref="262817027"/>
    <nd ref="10860321922"/>
    <nd ref="262817028"/>
    <nd ref="262817029"/>
    <nd ref="1887912259"/>
    <nd ref="1887912255"/>
    <nd ref="1887912274"/>
    <nd ref="7671813650"/>
    <nd ref="7671813651"/>
    <nd ref="1887912275"/>
    <nd ref="10722213071"/>
    <nd ref="10722213070"/>
    <nd ref="262817030"/>
    <nd ref="262817027"/>
    <tag k="addr:city" v="Aglantzia"/>
    <tag k="addr:housename" v="0EE01"/>
    <tag k="addr:housenumber" v="0EE01"/>
    <tag k="addr:street" v="Panepistimiou"/>
    <tag k="building" v="university"/>
    <tag k="building:levels" v="4"/>
    <tag k="building:levels:underground" v="1"/>
```

```
<tag k="department" v="mathematics_and_computer_science"/>
<tag k="faculty" v="science"/>
<tag k="name" v="Department of Mathematics & Statistics and
Computer Science"/>
<tag k="name:el" v="Τμήμα Μαθηματικών & Στατιστικής και
Πληροφορικής"/>
<tag k="name:en" v="Department of Mathematics & Statistics and
Computer Science"/>
</way>
```

Όπως φαίνεται πιο πάνω ένα Open Street Map xml αρχείο έχει την πιο πάνω μορφή. Πρώτα φαίνεται ο τρόπος με τον οποίο αποθηκεύονται οι χρήστες της πλατφόρμας. Στοιχεία που είναι διακριτά για τους χρήστες είναι το όνομα τους και η τοποθεσία όπου βρίσκονταν όταν πρόσθεσαν πληροφορίες στη βάση.

Έπειτα, φαίνεται ο τρόπος με τον οποίο παρουσιάζονται στο αρχείο σημεία πάνω στον χάρτη. Είναι διακριτό ότι το πιο πάνω απόσπασμα αναφέρεται στο Κτήριο Θετικών και Εφαρμοσμένων Επιστημών 01 του Πανεπιστημίου Κύπρου. Άλλα στοιχεία που αφορούν το σημείο αυτό είναι ότι αποτελεί κτήριο. Επίσης, φαίνονται και πιο συγκεκριμένα στοιχεία όπως το ποιες επιστήμες αφορά το κτήριο αυτό. Επιπρόσθετα, οι τιμές με ετικέτα ‘ref’ που φαίνονται στο πιο πάνω κουτί, αποτελούν τιμή για την ακριβής τοποθεσία του σημείου.

Για την εκπαίδευση ενός μοντέλου πάνω σε αυτά τα δεδομένα ήταν απαραίτητη η μετατροπή τους σε κείμενο το οποίο θα ήταν εύκολα κατανοητό από το Large Language Model. Για να γίνει αυτή η μετατροπή θα έπρεπε να γίνει ανάλυση των δεδομένων από το αρχείο. Αυτό θα ήταν μια χρονοβόρα και περίπλοκη διαδικασία αφού τα αρχεία ήταν μεγάλα και με πάρα πολλές πληροφορίες.

Άρα η ιδέα για δημιουργία ενός μοντέλου με δεδομένα από το Open Street Map απορριπτική. Έτσι η αρχική ιδέα άλλαξε και πλέον η ιδέα ήταν η εκπαίδευση του μοντέλου σε νομικά δεδομένα που αφορούν το Πανεπιστήμιο Κύπρου. Σαφώς τέτοια δεδομένα θα χρειαζόνταν λιγότερο καθάρισμα και επεξεργασία για να φτάσουν σε μορφή κατανοητή για το μοντέλο.

3.2 Βήματα για Βελτιστοποίηση

Έτσι ξεκίνησε η έρευνα για την εκπαίδευση ενός μοντέλου. Στόχος ήταν τόσο η χρήση καθαρού fine tuning όσο και η χρήση Retrieval Augmented Generation μεθόδων. Οι δύο μέθοδοι αυτοί έχουν ως βασικό βήμα την απόκτηση των κατάλληλων δεδομένων για τον σκοπό που θα εκπαιδευόταν το μοντέλο.

Το στάδιο αυτό αποδείχθηκε σχετικά εύκολο αφού στο διαδίκτυο υπάρχει μια πληθώρα από νομικά έγγραφα που αφορούν το Πανεπιστήμιο Κύπρου. Πέρα από νομικά έγγραφα μαζεύτηκαν και έγγραφα με πληροφορίες φοίτησης στο Πανεπιστήμιο. Τέτοια έγγραφα αποτελούν οι Οδηγοί Σπουδών σε Μεταπτυχιακό και Προπτυχιακό επίπεδο, κανονισμοί φοίτησης, και γενικές πληροφορίες για το Πανεπιστήμιο Κύπρου [55][56][57][58][59][60][61][62].

Τα έγγραφα αυτά είναι σε μορφή PDF και είναι κατά κύριο λόγο γραμμένα στην Αγγλική γλώσσα. Ο λόγος που επιλέχτηκε η Αγγλική γλώσσα είναι ότι η πλειοψηφία των έτοιμων μοντέλων είναι καλά εκπαιδευμένα στα αγγλικά, σε αντίθεση με την ελληνική γλώσσα στην οποία κάνουν γραμματικά και συντακτικά λάθη.

Η εκπαίδευση του μοντέλου θα γινόταν σε ένα ανοιχτού κώδικα γλωσσικό μοντέλο. Τέτοια μοντέλα είναι διαθέσιμα μέσω του hugging face. Έτσι με την δημιουργία λογαριασμού στην συγκεκριμένη πλατφόρμα είναι πλέον δυνατή η δημιουργία ενός token το οποίο μπορεί να χρησιμοποιηθεί για την πρόσβαση σε τέτοια μοντέλα.

Μέσω του hugging face, οι χρήστες έχουν την δυνατότητα να εγκαταστήσουν τοπικά στον ηλεκτρονικό τους υπολογιστή όποιο μοντέλο ανοιχτού κώδικα επιθυμούν, με την προϋπόθεση ότι ο υπολογιστής διαθέτει την απαραίτητη μνήμη.

Έτσι ως αρχικό στάδιο χρησιμοποιήθηκε το Llama 3.2 με 3 δισεκατομμύρια παραμέτρους. Αυτό το μοντέλο απαιτεί σχετικά μικρή υπολογιστική μνήμη, έτσι η εγκατάσταση και λειτουργία του στον προσωπικό υπολογιστή είναι αρκετά εύκολη. Το μοντέλο αυτό έπειτα θα ήταν η βάση στην οποία θα γινόταν το fine tuning.

Πριν την εκκίνηση της διαδικασίας, μαζεύτηκαν πολλές ερωτήσεις με τις οποίες θα συγκρινόταν η ακρίβεια των απαντήσεων του μοντέλου πριν και μετά την εκπαίδευση του. Αυτές οι ερωτήσεις αφορούσαν συγκεκριμένα τις νομοθεσίες και τους κανόνες του Πανεπιστημίου Κύπρου.

Το επόμενο στάδιο είναι η δημιουργία σωστά δομημένων και ακριβή δεδομένων. Για να γίνει σωστή εκπαίδευση ενός μοντέλου χρειάζεται τα δεδομένα που θα λάβει να είναι σωστά προς το περιεχόμενο, διαφορετικά θα «μάθει» πάνω σε ψευδή πληροφορίες.

Στα πλαίσια της προσπάθειας για εκπαίδευση του μοντέλου με την χρήση της μεθόδου fine tune δημιουργήθηκαν μικρά προγράμματα τα οποία εκτελούν βήματα χρήσιμα για την εκπαίδευση του μοντέλου.

Αρχικά, χρησιμοποιήθηκε ένα πρόγραμμα το οποίο διαβάζει αρχεία και τα σπάει σε tokens και δημιουργεί ένα λεξιλόγιο με βάση αυτά. Η δημιουργία των tokens πραγματοποιείται με την χρήση της βιβλιοθήκης SentencePiece [12].

```
import os
import sentencepiece as spm
#The path of the datasets
data_dir = 'path/to/data'

#the name of the tokenizer
tokenizer_prefix = os.path.join(data_dir, 'tokenizer')

#the second argument is the name of the example txt file
combined_tokens_path = os.path.join(data_dir, 'laws_dataset.txt')

# train the SentencePiece tokenizer
print("Training SentencePiece tokenizer...")
spm.SentencePieceTrainer.Train(
    input=combined_tokens_path,
    model_prefix=tokenizer_prefix,
    # changed value to 127 for OSM.txt
    vocab_size=1566,
    character_coverage=1.0,
    model_type='unigram'
```

```
)
print(f"Tokenizer trained with vocabulary size: 1566")
```

Στη συνέχεια, για σκοπούς δοκιμής δημιουργήθηκε ένας άλλος κώδικας ο οποίος διαβάζει έτοιμα token ids και τα σπάζει σε ακολουθίες όπου η μια αποτελεί την είσοδο και η άλλη τον στόχο. Η ακολουθία εισόδου διαφέρει κατά ένα token από την ακολουθία στόχου. Αυτές οι ακολουθίες χρησιμοποιούνται για την εκπαίδευση του μοντέλου στο να προβλέπει την επόμενη λέξη. Η δημιουργία των ακολουθιών πραγματοποιείται με την χρήση της βιβλιοθήκης PyTorch [45].

```
import torch
from torch.utils.data import Dataset

class LanguageModelDataset(Dataset):
    def __init__(self, token_ids, seq_length):
        self.token_ids = token_ids
        self.seq_length = seq_length

    def __len__(self):
        # Ensure the length is valid for the sequence size
        return len(self.token_ids) - self.seq_length

    def __getitem__(self, idx):
        # Handle the token slicing and return input-target pairs
        input_ids = torch.tensor(self.token_ids[idx:idx+self.seq_length],
dtype=torch.long)
        target_ids =
torch.tensor(self.token_ids[idx+1:idx+self.seq_length+1],
dtype=torch.long)
        return input_ids, target_ids
```

Ακολούθως, ένα άλλο πρόγραμμα καθορίζει ένα Transformer μοντέλο το οποίο περιέχει το στρώμα ενσωμάτωσης το οποίο μετατρέπει τα IDs σε διανύσματα, το στρώμα όπου υπάρχουν οι transformer κωδικοποιητές (encoders) [45] οι οποίοι βοηθούν το μοντέλο να κατανοήσει την σημασία των λέξεων, και μια τελική στρώση η οποία προβλέπει την επόμενη λέξη.

```
import torch.nn as nn
```

```

class SmallLanguageModel(nn.Module):
    def __init__(self, vocab_size, embed_size=128, num_heads=2,
hidden_dim=256, num_layers=2):
        super(SmallLanguageModel, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size)

        # Define a Transformer Encoder layer
        encoder_layer = nn.TransformerEncoderLayer(
            d_model=embed_size,          # Embedding size
            nhead=num_heads,             # Number of attention heads
            dim_feedforward=hidden_dim,  # Hidden size in feedforward
network
            batch_first=True             # Ensures batch comes first in
data shape
        )

        # Transformer encoder stack
        self.transformer = nn.TransformerEncoder(
            encoder_layer,
            num_layers=num_layers       # Number of transformer layers
        )

        # Output layer to map back to vocabulary size
        self.fc_out = nn.Linear(embed_size, vocab_size)

    def forward(self, src):
        # src shape: [batch_size, seq_length]
        embed = self.embedding(src)        # [batch_size,
seq_length, embed_size]
        output = self.transformer(embed)   # [batch_size,
seq_length, embed_size]
        logits = self.fc_out(output)       # [batch_size,
seq_length, vocab_size]
        return logits

```

Τέλος, δημιουργήθηκε ένα πρόγραμμα με στόχο την εκπαίδευση ενός μοντέλου σε νέα δεδομένα. Το πρόγραμμα αυτό φορτώνει το μοντέλο [18] και το tokenizer, διαβάζει δεδομένα, και τα σπάζει σε τμήματα [12]. Μετά εκπαιδεύει το μοντέλο [45] με τα δεδομένα και το αποθηκεύει.

```

# Create dataset and data loader
new_dataset = LanguageModelDataset(new_data_ids, seq_length)
new_loader = DataLoader(
    new_dataset,
    batch_size=batch_size,

```

```

        shuffle=True,
        pin_memory=True,
        num_workers=num_workers
    )

    # Fine-tuning loop
    for epoch in range(fine_tune_epochs):
        model.train()
        total_loss = 0.0
        optimizer.zero_grad()

        for batch_idx, (input_ids, target_ids) in enumerate(new_loader):
            input_ids = input_ids.to(device)
            target_ids = target_ids.to(device).long()

            # Mixed precision forward pass with device type specified
            with autocast(device_type='cuda'):
                outputs = model(input_ids)
                loss = criterion(outputs.view(-1, vocab_size),
                                target_ids.view(-1))

            # Normalize loss by accumulation steps
            loss = loss / accumulation_steps

            # Mixed precision backward pass
            scaler.scale(loss).backward()

            # Perform optimizer step every accumulation_steps
            if (batch_idx + 1) % accumulation_steps == 0 or (batch_idx + 1)
            == len(new_loader):
                # Unscale gradients and perform gradient clipping
                scaler.unscale_(optimizer)
                if grad_clip is not None:
                    torch.nn.utils.clip_grad_norm_(model.parameters(),
grad_clip)
                scaler.step(optimizer)
                scaler.update()
                optimizer.zero_grad()

            total_loss += loss.item() * accumulation_steps

        # Print progress every 100 steps
        if (batch_idx + 1) % 100 == 0:
            avg_loss = total_loss / ((batch_idx + 1) * batch_size)
            print(f'Fine-tuning Epoch [{epoch+1}/{fine_tune_epochs}],
Step [{batch_idx+1}/{len(new_loader)}], Loss: {avg_loss:.4f}')

```

Η διαδικασία εκπαίδευσης του μοντέλου είναι ακριβή. Επίσης, τα δεδομένα που θα διαβαστούν για την εκπαίδευση του μοντέλου χρειάζεται να έχουν σωστή μορφή. Για ένα μοντέλο όπως αυτό που χρησιμοποιήθηκε, τα δεδομένα χρειάζεται να έχουν μορφή ερώτησης-απάντησης. Αυτή η συγκεκριμένη δομή δεδομένων ονομάζεται Alpaca Format.

Το Alpaca Format έχει την πιο κάτω μορφή:

```
[
  {
    "instruction": "Explain the concept of photosynthesis in simple terms.",
    "input": "",
    "output": "Photosynthesis is like a plant's way of making its own food. Imagine plants as tiny factories that use sunlight as their power source. They take in water from the soil through their roots and carbon dioxide from the air through tiny holes in their leaves. Then, using the energy from sunlight, they mix these ingredients together to make glucose, which is a type of sugar that plants use for food. As a bonus, this process also produces oxygen, which is released into the air for us and other animals to breathe. So, in simple terms, photosynthesis is how plants turn sunlight, water, and air into food and oxygen."
  }
]
```

Δηλαδή χρειάζονται 3 πεδία. Το πρώτο πεδίο “instruction” αφορά την ερώτηση/εντολή που θα δώσει ο χρήστης στο μοντέλο. Το δεύτερο πεδίο “input” αφορά ότι επιπλέον στοιχεία θέλει να προσθέσει ο χρήστης. Αυτό το πεδίο συχνά παραμένει άδειο. Τέλος, το τελευταίο πεδίο “output” περιέχει την αναμενόμενη απάντηση από το μοντέλο για την τιμή του πρώτου πεδίου [68].

Η μεγάλη δυσκολία της διαδικασίας της δημιουργίας των σωστών δεδομένων είναι η μετατροπή φυσικού κειμένου, το οποίο υπάρχει στα έγγραφα, σε μορφή ερωτοαπαντήσεων. Τα κείμενα είναι μεγάλα με πολλές πληροφορίες, έτσι η μηχανική μετατροπή τους σε μορφή ερωτήσεων-απαντήσεων θα οδηγούσε σε ελλιπή ή λανθασμένα αποτελέσματα.

Με την εμφάνιση αυτού του εμποδίου άρχισε η πιο εντατική μελέτη για την χρήση της μεθόδου RAG, η οποία δεν απαιτεί τόση επεξεργασία των δεδομένων. Η μέθοδος αυτή είναι εύκολο να υλοποιηθεί με την χρήση μιας διανυσματικής βάσης δεδομένων και την εξίσωση του cosine similarity τα οποία επεξηγήθηκαν στο Κεφάλαιο 2.

3.3 Αποθήκευση στη ChromaDB

Αρχικά απαιτείται η εγκατάσταση μιας διανυσματικής βάσης έτσι ώστε να μπορούν με ευκολία να αποθηκευτούν τα δεδομένα ως διανύσματα. Για τον σκοπό αυτό έγινε η εγκατάσταση της διανυσματικής βάσης ChromaDB [7].

Η Chroma είναι εύκολη στην εγκατάστασή της και την χρήση της. Επιπρόσθετα, η Chroma προσφέρει την δυνατότητα στο χρήστη να αποθηκεύσει τοπικά στον υπολογιστή τα δεδομένα έτσι ώστε να μην χρειάζεται η αποθήκευση των δεδομένων με κάθε τρέξιμο του προγράμματος.

Αφού έγινε επιτυχώς η εγκατάσταση της Chroma στον υπολογιστή, ξεκίνησε η διαδικασία της δημιουργίας των διανυσμάτων και της αποθήκευσης τους στην τοπική Chroma. Για την επίτευξη του στόχου αυτού, έγινε η χρήση του πιο κάτω Python script [43], το οποίο θα εξηγηθεί σε κομμάτια. Επίσης, ως πρώτο στάδιο μαζεύτηκαν εννέα αρχεία με μεγέθη που κυμαίνονται από 375 KB μέχρι 34 MB [55][56][57][58][59][60][61][62].

Αρχικά, γίνεται η χρήση της συνάρτησης PyPDFDirectoryLoader() η οποία ανήκει στο langchain_community module [22], το οποίο χρησιμοποιείται για την ανάγνωση PDF, φόρτωση ιστοσελίδων, αλλά και την ανάκτηση δεδομένων από διάφορα εργαλεία όπως το Google Drive. Με την χρήση της συγκεκριμένης συνάρτησης το πρόγραμμα διαβάζει αρχεία από συγκεκριμένο κατάλογο του υπολογιστή.

Το επόμενο στάδιο είναι ο διαχωρισμός του κειμένου σε πιο μικρά τμήματα. Η πιο κάτω συνάρτηση χρησιμοποιείται για τον σκοπό αυτό.

```
def split_documents(documents: list[Document]):
    text_splitter = RecursiveCharacterTextSplitter [24](
        chunk_size=1000,
        chunk_overlap=200,
        length_function=len,
        is_separator_regex=False,
    )
    # print("OK\n")
    return text_splitter.split_documents(documents)
```

Η συνάρτηση αυτή παίρνει ως είσοδο μια λίστα από αντικείμενα τύπου Document. Κάθε αντικείμενο έχει δύο παραμέτρους. Η πρώτη παράμετρος είναι μια συμβολοσειρά η οποία περιέχει το περιεχόμενο του κειμένου. Η δεύτερη παράμετρος είναι μια δομή δεδομένων η οποία αποθηκεύει μετά-πληροφορίες του αρχείου όπως το όνομα και τον αριθμό σελίδων.

Μέσα στην συνάρτηση γίνεται ο διαχωρισμός των κειμένων με βάση τις παραμέτρους που δηλώνονται [24]. Η πρώτη παράμετρος “chunk_size” δηλώνει το μέγεθος του κάθε τμήματος σε χαρακτήρες, δηλαδή στην συγκεκριμένη περίπτωση το κάθε τμήμα είναι 1000 χαρακτήρες. Η δεύτερη μεταβλητή αφορά την επικάλυψη, δηλαδή πόσοι από τους χαρακτήρες του προηγούμενου τμήματος θα είναι και στο επόμενο τμήμα.

Η επόμενη παράμετρος “length_function” χρησιμοποιεί την συνάρτηση “len” της Python. Αυτή η συνάρτηση δηλώνει ότι το μέγεθος κάθε τμήματος θα μετριέται σε χαρακτήρες. Τέλος, η τελευταία παράμετρος “is_separator_regex” αφορά τον τρόπο με τον οποίο θα γίνεται η ανάγνωση και ο διαχωρισμός των ειδικών χαρακτήρων. Ο τρόπος που γίνεται ο διαχωρισμός, είναι ότι η συνάρτηση ψάχνει χαρακτήρες όπως ["\\n\\n", "\\n", ".", " ", ""] αυτούς. Όταν η τιμή στην παράμετρο είναι “False” η συνάρτηση αναγνωρίζει τους χαρακτήρες αυτούς ως συμβολοσειρά και όχι ως κανονική έκφραση. Για παράδειγμα ο χαρακτήρας ‘.’ τώρα αναγνωρίζεται ως απλή τελεία, ενώ αν η τιμή ήταν “True” θα αναγνωρίζουν ως η τελεία των κανονικών εκφράσεων και άρα θα συμβόλιζε έναν οποιοδήποτε χαρακτήρα. Ουσιαστικά με την χρήση της παραμέτρου αυτής το κείμενο διαβάζεται σωστά από την συνάρτηση.

Τέλος, η συνάρτηση επιστρέφει μια νέα λίστα από αντικείμενα τύπου Documents τα οποία ως “page_content” έχουν τα τμήματα που δημιούργησε η συνάρτηση με βάση τις παραμέτρους που δόθηκαν, ενώ ως μετά-πληροφορίες έχουν το όνομα του εγγράφου το οποίο ανήκουν και την σελίδα την οποία βρίσκονται.

Ένα παράδειγμα για το πώς λειτουργεί η συνάρτηση έχει ως εξής. Έστω ότι η είσοδος της συνάρτησης είναι ένα αντικείμενο με την εξής δομή:

```
Document (
    page_content="This is a very long text from page 1...",
    metadata={"source": "law1.pdf", "page": 1}
)
```

Η συνάρτηση θα επιστρέψει τα εξής τμήματα από αυτό το αντικείμενο:

```
[
    Document(page_content="This is a very long text from page 1 [first
1000 characters]", metadata={"source": "law1.pdf", "page": 1}),
    Document(page_content="[characters 800-1800]", metadata={"source":
"law1.pdf", "page": 1}),
    Document(page_content="[characters 1600-end]", metadata={"source":
"law1.pdf", "page": 1}),
]
```

Με αυτή την δομή, είναι εύκολο να αποθηκευτούν τόσο τα δεδομένα που περιέχει κάθε έγγραφο σε μικρότερα τμήματα, όσο και η πηγή που ανήκουν τα τμήματα αυτά.

Το τελευταίο βήμα είναι αποθήκευση των τμημάτων που σχηματίστηκαν στο προηγούμενο στάδιο, στην βάση. Αυτό πραγματοποιείται με την χρήση της συνάρτησης “add_to_chroma” η οποία παίρνει ως είσοδο τα τμήματα των κειμένων.

Το πρώτο βήμα είναι η αρχικοποίηση ή η φόρτωση της βάσης αν αυτή υπάρχει ήδη. Αυτό γίνεται με τον εξής τρόπο:

```
db = Chroma [7](
    persist_directory=CHROMA_PATH, embedding_function =
get_embedding_function()
)
```

Πρώτα δηλώνεται το μονοπάτι στο οποίο βρίσκεται η βάση, το οποίο όπως προαναφέρθηκε μπορεί να είναι τοπικά στον υπολογιστή. Έπειτα, για να λειτουργήσει η μετατροπή του κειμένου σε διανύσματα χρειάζεται η χρήση ενός μοντέλου ενσωμάτωσης. Για την αρχικοποίηση του μοντέλου αυτού καλείται μια βοηθητική κλάση η οποία έχει την συνάρτηση “get_embedding_function”. Η δομή της κλάσης αυτής είναι η ακόλουθη:

```
from langchain_ollama import OllamaEmbeddings [22][40]

def get_embedding_function():
    embeddings = OllamaEmbeddings(model="nomic-embed-text") [38]
    return embeddings
```

Για την χρήση του μοντέλου ενσωμάτωσης χρειάζεται το εν λόγω μοντέλο να είναι αποθηκευμένο τοπικά στον υπολογιστή. Αυτό γίνεται εύκολα με την χρήση του Ollama [38][39]. Η εγκατάστασή του γίνεται με την χρήση της εντολής “ollama pull” όπως ακριβώς και η εγκατάσταση οποιουδήποτε ollama μοντέλου [40].

Το μοντέλο αυτό πρέπει να συνδεθεί με τον κώδικα. Έτσι γίνεται χρήση του εργαλείου langchain_ollama [23] το οποίο αναλαμβάνει την επικοινωνία μεταξύ του κώδικα και του μοντέλου με ελάχιστη πολυπλοκότητα από την πλευρά του χρήστη.

Στη συνέχεια γίνεται αρχικοποίηση του μοντέλου ενσωμάτωσης, συγκεκριμένα το μοντέλο ενσωμάτωσης είναι το “nomic-embed-text” [38]. Η δουλειά του μοντέλου αυτού είναι να μετατρέψει τα κομμάτια του κειμένου που δημιουργήθηκαν στο προηγούμενο στάδιο, σε αριθμητικά διανύσματα δεκαδικών αριθμών.

Ο τρόπος που κάνει την μετατροπή το μοντέλο δεν φαίνεται στον κώδικα αλλά γίνεται με τέσσερα πολύ απλά βήματα. Αρχικά, σπάζει τις προτάσεις σε λέξεις. Έπειτα, με την χρήση της γνώσης του μοντέλου κάθε λέξη αντιστοιχίζεται σε ένα διάνυσμα, το οποίο

αναπαρίσταται ως μια λίστα δεκαδικών αριθμών. Τα διανύσματα αυτά έχουν εννοιολογική σημασία, δηλαδή λέξεις που έχουν παρόμοιο νόημα έχουν και παρόμοια διανύσματα. Με αυτό τον τρόπο το σύστημα κατανοεί το νόημα και όχι απλά τις λέξεις. Στη συνέχεια, το μοντέλο συνδυάζει όλα τα διανύσματα της κάθε λέξης σε ένα μοναδικό διάνυσμα για όλο το κομμάτι κειμένου [38].

Το αποτέλεσμα είναι για το κάθε κομμάτι κειμένου να υπάρχει το αντίστοιχο διάνυσμα. Όλα τα διανύσματα έχουν τις ίδιες διαστάσεις. Στο τέλος, η συνάρτηση αυτή επιστρέφει τα διανύσματα που δημιουργήθηκαν.

Αφού δημιουργηθεί η βάση και υπάρχουν τα διανύσματα, η συνάρτηση “add_to_chroma” συνεχίζει καλώντας μια βοηθητική συνάρτηση η οποία υπολογίζει μια μοναδική ταυτότητα για κάθε κομμάτι κειμένου.

```
def calculate_chunk_ids(chunks):  
  
    # This will create IDs like "data/law.pdf:6:2"  
    # Page Source : Page Number : Chunk Index  
  
    last_page_id = None  
    current_chunk_index = 0  
  
    for chunk in chunks:  
        source = chunk.metadata.get("source")  
        page = chunk.metadata.get("page")  
        current_page_id = f"{source}:{page}"  
  
        # If the page ID is the same as the last one, increment the  
index.  
        if current_page_id == last_page_id:  
            current_chunk_index += 1  
        else:  
            current_chunk_index = 0  
  
        # Calculate the chunk ID.  
        chunk_id = f"{current_page_id}:{current_chunk_index}"  
        last_page_id = current_page_id  
  
        # Add it to the page meta-data.  
        chunk.metadata["id"] = chunk_id
```

Η πιο πάνω συνάρτηση χρησιμοποιείται για τον υπολογισμό της μοναδικής ταυτότητας του κάθε τμήματος κειμένου. Η μοναδική αυτή ταυτότητα υποδεικνύει το όνομα του αρχείου, την σελίδα, και το τμήμα της σελίδας από το οποίο προέρχεται το τμήμα κειμένου.

Αρχικά, η συνάρτηση αυτή παίρνει ως είσοδο όλα τα κομμάτια του κειμένου που δημιουργήθηκαν. Επιπρόσθετα, δημιουργεί δύο βοηθητικές μεταβλητές, η μια υποδηλώνει την ταυτότητα του προηγούμενου κειμένου, ενώ η άλλη λειτουργεί ως δείκτης.

Η συνάρτηση αρχίζει να διατρέχει όλα τα τμήματα ένα προς ένα. Για κάθε τμήμα χρησιμοποιεί τα μετά-δεδομένα που αποθηκεύτηκαν για αυτό για να μάθει το όνομα, και την σελίδα του αρχείου. Αυτά τα δύο στοιχεία τα αποθηκεύει μαζί σε μια μεταβλητή.

Στη συνέχεια, πρέπει να βρει το τμήμα της σελίδας στην οποία ανήκει το κείμενο. Για να το κάνει αυτό χρησιμοποιεί την μεταβλητή στην οποία αποθηκεύει την σελίδα στην οποία ανήκει το προηγούμενο τμήμα. Αν η σελίδα αυτή είναι ίδια με την τρέχουσα, τότε αυξάνει κατά ένα την μεταβλητή που λειτουργεί ως δείκτη. Η νέα τιμή αποτελεί το μέρος της σελίδας στο οποίο ανήκει το κείμενο, αν οι σελίδες διαφέρουν τότε η τιμή αυτή είναι ίση με μηδέν.

Στο τέλος συνδυάζει το όνομα, τη σελίδα, και το μέρος του αρχείου για να δημιουργήσει την ταυτότητα του τμήματος. Την ταυτότητα αυτή την αποθηκεύει ως μετά-δεδομένο του τμήματος. Τέλος αποθηκεύει την τρέχουσα σελίδα στην μεταβλητή που δηλώνει την προηγούμενη σελίδα. Η διαδικασία αυτή επαναλαμβάνεται μέχρι όλα τα τμήματα να αποκτήσουν μια μοναδική ταυτότητα. Με το τέλος της επανάληψης, επιστρέφει όλα τα τμήματα στην συνάρτηση.

Αφού όλα τα τμήματα αποκτήσουν μοναδικές ταυτότητες, η συνάρτηση “add_to_chroma” συνεχίζει με το να βρει τα τμήματα τα οποία είναι ήδη αποθηκευμένα

στην βάση έτσι ώστε να αποφευχθούν τα διπλότυπα. Ο έλεγχος αυτός γίνεται με την χρήση των ταυτοτήτων όπως φαίνεται στον πιο κάτω κώδικα:

```
# Add or Update the documents.
existing_items = db.get(include=[][7]) # IDs are always included by default
existing_ids = set(existing_items["ids"])
print(f"Number of existing documents in DB: {len(existing_ids)}")
```

Ουσιαστικά βρίσκει τις ταυτότητες τις οποίες ήδη υπάρχουν στην βάση και εκτυπώνει τον αριθμό αυτό. Μόλις βρει τις ταυτότητες των κειμένων που υπάρχουν ήδη στην βάση, θα τις συγκρίνει με τις ταυτότητες οι οποίες δημιουργήθηκαν στο προηγούμενο βήμα, για να αποθηκεύσει τις νέες στην βάση.

```
# Only add documents that don't exist in the DB.
new_chunks = []
for chunk in chunks_with_ids:
    if chunk.metadata["id"] not in existing_ids:
        new_chunks.append(chunk)

if len(new_chunks):
    print(f"Adding new documents: {len(new_chunks)}")
    new_chunk_ids = [chunk.metadata["id"] for chunk in new_chunks]
    db.add_documents(new_chunks, ids=new_chunk_ids)
    print("Documents added successfully.")
else:
    print("No new documents to add")
```

Για την εισδοχή των δεδομένων, αρχικά δημιουργείται μια άδεια λίστα. Κάθε φορά που εμφανίζεται μια νέα ταυτότητα, προσθέτει το τμήμα στην λίστα αυτή. Όταν όλα τα νέα κείμενα προστεθούν στη λίστα, τα προσθέτει ένα προς ένα με τις ταυτότητες τους στην βάση. Στην περίπτωση που δεν βρεθούν νέα κομμάτια κειμένων, δεν εισάγεται τίποτα στην βάση.

Κατά τη διάρκεια εισαγωγής των δεδομένων, χρονομετρήθηκε ο χρόνος που χρειάστηκε για την εισδοχή των δεδομένων. Ο συνολικός χρόνος ήταν περίπου στα 8.5 λεπτά

3.4 Ανάκτηση Πληροφορίας με RAG

Το επόμενο βήμα μετά την αποθήκευση των δεδομένων είναι ο έλεγχος της διαδικασίας Retrieval Augmented Generation. Σκοπός του σταδίου αυτού είναι αξιολόγηση του προγράμματος αυτού τόσο στο να βρίσκει σχετικά και χρήσιμα συμφραζόμενα από τα έγγραφα, όσο και στην ικανότητα του να απαντά με φυσική γλώσσα με βάση τα συμφραζόμενα που δίνονται.

Για την εκτέλεση του πιο πάνω σταδίου χρησιμοποιείται ένα πρόγραμμα το οποίο εκτελεί τις ακόλουθες διαδικασίες. Πρώτα διαβάζει την ερώτηση του χρήστη, και έπειτα βρίσκει τα συμφραζόμενα. Στη συνέχεια, δημιουργεί ένα κείμενο εισόδου στη σωστή μορφή για το μοντέλο, και τέλος, καλεί το μοντέλο με είσοδο την εντολή που δημιούργησε. Η main συνάρτηση αυτού του κώδικα είναι η πιο κάτω.

```
def main():
    sys.stdout.reconfigure(encoding="utf-8")

    # Create CLI.
    parser = argparse.ArgumentParser() [39]
    parser.add_argument("query_text", type=str, help="The query text.")
    args = parser.parse_args()
    query_text = args.query_text
    query_rag(query_text)
```

Ουσιαστικά πρώτα διαβάζει την είσοδο του χρήστη από το τερματικό, η οποία λειτουργεί ως η ερώτηση, και έπειτα καλεί μια βοηθητική συνάρτηση η οποία αναλαμβάνει τα επόμενα βήματα. Η query_rag συνάρτηση έχει την ακόλουθη μορφή.

```
def query_rag(query_text: str):
    # Prepare the DB.
    db = Chroma [7](persist_directory=CHROMA_PATH, embedding_function =
get_embedding_function())

    # Search the DB.
    results = db.similarity_search_with_score [7](query_text, k=5)

    context_text = "\n\n---\n\n".join([doc.page_content for doc, _score
in results])
```

```

prompt_template = ChatPromptTemplate.from_template(PROMPT_TEMPLATE)
prompt = prompt_template.format(context=context_text,
question=query_text)
print(prompt)

```

```

# model = OllamaLLM(model="llama3.2")
model = OllamaLLM [39](
    model="llama3.2",
    temperature=0.5,
    max_tokens=1024
)

response_text=model.invoke(prompt)

# sources = [doc.metadata.get("id", None) for doc, _score in results]
sources = [doc.metadata.get("id", "").replace("\\", "/").split("/")[-
1] for doc, _score in results]
formatted_response = f"Response: {response_text}\nSources: {sources}"
print(formatted_response)
return response_text

```

Αρχικά, η συνάρτηση φορτώνει την βάση, με τα δεδομένα της, η οποία είναι αποθηκευμένη στο δίσκο. Ταυτόχρονα με την βάση, φορτώνει επίσης και την συνάρτηση ενσωμάτωσης η οποία χρησιμοποιήθηκε και κατά την διάρκεια της προσθήκης δεδομένων.

Ακολούθως, η συνάρτηση χρησιμοποιεί μια έτοιμη λειτουργία της ChromaDB, την “similarity_search_with_score”. Αυτή η λειτουργία παίρνει ως είσοδο την ερώτηση του χρήστη και ένα αριθμό k, ο οποίος δηλώνει πόσα στοιχεία θα επιστρέψει.[7]

Η διαδικασία που ακολουθείται είναι παρόμοια με αυτή που χρησιμοποιήθηκε για την δημιουργία των διανυσμάτων για κάθε κομμάτι κειμένου. Πρώτα, μετατρέπει την ερώτηση του χρήστη σε διάνυσμα, με ακριβώς τον ίδιο τρόπο με τον οποίο μετέτρεψε και το κείμενο σε διάνυσμα. Έπειτα, συγκρίνει το διάνυσμα της ερώτησης με όλα τα διανύσματα που είναι αποθηκευμένα στην βάση. Η σύγκριση γίνεται με την χρήση της εξίσωσης “cosine similarity” [65] η οποία επεξηγήθηκε στο κεφάλαιο 2. Στο τέλος, αφού πραγματοποιηθούν όλοι οι έλεγχοι, η συνάρτηση επιστρέφει τα k πιο κοντινά

αποτελέσματα. Τα αποτελέσματα αυτά είναι δυάδες όπου η πρώτη τιμή είναι κομμάτι του κειμένου, και η δεύτερη είναι ο βαθμός ομοιότητας.

Στη συνέχεια, η συνάρτηση ενώνει τα αποτελέσματα μαζί ώστε να μπορεί να τα βάλει στη τελική είσοδο που θα στείλει στο μοντέλο. Ακολουθώς, συμπληρώνει το προκαθορισμένο πρότυπο το οποίο ακολουθεί η είσοδος που λαμβάνει το μοντέλο. Το πρότυπο αυτό δηλώνεται ως σταθερά στην αρχή του κώδικα και έχει την ακόλουθη μορφή:

```
PROMPT_TEMPLATE = """
Answer the question based only on the following context:

{context}

---

Answer the question based on the above context: {question}
"""
```

Όπως φαίνεται πιο πάνω, στο πρότυπο αυτό υπάρχουν προκαθορισμένοι χώροι για το κάθε μέρος του κειμένου εισόδου. Στο χώρο “context” μπαίνουν τα συνενωμένα συμφραζόμενα που βρέθηκαν πριν, ενώ στο πεδίο “question” μπαίνει η ερώτηση που έβαλε ο χρήστης.

Έπειτα, η συνάρτηση καλεί το τοπικά εγκατεστημένο μοντέλο, το οποίο όπως προαναφέρθηκε είναι το “llama3.2” [39]. Ακόμη, με την κλήση του μοντέλου χρησιμοποιούνται επιπλέον δύο παράμετροι. Η πρώτη δηλώνει πόσο δημιουργικό θα είναι το μοντέλο. Η μεταβλητή αυτή κυμαίνεται από 0 μέχρι 1, όσο πιο κοντά στο 0 βρίσκεται, τόσο λιγότερο εξερευνά απαντήσεις το μοντέλο. Η δεύτερη μεταβλητή δηλώνει το μέγιστο όριο χαρακτήρων της απάντησης που θα δώσει το μοντέλο. Στο μοντέλο αυτό στέλνει το prompt στην τελική του μορφή. Έπειτα παίρνει την απάντηση του μοντέλου και την αποθηκεύει.

Αφού βρει την απάντηση του μοντέλου, χρησιμοποιώντας τα μετά-δεδομένα των κομματιών του κειμένου που χρησιμοποιήθηκαν, βρίσκει τις πηγές, δηλαδή την

ταυτότητα του τμήματος αυτού όπως αυτή αποθηκεύτηκε κατά την εισαγωγή δεδομένων στην βάση. Στο τέλος, τυπώνει την απάντηση του μοντέλου, μαζί με τις πηγές.

Πριν αρχίσει η εισδοχή δεδομένων στην βάση, πρέπει να ξεκινήσει μικρός server στον υπολογιστή έτσι ώστε να είναι δυνατή η επικοινωνία του προγράμματος με τα μοντέλα. Με αυτό τον τρόπο η επικοινωνία γίνεται τοπικά χωρίς την ανάγκη για σύνδεση στο διαδίκτυο. Η ενεργοποίηση του server πραγματοποιείται με την εντολή “ollama serve”.

Αφού γίνει η καθιέρωση της σύνδεσης, μπορούν τόσο ο κώδικας για την εισδοχή δεδομένων στην βάση, όσο και ο κώδικας για την χρήση του μοντέλου να τρέξουν.

3.5 Αξιολόγηση Μεθόδου RAG

Με την ολοκλήρωση της διαδικασίας για την προσθήκη των διανυσμάτων στην βάση, ξεκίνησε η διαδικασία ελέγχου της λειτουργίας και αξιολόγησης του προγράμματος. Ο έλεγχος έγινε με την εκτέλεση του δεύτερου προγράμματος με είσοδο ερωτήσεις που αφορούν τα έγγραφα τα οποία αποθηκεύτηκαν στην βάση.

Για να συγκριθεί η αύξηση της ακρίβειας των απαντήσεων του μοντέλου, ερωτήθηκαν οι ίδιες ερωτήσεις στο μοντέλο χωρίς να μεσολαβεί η διαδικασία RAG αλλά και με την χρήση της διαδικασίας RAG [49].

Πιο κάτω φαίνονται τα αποτελέσματα όσον αφορά τις απαντήσεις του ίδιου μοντέλου χωρίς και με RAG.

Ερώτηση 1: What are the official languages of University of Cyprus?

Απάντηση χωρίς RAG:

The official languages of the University of Cyprus are Greek and English. According to 2.5the university's website, both languages are used in academic and administrative contexts.

Greek is the primary language used for teaching and research purposes, while English is widely used for international collaborations, publications, and student mobility.

Many programs at the University of Cyprus offer courses taught in both Greek and English, allowing students to choose their preferred language of instruction.

Απάντηση με RAG:

The official languages of the University of Cyprus are Greek and Turkish.

Εκτός από την απάντηση του μοντέλου, στο τερματικό τυπώνονται επίσης και τα συμφοραζόμενα που στάλθηκαν στο μοντέλο, αλλά και τα αρχεία στα οποία βρίσκονται. Για την συγκεκριμένη ερώτηση τα συμφοραζόμενα ήταν τα ακόλουθα:

the University of Cyprus, only one course of the first level of foreign language can count as an elective course.

However, students have the option of either completing one course from the first level of a foreign language or the second level of the same language. In the latter case, both levels can count as elective courses.

Foreign Language

In addition to the above courses, students must complete at least 10 ECTS in a foreign language.

OTHER PRIORITIES OF THE DEPARTMENT

- Establishing its presence in Cyprus, as well as in the rest of the Greek world and Europe. To achieve this goal, the Department actively participates in joint research projects with other universities and international organisations, such as UNESCO, the Council of Europe, the European Union and the Commonwealth. For the same purpose, the Department organises international conferences, lectures and seminars, and publishes a journal.
- Assisting and promoting school development by

14.1 The languages of teaching of the University are the official languages of the Republic of Cyprus, namely Greek and Turkish. Teaching in another language is permitted only in the cases provided for in the following paragraphs of this article:

14.1.1 The University may, upon the recommendation of the relevant Faculty and upon the decision of the Senate, to offer some of the postgraduate programmes offered in one of the teaching languages of the University and in other languages.

14.1.2 The Senate may decide, upon a recommendation from the relevant Faculty, to offer a postgraduate programme in a language other than the official languages of the Republic without offering it in one of the official languages. The postgraduate programmes requires to be approved by the University Council.

14.1.3 The University may by the decision of the Senate, offer in any language inter-university programmes.

186

LANGUAGE CENTRE

The Language Centre of the University of Cyprus provides the opportunity to UCY students, from all Faculties, as well to the general public to learn a variety of languages. In particular, it offers courses in the English, French, German, Spanish, Italian, Turkish, Russian and Chinese (in collaboration with the Confucius Institute), as well as in Cypriot Sign

Language. It also provides discipline-specific language courses to UCY students. Within this framework, learners acquire speaking, listening, reading and writing skills. In parallel, they develop the social and cultural knowledge that will enable them to integrate into the contemporary globalized social, academic or professional environment.

ACTING DIRECTOR

Fryni Kakoyianni-Doa

SPECIAL TEACHING STAFF

Elias Argyrou

Maria-Gabriella Arru

Christina Matsi-Azaïs

Rowena Burlenghi

Stefan Schlaefli Fust

Düriye Gökçebağ

Bernadette Horton-Savvides

Shaunna Joannidou

Undergraduate Studies

The academic year comprises two semesters. Eight semesters are normally required for graduation, but in special cases, the duration of studies may be extended to a maximum of twelve semesters. Additional courses are also offered during the Summer Semester. Attendance is compulsory. The languages of instruction are Greek and Turkish (the official languages as stipulated by the Constitution of the Republic of Cyprus).

Each year, around 1500 undergraduate students enter the University of Cyprus.

Currently, there are approximately 5300 undergraduate students.

Fees

The fees for Cypriot students and students admitted from

EU countries total €1.709 per semester and they are paid by the State. For foreign students, the fees total €3.417 per semester.

Pancyprian Examinations

Admission for the majority of the students that enter the University of Cyprus is based on the Pancyprian Examinations set by the Cyprus Ministry of Education,

who hold degrees from Universities where the language of instruction is not Greek, are not required to submit a Language Certificate.

More information on the Department's Graduate Programmes can be found on the following webpage:

<https://www.ucy.ac.cy/ppmg/el/>

For information about the structure, duration and institutional framework of

graduate studies at the University of Cyprus, as well as information regarding tuition fees, please consult the Graduate School's webpage:

<https://www.ucy.ac.cy/graduateschool/el/postgraduate-studies/postgraduate-studies-regulations>

<https://www.ucy.ac.cy/graduateschool/el/postgraduate-studies/fees>

Coordinator -

Programme Director

Associate Professor Marinos Pourgouris

Details of person(s)

for information on the

programme

Christiana Mylona

Telephone: 22 894302

E-mail: mylona.christiana@ucy.ac.cy

Πηγές: ['UndergraduateProspectusEN24-26.pdf:297:1'[62], 'PostGraduate-Study-Rules.pdf:27:1'[60], 'UndergraduateProspectusEN24-26.pdf:185:0'[62], 'UndergraduateProspectusEN24-26.pdf:16:0'[62], 'Agglik-i-Prokirixi-xeimerino-2025-2026.pdf:16:2'[57]]

Φαίνεται ότι τα 5 πιο σχετικά με την ερώτηση κομμάτια κειμένου έχουν όντως την απάντηση της ερώτησης την οποία έδωσε ως είσοδο ο χρήστης. Επίσης, η απάντηση που παραχώρησε το μοντέλο με την χρήση RAG είναι ορθή, σε σύγκριση με την πρώτη απάντηση, αφού με βάση τους κανόνες του Πανεπιστημίου Κύπρου, οι επίσημες γλώσσες του Πανεπιστημίου είναι τα Ελληνικά και τα Τουρκικά [47]. Πιο κάτω παρουσιάζονται περισσότερα παραδείγματα για σύγκριση.

Ερώτηση 2: What documents are required when applying for a postgraduate program at the University of Cyprus?

Απάντηση χωρίς RAG:

According to the University of Cyprus's website, when applying for a postgraduate program, applicants typically need to provide the following documents:

1. Application form: Completed and signed application form.
2. Transcripts: Official transcripts from all previous academic institutions attended, showing the grades and credits earned in relevant fields.
3. Diploma: Original or certified copy of your diploma (if you're a recent graduate) or proof of completion of your undergraduate degree.
4. Bachelor's degree certificate: Certified copy of your bachelor's degree certificate.
5. Language proficiency test results: Official scores from English language proficiency tests such as IELTS, TOEFL iBT, or Cambridge Certificate in English (CPE), depending on the program requirements.
6. Academic references: One academic reference letter from a professor or academic supervisor who can speak to your academic abilities and potential for success in the postgraduate program.
7. Personal statement: A written statement outlining your research interests, career goals, and reasons for applying to the University of Cyprus's postgraduate program.

8. Proof of nationality: Passport or national ID card that shows your nationality.

9. Passport photo: A recent passport-sized photograph.

Additional requirements:

For some programs, additional documents such as:

- + GRE/GMAT scores
- + CV (Curriculum Vitae)
- + Research proposal (for research-based programs)

Please note that specific requirements may vary depending on the program you're applying to. I recommend checking with the University of Cyprus's admissions office for the most up-to-date and accurate information.

Απάντηση με RAG:

According to the provided context, the following documents are required when applying for a postgraduate program at the University of Cyprus:

1. A Curriculum Vitae.
2. Certified copies of University degrees or confirmation of graduation (admitted candidates will need to submit copies of the certified degrees to the University of Cyprus before their registration).
3. Copies of Transcripts for all programmes of study.
4. A brief Personal Statement of personal goals and research interests (up to 2 pages).
5. Names and contact details of at least two referees (university professors) familiar with the candidate and his/her academic performance.
6. Completed application form for admission into the M.A. programme.
7. Brief CV and description of scholarly and research interests.
8. Scanned copies of university diplomas and other postgraduate diplomas (if applicable), or a verification of expected graduation in the summer or fall.

Στη πιο πάνω ερώτηση και οι δύο απαντήσεις είναι αρκετά σωστές. Ωστόσο, η απάντηση χωρίς την τεχνική RAG έχει ψευδή πληροφορία όπως ότι χρειάζεται μόνο μια συστατική επιστολή ενώ χρειάζονται το ελάχιστο δύο. Επίσης, αναφέρει ότι σε μόνο μερικά

προγράμματα επιβάλλεται η υποβολή βιογραφικού σημειώματος, ενώ είναι απαραίτητο σε όλα. Από την άλλη η απάντηση με την χρήση RAG είναι πιο ακριβής και εξηγεί τα απαραίτητα στοιχεία όπως παρουσιάζονται στα επίσημα έγγραφα του Πανεπιστημίου, παρά το γεγονός ότι επαναλαμβάνει το αντίγραφο πτυχίου. Συμπερασματικά, η απάντηση με την χρήση RAG για ακόμη μια φορά είναι πιο σωστή.

Ερώτηση 3: Under what conditions can a doctoral student transfer from another university to the University of Cyprus, and how are ECTS credits handled in such cases?

Απάντηση χωρίς RAG:

According to the University of Cyprus's website, students wishing to transfer from another university to pursue their doctoral studies at the University of Cyprus can be considered under certain conditions.

Eligibility criteria:

1. Academic background: The student must have a bachelor's or master's degree from an accredited institution recognized by the Ministry of Education and Culture.
2. Language proficiency: The student must demonstrate language proficiency in English (or Greek, depending on the program).
3. Research potential: The student must demonstrate research potential and a clear academic goal.

Transfer procedure:

1. Initial assessment: The University of Cyprus's Admissions Committee will assess the student's application, academic background, and research potential.
2. Academic evaluation: If the student is deemed eligible, their academic record will be evaluated to determine the number of credits that can be transferred.
3. Course selection: The student must select courses that align with their research interests and goals.

ECTS credits:

1. Transfer credit assessment: The University of Cyprus's Academic Registry will assess the transfer credits and determine which ones are equivalent to ECTS credits.
2. Recognition: The recognized credits will be recognized as part of the student's cumulative record at the University of Cyprus.
3. Unrecognized credits: Unrecognized credits will not be awarded as ECTS credits, but may still contribute to the student's overall credit load.

Important notes:

The transfer process is subject to the approval of the University of Cyprus's Admissions Committee and the Academic Senate.

Students who are transferring from another university must meet all the requirements set by the University of Cyprus for new admissions.

If you're a prospective doctoral student considering transferring to the University of Cyprus, I recommend consulting with the university's Admissions Office or Academic Registry for personalized guidance on the transfer process and ECTS credits.

Απάντηση με RAG:

A doctoral student can transfer from another university to the University of Cyprus under the following conditions:

1. The PhD candidate must come from an establishment and/or research institute of higher education that is officially recognised in the country where it operates.
2. Transfer candidates must have successfully completed at least two (2) semesters at their educational establishment of origin.
3. The transfer candidate must fulfil all the conditions and requirements of the doctoral programme of study of the UCY Department to which they wish to transfer.

In cases of transfers, ECTS credits are handled as follows:

* The minimum total length of studies for transferring students is six (6) semesters, with a maximum total length of sixteen (16) semesters.

* At least two (2) semesters must be completed at the University of Cyprus.

* A doctoral student transferring to UCY pays fees for the semesters of study at UCY as specified in the Postgraduate Rules.

Additionally, the transfer is subject to approval by the Department's Board and the Graduate School, and the number of places for transfers is specified by the Department's Board.

Για ακόμη μια φορά φαίνεται ότι η απάντηση από το μοντέλο που είχε ως είσοδο συμφραζόμενα με την χρήση της μεθόδου RAG, παρείχε ακριβέστερη και ορθότερη απάντηση από την απάντηση του γενικού μοντέλου. Συγκεκριμένα, το γενικό μοντέλο δεν αναφέρει ξεκάθαρα τι απαιτείται για την μεταγραφή των φοιτητών, όπως ότι πρέπει να τελείωσαν με επιτυχία τα προηγούμενα δύο εξάμηνά τους.

Συμπερασματικά, με βάση τους πιο πάνω ελέγχους που έγιναν στο ίδιο μοντέλο, με και χωρίς την χρήση της τεχνικής RAG είναι ξεκάθαρο ότι αυτή η τεχνική βοηθάει πολύ το μοντέλο στην παραγωγή καλύτερων απαντήσεων.

3.6 Προβλήματα με RAG

Πιο πάνω φαίνεται ότι η μέθοδος RAG αναβαθμίζει πολύ τις απαντήσεις από το μοντέλο. Ωστόσο, το σύστημα που αναπτύχθηκε είχε αρκετά προβλήματα τα οποία χρειάζονται βελτίωση.

Αρχικά, υπάρχει περιορισμός στην γλώσσα επικοινωνίας με τον μοντέλο. Ο τρόπος με τον οποίο λειτουργεί η διανυσματική βάση μπορεί με επιτυχία να αποθηκεύσει διανύσματα που προέρχονται από ελληνικό κείμενο. Επίσης, είναι δυνατή η αναζήτηση παρόμοιου κειμένου με την ερώτηση του χρήστη, όπως ακριβώς γίνεται και με τις αγγλικές ερωτήσεις.

Το πρόβλημα εμφανίζεται στον περιορισμό του μοντέλου να απαντήσει με σωστά ελληνικά αλλά και στον περιορισμό για πιο ακριβής εύρεση σωστών συμφραζόμενων. Για τον έλεγχο της υπόθεσης αυτής στο μοντέλο στέλνεται η ερώτηση «Ποια έγγραφα

απαιτούνται για την υποβολή αίτησης σε μεταπτυχιακό πρόγραμμα στο Πανεπιστήμιο Κύπρου;», η οποία ρωτήθηκε πιο πριν στα αγγλικά.

Το μοντέλο σε αυτή την ερώτηση απάντησε ως εξής: «Η εγκυρότητα της αίτησης και η αναφορά του προηγουμένως εγκρινού Ερευνητικού Συμβούλου είναι απαραίτητα.». Όπως φαίνεται, η απάντηση είναι λακωνική και λανθασμένη, ενώ ταυτόχρονα περιέχει και λατινικούς χαρακτήρες. Ταυτόχρονα, τα συμφραζόμενα που δόθηκαν στην ερώτηση, παρά το ότι είναι στα Ελληνικά, δεν έχουν μεγάλη συσχέτιση με την ερώτηση του χρήστη.

Για τα δύο πιο πάνω προβλήματα η πιο ξεκάθαρη λύση είναι η χρήση ενός καλύτερου μοντέλου. Το μοντέλο ενσωμάτωσης που χρησιμοποιήθηκε αφορά συγκεκριμένα την αγγλική γλώσσα. Υπάρχουν άλλα μοντέλα ενσωμάτωσης τα οποία είναι εκπαιδευμένα τόσο στα αγγλικά όσο και σε άλλες γλώσσες. Επιπρόσθετα, το γενικό μοντέλο που χρησιμοποιήθηκε για σκοπούς ερωτοαπαντήσεων όπως προαναφέρθηκε είναι πολύ μικρό. Λόγω του μικρού του μεγέθους η αδυναμία του να απαντήσει σε άλλες γλώσσες είναι λογική επίπτωση.

Ακόμη ένας τρόπος βελτιστοποίησης του συστήματος είναι να αλλάξει ο τρόπος αποθήκευσης των δεδομένων. Πιο πάνω αναφέρθηκε ότι το κείμενο χωρίζεται σε κομμάτια μεγέθους 1000 χαρακτήρων με επικάλυψη 200 χαρακτήρων κάθε φορά. Η αλλαγή αυτών των παραμέτρων, και άρα η αποθήκευση κομματιών διαφορετικού μεγέθους μπορεί να οδηγήσει σε καλύτερες απαντήσεις από το μοντέλο.

Οι πιο πάνω βελτιώσεις θα ήταν το επόμενο βήμα. Ωστόσο, το τελικό θέμα της διπλωματικής άλλαξε. Η ιδέα για την χρήση της τεχνικής RAG παραμένει με την διαφορά ότι πλέον ο κάθε χρήστης θα επιλέγει τα δεδομένα που θα είναι αποθηκευμένα, και θα υλοποιηθεί ένα online σύστημα.

Κεφάλαιο 4

Αρχιτεκτονική Συστήματος

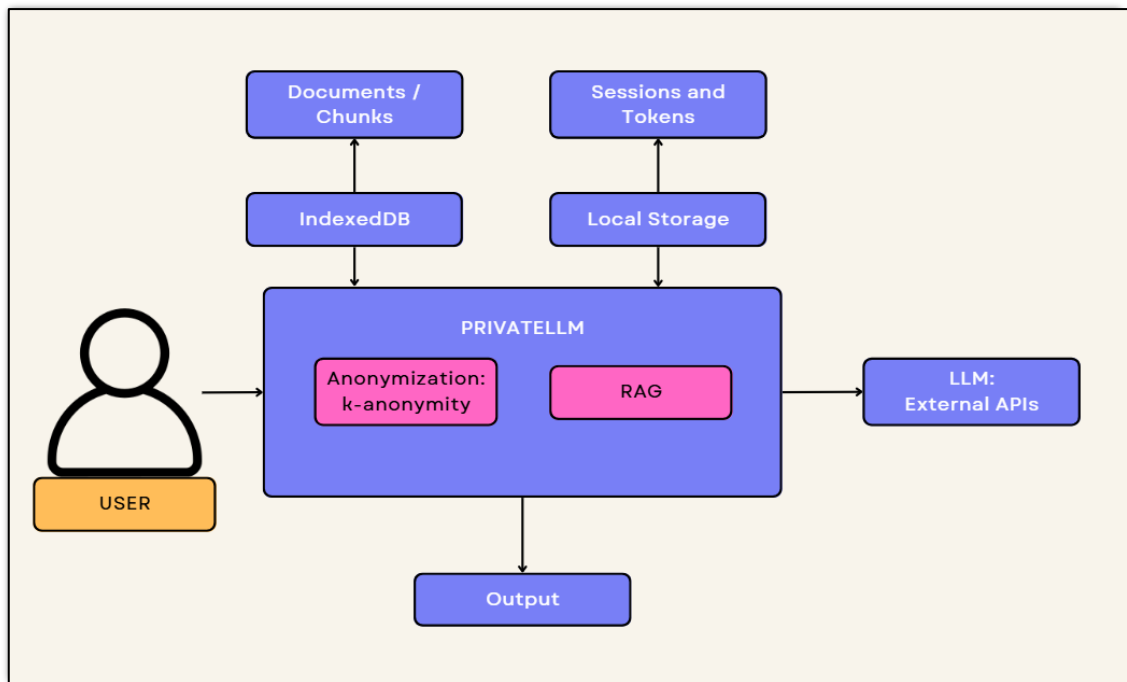
<i>4.1 Επισκόπηση Αρχιτεκτονικής</i>	<i>59</i>
<i>4.2 Αποθήκευση & Επεξεργασία Εγγράφων.....</i>	<i>61</i>
<i>4.2.1 Φόρτωση Αρχείων</i>	<i>61</i>
<i>4.2.2 Αποθήκευση Διανυσμάτων.....</i>	<i>64</i>
<i>4.2.3 Προβολή Εγγράφων.....</i>	<i>66</i>
<i>4.2.4 Διαχείριση Εγγράφων</i>	<i>69</i>
<i>4.3 Ανάλυση & Απάντηση Ερωτήσεων</i>	<i>70</i>
<i>4.3.1 Εύρεση Συμφραζόμενων</i>	<i>70</i>
<i>4.3.2 Επικοινωνία με Γλωσσικά Μοντέλα</i>	<i>74</i>
<i>4.3.3 Επεξεργασία Ερωτήματος</i>	<i>80</i>
<i>4.3.4 Μέθοδοι Προστασίας Ιδιωτικότητας & Υλοποίηση Κ-Ανωνυμίας</i>	<i>87</i>
<i>4.4 Διαχείριση Συνεδριών Χρήστη.....</i>	<i>93</i>

4.1 Επισκόπηση Αρχιτεκτονικής

Σε αυτή την ενότητα θα γίνει ανάλυση της αρχιτεκτονικής του συστήματος αλλά και τις λειτουργίες του. Το σύστημα που υλοποιήθηκε έχει ως σκοπό να ευκολύνει τους χρήστες να κατανοούν τα έγγραφα τους. Σε υψηλό επίπεδο, το σύστημα έχει ως βασική λειτουργία την αποθήκευση εγγράφων του χρήστη σε μια τοπική διανυσματική βάση στον ιστότοπο του κάθε χρήστη.

Αρχικά, το σύστημα είναι υλοποιημένο σε γλώσσες του ιστού. Πιο συγκεκριμένα οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν είναι η HTML και η JavaScript. Η αρχιτεκτονική εκτελείται καθαρά στο πρόγραμμα περιήγησης, χωρίς την ανάγκη για κάποιο server (βλ. Σχήμα 4.1).

Οι δυνατότητες του συστήματος, είναι ο διαμοιρασμός κειμένου σε μικρότερα τμήματα, η μετατροπή τους σε διανύσματα, και η αποθήκευσή τους σε τοπική βάση. Επίσης, ο χρήστης μπορεί να ρωτήσει ένα μεγάλο γλωσσικό μοντέλο της επιλογής του για ό,τι αφορά τα έγγραφα που ανέβασε. Ταυτόχρονα, το σύστημα δίνει την δυνατότητα στους χρήστες να διαχειριστούν τα δεδομένα τα οποία αποθηκεύονται στην βάση. Τέλος, η πλατφόρμα δίνει αρκετή έμφαση στην προστασία των δεδομένων μέσα στα έγγραφα με την χρήση τεχνικών όπως το k-anonymity.



Σχήμα 4.1: Αρχιτεκτονική Συστήματος

Το σύστημα λειτουργεί εξ ολοκλήρου από την πλευρά του χρήστη, επομένως δεν εφαρμόζεται ο παραδοσιακός διαχωρισμός σε επίπεδα αρχιτεκτονικής. Έτσι, στα υποκεφάλαια που ακολουθούν, θα γίνει ανάλυση των λειτουργιών του συστήματος, και πώς συντονίζονται από τα επιμέρους τμήματά του.

4.2 Αποθήκευση & Επεξεργασία Εγγράφων

Ο χρήστης έχει την δυνατότητα να ανεβάσει δικά του έγγραφα στην πλατφόρμα. Μετά, το σύστημα επεξεργάζεται αυτά τα έγγραφα ώστε να αποθηκεύσει σε διανυσματική μορφή κομμάτια από το κείμενο που βρίσκεται στο έγγραφο. Επιπρόσθετα, ο χρήστης μπορεί να δει τα έγγραφα του μαζί με τα αντιπροσωπευτικά τμήματα. Τέλος, ο χρήστης είναι ικανός να διαγράψει τα έγγραφα ή τμήματά του. Οι λειτουργίες που αφορούν την αποθήκευση, προβολή, και των εγγράφων θα αναλυθούν στις επόμενες υποενότητες.

4.2.1 Φόρτωση Αρχείων

Μια από τις κύριες λειτουργίες του συστήματος είναι το ανέβασμα αρχείων από τον χρήστη στην ιστοσελίδα. Ο χρήστης έχει την δυνατότητα να επιλέξει PDFs από τον υπολογιστή του και να τα αποθηκεύσει στην τοπική βάση δεδομένων εντός του προγράμματος περιήγησης.

Με την βοήθεια κουμπιών Bootstrap [3][4], η διεπαφή του χρήστη είναι εύκολη στην κατανόηση. Όταν ο χρήστης πατάει το κουμπί “Upload” (βλ. Σχήμα 4.2) καλείται από την HTML η συνάρτηση `startUpload()` η οποία βρίσκεται στο JavaScript αρχείο.



Upload PDF file

Choose Files No file chosen Upload

Session: Session 1

Σχήμα 4.2: Διεπαφή για Ανέβασμα Αρχείων

Αρχικά η συνάρτηση καλεί μια JavaScript βιβλιοθήκη, την PDF.js η οποία είναι υπεύθυνη για την ανάγνωση PDF αρχείων άμεσα στον ιστότοπο. Ταυτόχρονα, η βιβλιοθήκη αυτή αναθέτει σε ένα Web Worker την εργασία για ανάγνωση των αρχείων, έτσι ώστε να κρατήσει την διεπαφή ομαλή για τον χρήστη και χωρίς καθυστερήσεις. Πιο

κάτω εμφανίζεται ο τρόπος με τον οποίο καλείται η βιβλιοθήκη. Ακόμη, φαίνεται ότι ενημερώνει τον χρήστη για τον αριθμό των αρχείων τα οποία επεξεργάζεται.

```
pdfjsLib.GlobalWorkerOptions.workerSrc =  
  "https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.4.120/pdf.worker.min  
.js";  
document.getElementById(  
  "status"  
)<div data-bbox="161 296 889 437" data-label="Text">

Αφού ολοκληρωθεί η αρχικοποίηση του Web Worker, θα ξεκινήσει η διαδικασία της ανάγνωσης και του διαχωρισμού του κάθε κειμένου σε μικρότερα τμήματα. Αρχικά, γίνεται έλεγχος αν το αρχείο είναι ήδη αποθηκευμένο στην βάση, και σε περίπτωση που υπάρχει, προειδοποιεί τον χρήστη ότι θα γίνει αντικατάσταση του. Στη συνέχεια, γίνεται η ανάγνωση σε δυαδική μορφή του κάθε αρχείου ξεχωριστά, και έπειτα τα μετατρέπει σε Unit8Array ώστε να μπορεί να τα διαβάσει το PDF.js [35].


```

Με την ανάγνωση ενός αρχείου ξεκινάει η επεξεργασία του κειμένου του ώστε να μπορεί να το διαχωριστεί σωστά το σύστημα. Πρώτα, δημιουργείται μια συμβολοσειρά η οποία περιέχει ολόκληρο το κείμενο, από όλες τις σελίδες του αρχείου. Αυτό επιτυγχάνεται με την χρήση ενός απλού βρόγχου επανάληψης. Έπειτα, με δυναμικό τρόπο επιλέγονται οι μεταβλητές που υποδηλώνουν το μέγεθος του κάθε τμήματος αλλά και η επικάλυψη που θα έχουν τα τμήματα μεταξύ τους. Όσο πιο μεγάλο το κείμενο, τόσο πιο μεγάλο μέγεθος και επικάλυψη θα έχουν τα τμήματά του.

Αφού οριστούν οι πιο πάνω μεταβλητές, ξεκινάει ο διαχωρισμός του κειμένου σε μικρότερα τμήματα. Ο διαχωρισμός επιτυγχάνεται με την χρήση της συνάρτησης `splitIntoChunks` η οποία εμφανίζεται πιο κάτω.

```
function splitIntoChunks(text, size, overlap) {  
  const chunks = [];  
  for (let i = 0; i < text.length; i += size - overlap) {  
    const chunk = text.slice(i, i + size);  
    if (chunk.length > 0) {  
      chunks.push(chunk);  
    }  
  }  
}
```

```

    }
  }
  return chunks;
}

```

Η συνάρτηση αυτή παίρνει ως είσοδο τις μεταβλητές μεγέθους και επικάλυψης οι οποίες υπολογίστηκαν στο προηγούμενο στάδιο, αλλά και ολόκληρο το κείμενο. Η συνάρτηση αυτή διαβάζει το κείμενο σε τμήματα μεγέθους “size” χαρακτήρων και προχωρώντας “size-overlap” χαρακτήρες κάθε φορά. Σε κάθε βήμα χωρίζει το κείμενο με βάση το παράθυρο ανάγνωσης του κάθε βήματος με την βοήθεια της συνάρτησης “split” της Python [44]. Ταυτόχρονα ελέγχει ότι δεν θα υπάρχει άδειο κομμάτι κειμένου αποθηκευμένο. Έπειτα, προσθέτει το κομμάτι που υπολογίστηκε σε ένα πίνακα. Με το τέλος της ανάγνωσης του κειμένου, επιστρέφει τον τελικό πίνακα.

Ο πίνακας με τα κομμάτια κειμένου επιστρέφεται στην συνάρτηση startUpload(), η οποία επιστρέφει στον χρήστη, μέσω της διεπαφής του χρήστη, τον αριθμό των τμημάτων που δημιουργήθηκαν για το συγκεκριμένο αρχείο το οποίο επεξεργάζεται.

Αφού πλέον υπάρχουν τα τμήματα για το αρχείο πρέπει να γίνει η ενσωμάτωσή τους σε διανύσματα. Για την ενσωμάτωση αυτή καλείται το μοντέλο “Universal Sentence Encoder” της Google [9] μέσω της βιβλιοθήκης TensorFlow.js [52]. Το συγκεκριμένο μοντέλο μετατρέπει το κείμενο σε διανύσματα μεγέθους 512 διαστάσεων. Το διάνυσμα αυτό αποτυπώνει την σημασιολογία του κειμένου όπως ακριβώς εξηγήθηκε και στο Κεφάλαιο 3.

```

const embeddings = [];

for (let i = 0; i < chunks.length; i++) {
  const tensor = await model.embed([chunks[i]]);
  const embedding = tensor.arraySync()[0];
  embeddings.push({
    chunk: chunks[i],
    embedding: embedding,
    source: fileName,
  });
}

```

Στη συνέχεια, όπως φαίνεται και στο πιο πάνω απόσπασμα κώδικα, η συνάρτηση διατρέχει τον πίνακα με τα τμήματα του κειμένου. Για κάθε τμήμα, καλείται το μοντέλο το οποίο ενσωματώνει το κείμενο σε διανυσματική αναπαράσταση. Η συνάρτηση “embed” του μοντέλου επιστρέφει αντικείμενο τύπου Tensor [52]. Για να εξαχθεί το πραγματικό διάνυσμα στο αντικείμενο Tensor, πρέπει να μετατραπεί το αντικείμενο σε πίνακα και να διαβαστεί το διάνυσμα το οποίο εμφανίζεται στην πρώτη θέση.

4.2.2 Αποθήκευση Διανυσμάτων

Εφόσον πραγματοποιηθεί η μετατροπή του κειμένου σε διάνυσμα ήρθε η ώρα να αποθηκευτεί το τμήμα. Το κάθε τμήμα αποθηκεύεται σε μια μεταβλητή τύπου πίνακα η οποία περιέχει το τμήμα σε κανονική μορφή, το αντίστοιχο διάνυσμα, και το όνομα του αρχείου στο οποίο ανήκει ως μετά-δεδομένο.

Όταν ολοκληρωθεί η δημιουργία των διανυσμάτων για όλο το κείμενο, ακολουθεί η αποθήκευση τους στην τοπική IndexedDB [29][1] εντός του προγράμματος περιήγησης του χρήστη. Πρώτα, η εφαρμογή ελέγχει αν υπάρχει ήδη η βάση και την ανοίγει, διαφορετικά πρώτα την δημιουργεί και έπειτα την ανοίγει.

Η διαδικασία για την ενεργοποίηση της βάσης είναι ασύγχρονη έτσι δημιουργείται ένα αντικείμενο τύπου “Promise” [30]. Το αντικείμενο αυτό δηλώνει αν μια χρονοβόρα διαδικασία έχει ολοκληρωθεί, και έπειτα επιστρέφει αν ήταν επιτυχής ή όχι. Με το αντικείμενο αυτό διασφαλίζεται ότι η υπόλοιπη εφαρμογή περιμένει την ολοκλήρωση της διαδικασίας.

Με το άνοιγμα της βάσης αποθηκεύεται το όνομα της, και ο αριθμός εκδοχής της. Ταυτόχρονα ελέγχει αν υπάρχει ο χώρος στον οποίο θα αποθηκευτούν τα δεδομένα. Όταν ολοκληρωθεί με επιτυχία το άνοιγμα ή η δημιουργία της βάσης, το “Promise” αντικείμενο ενημερώνει το πρόγραμμα.

```
async function storeChunksInIndexedDB(chunks) {  
  const db = await openDB();
```

```

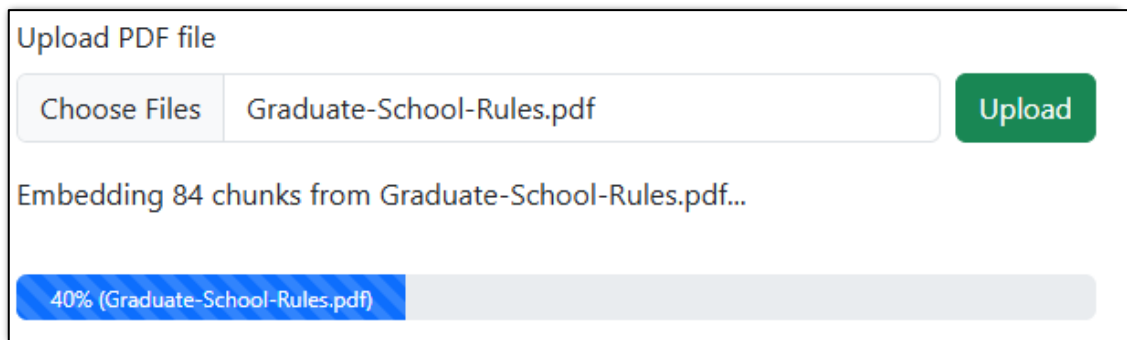
const tx = db.transaction("chunks", "readwrite");
const store = tx.objectStore("chunks");
for (const chunk of chunks) {
  await new Promise((res, rej) => {
    const req = store.add(chunk);
    req.onsuccess = () => res();
    req.onerror = () => rej(req.error);
  });
}
await tx.complete;
db.close();
}

```

Εφόσον η βάση είναι έτοιμη να δεχτεί δεδομένα, το πρόγραμμα αρχικοποιεί μια συναλλαγή για την βάση. Μια συναλλαγή είναι ένα σύνολο από λειτουργίες διαβάσματος και γραψίματος στη βάση. Η βάση έχει χαρακτηριστικό ατομικότητας, οπότε όλες οι ενέργειες ή θα πετύχουν ή θα αποτύχουν μαζί [29].

Κατά τη διάρκεια της αρχικοποίησης της συναλλαγής, πρέπει επίσης να δηλωθεί το αντικείμενο το οποίο θα επηρεάσει την βάση, αλλά και με ποιο τρόπο. Στην συγκεκριμένη περίπτωση δηλώνεται ότι θα εισαχθούν και θα διαβαστούν τα chunks. Στη συνέχεια, αρχικοποιείται ένας δείκτης στην περιοχή όπου είναι αποθηκευμένα τα “chunks” [29]. Ακολούθως, αρχίζει η διαδικασία της αποθήκευσης των τμημάτων του κειμένου. Το πρόγραμμα διαβάζει ένα προς ένα τα αντικείμενα που βρίσκονται στον πίνακα που δέχθηκε ως είσοδο η συνάρτηση. Σε αυτό τον πίνακα, όπως προαναφέρθηκε, περιέχονται το κείμενο, το αντίστοιχο διάνυσμα, και μια μοναδική ταυτότητα. Το κάθε στοιχείο του πίνακα με τα τρία χαρακτηριστικά αποθηκεύεται, με τη χρήση της μεθόδου “add” [29], το ένα μετά το άλλο, αυτό επιτυγχάνεται με την χρήση ενός “Promise” [30] αντικειμένου. Μετά την αποθήκευση όλων των δεδομένων, η συναλλαγή ολοκληρώνεται και η βάση κλείνει.

Κατά τη διάρκεια της διαδικασίας αυτής, ο χρήστης ενημερώνεται για την πρόοδο της ενσωμάτωσης, με την χρήση μιας ράβδου προόδου [3] (βλ. Σχήμα 4.3).



Σχήμα 4.3: Ενημέρωση για Αποθήκευση Τμημάτων

4.2.3 Προβολή Εγγράφων

Όταν ολοκληρωθεί η αποθήκευση των τμημάτων για κάθε έγγραφο, στην διεπαφή του χρήστη πλέον εμφανίζεται μια λίστα με όλα τα έγγραφα τα οποία ανέβασε (βλ. Σχήμα 4.4). Χρησιμοποιώντας την λίστα αυτή ο χρήστης μπορεί να διαχειριστεί τα έγγραφα αλλά και τα τμήματά τους.



Σχήμα 4.4: Λίστα Εγγραφών

Για την δημιουργία της πιο πάνω λίστας χρησιμοποιείται η συνάρτηση “populateDropdownFromIndexedDB”. Αρχικά η συνάρτηση καλεί μια βοηθητική συνάρτηση την “getAllChunksFromIndexedDB” η οποία διαβάζει όλα τα τμήματα από την βάση. Η συγκεκριμένη συνάρτηση χρησιμοποιείται και σε ακόλουθες λειτουργίες. Η

συνάρτηση αυτή ανοίγει την βάση και δημιουργεί μια συναλλαγή τύπου “readwrite” όπως ακριβώς και η συνάρτηση “storeChunksInIndexedDB”. Έπειτα, βρίσκει που είναι αποθηκευμένα τα τμήματα και τα διαβάζει όλα. Με την ολοκλήρωση επιστρέφει τα τμήματα που διαβάστηκαν.

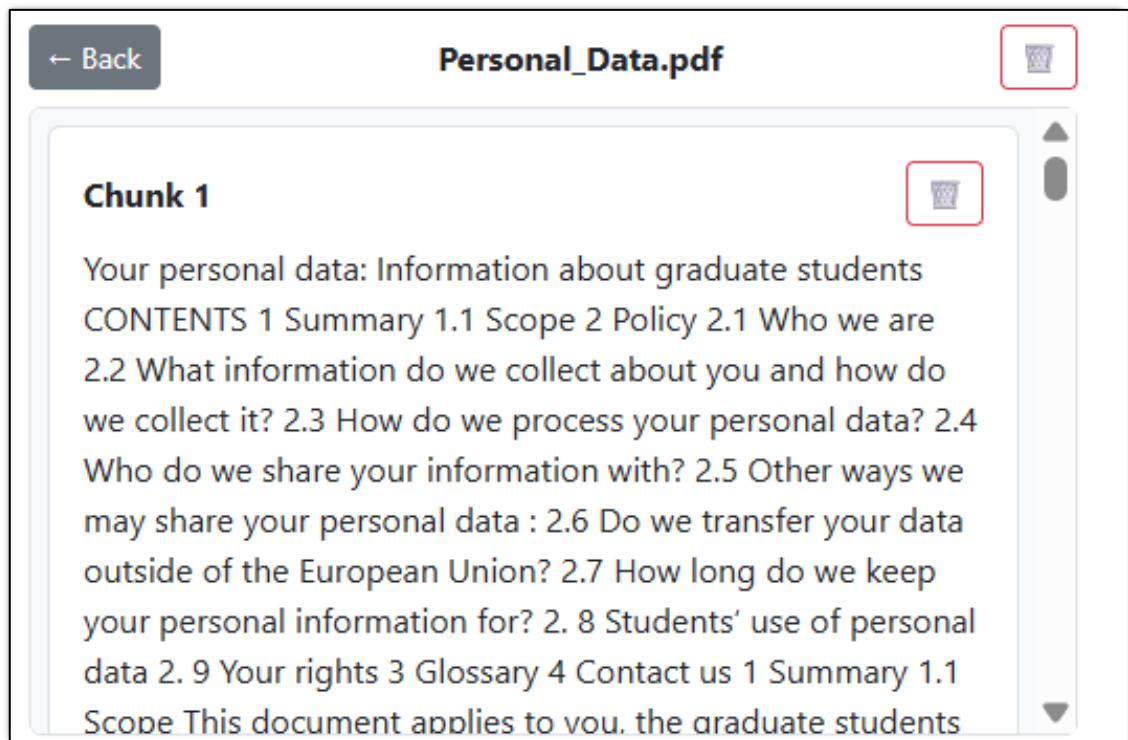
```
async function getAllChunksFromIndexedDB() {
  const db = await openDB();
  const tx = db.transaction("chunks", "readonly");
  const store = tx.objectStore("chunks");
  const chunks = await new Promise((resolve, reject) => {
    const request = store.getAll();
    request.onsuccess = () => resolve(request.result);
    request.onerror = () => reject(request.error);
  });
  db.close();
  return chunks;
}
```

Στη συνέχεια, η συνάρτηση επιλέγει το HTML container το οποίο χρησιμοποιείται για την λίστα των αρχείων. Ακολουθώς, επιλέγει το όνομα του αρχείου από τα μετά-δεδομένα του κάθε τμήματος. Για να εξασφαλίσει ότι το κάθε αρχείο θα εμφανίζεται μόνο μια φορά χρησιμοποιεί την δομή δεδομένων “Set” [31]. Τέλος, δημιουργεί ένα <div> element στο οποίο προσθέτει το όνομα του αρχείου.

Το κάθε όνομα αρχείου πρέπει να είναι επιλέξιμο από τον χρήστη έτσι ώστε να επιτρέπεται η επεξεργασία του. Έτσι, η συνάρτηση δημιουργεί ένα element το οποίο καθιστά το κάθε όνομα αρχείου επιλέξιμο για τον χρήστη. Ταυτόχρονα με την δημιουργία του element εμφανίζει στον χρήστη την δυνατότητα να διαγράψει το αρχείο από την βάση, αυτή η λειτουργία επεξηγείται στην συνέχεια.

Επιπρόσθετα, ο χρήστης μπορεί να επιλέξει ένα έγγραφο για να δει όλα τα τμήματα τα οποία το αντιπροσωπεύουν. Αυτή η λειτουργία πραγματοποιείται με την χρήση της συνάρτησης “showPdfDetails” η οποία παίρνει ως είσοδο το όνομα του αρχείου και το συγκεκριμένο element το οποίο επέλεξε ο χρήστης.

Αρχικά, η συνάρτηση αυτή ετοιμάζει την διεπαφή του χρήστη ώστε να εμφανίσει τις λεπτομέρειες για το συγκεκριμένο έγγραφο (βλ. Σχήμα 4.5). Στη συνέχεια, διαβάζει τα τμήματα κειμένου του συγκεκριμένου αρχείου με τη χρήση μιας βοηθητικής μεθόδου “getChunksBySource” [29] η οποία είναι παρόμοια με την συνάρτηση “getAllChunksFromIndexedDB”, με τη διαφορά ότι παίρνει ως είσοδο το όνομα του αρχείου για το οποίο θέλει να επιστρέψει τα τμήματα.



Σχήμα 4.5: Λεπτομέρειες Εγγράφου

Έπειτα, διαβάζει το κάθε τμήμα ένα προς ένα. Χρησιμοποιώντας τα δεδομένα που είναι αποθηκευμένα για κάθε τμήμα κειμένου ετοιμάζει την διεπαφή χρήστη. Για κάθε τμήμα παρουσιάζει τον αριθμό του, και το κείμενο το οποίο αντιπροσωπεύει. Επίσης, δίνει την δυνατότητα στο χρήστη να διαγράψει συγκεκριμένα τμήματα που επιθυμεί. Οι διαδικασίες διαχείρισης των εγγράφων αναλύονται στο επόμενο υποκεφάλαιο.

4.2.4 Διαχείριση Εγγράφων

Ένας τρόπος με τον οποίο ο χρήστης διαχειρίζεται τις εγγραφές είναι ότι μπορεί να διαγράψει ολόκληρο το έγγραφο, μαζί με τα τμήματά του. Αυτό γίνεται όταν επιλέξει το εικονίδιο – κουμπί, [3][4] του κάδου δίπλα από το όνομα του αρχείου. Μόλις πατήσει το εικονίδιο ο χρήστης καλείται η συνάρτηση “deleteChunksBySource(fileName)”, η οποία δέχεται ως είσοδο το όνομα του αρχείου το οποίο επιθυμεί να διαγράψει ο χρήστης.

Η συνάρτηση αυτή λειτουργεί με τον ίδιο τρόπο όπως και η “storeChunksInIndexedDB” η οποία αποθηκεύει τα τμήματα. Αρχικά, ενεργοποιεί την βάση και αρχικοποιεί μια συναλλαγή τύπου “readwrite” [29] η οποία έχει πρόσβαση στα τμήματα (chunks). Στη συνέχεια, η συνάρτηση διαβάζει όλα τα αποθηκευμένα τμήματα της βάσης με την βοήθεια της συνάρτησης “getAllChunksFromIndexedDB”. Έπειτα, διαβάζει τα τμήματα που ανάκτησε ένα προς ένα και όσα ανήκουν στο συγκεκριμένο έγγραφο τα διαγράφει από την βάση. Ο έλεγχος για το έγγραφο στο οποίο ανήκουν γίνεται με την σύγκριση της μεταπληροφορίας “source” με την είσοδο της συνάρτησης. Στο τέλος η συναλλαγή ολοκληρώνεται και η βάση κλείνει [29].

Με παρόμοιο τρόπο ο χρήστης έχει την δυνατότητα να διαγράψει συγκεκριμένο τμήμα του εγγράφου από την βάση. Η διαδικασία για την διαγραφή συγκεκριμένου τμήματος είναι λιγότερο περίπλοκη αφού δεν απαιτεί την ανάκτηση όλων των τμημάτων από την βάση. Η είσοδος της συνάρτησης που είναι υπεύθυνη για την διαγραφή του τμήματος είναι η ταυτότητα του συγκεκριμένου τμήματος. Επομένως, η διαγραφή γίνεται εύκολα με τη χρήση της ταυτότητας αυτής και τη μέθοδο “delete” της IndexedDB [29].

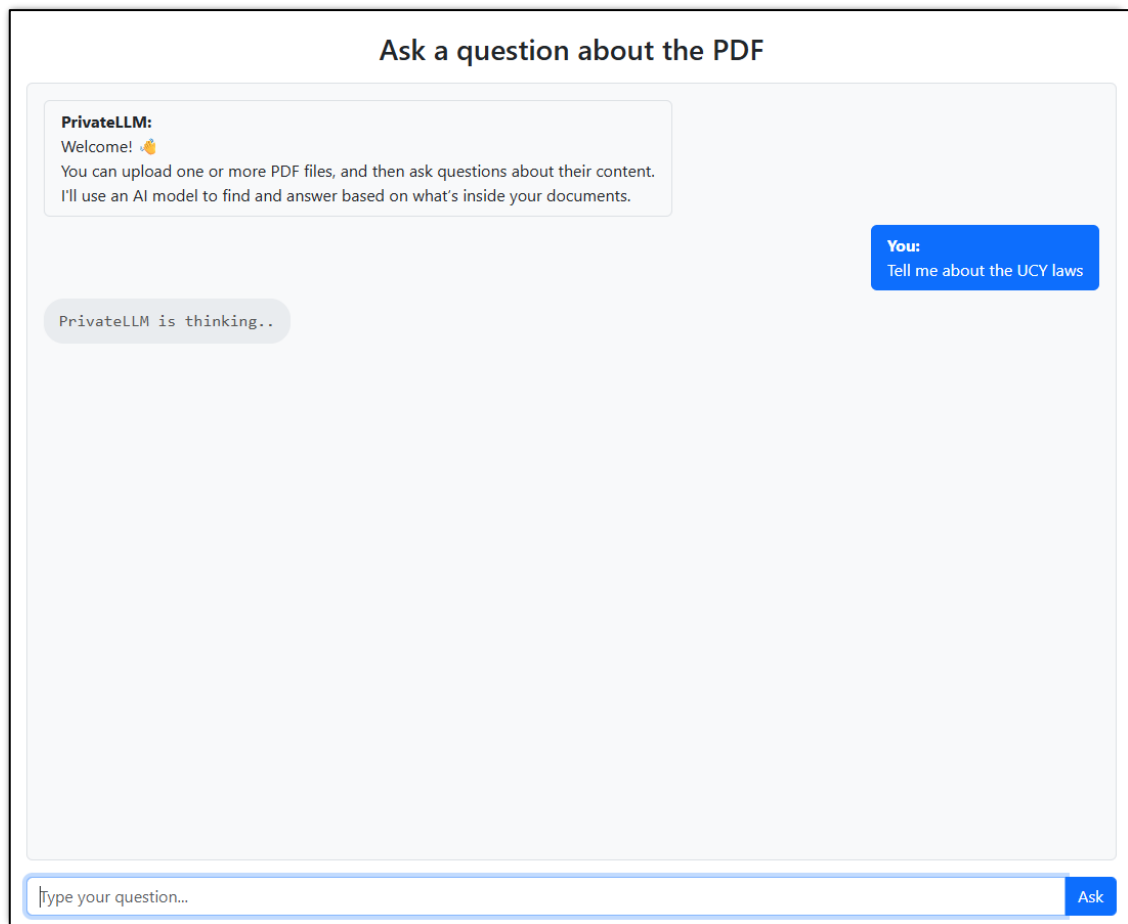
Ακόμη, υπάρχει η δυνατότητα ο χρήστης να διαγράψει τα πάντα από την βάση. Όταν ο χρήστης πατήσει το κουμπί για διαγραφή ολόκληρης της βάσης καλείται η συνάρτηση “clearSessionIndexedDB”. Η συνάρτηση αυτή βρίσκει το όνομα της συγκεκριμένης βάσης που θέλει να διαγράψει ο χρήστης. Με βάση αυτό το όνομα καλεί την μέθοδο “deleteDatabase” [29] της IndexedDB και διαγράφει την συγκεκριμένη βάση. Τέλος, ενημερώνει το χρήστη για την επιτυχία της διαγραφής και φέρνει την λίστα με τα αρχεία (βλ. Σχήμα 4.4) στην αρχική της μορφή.

4.3 Ανάλυση & Απάντηση Ερωτήσεων

Η φόρτωση και η αποθήκευση των εγγράφων και των τμημάτων του κειμένου αποτελεί τη πρώτη βασική λειτουργία. Αφού ολοκληρωθεί η διαδικασία αυτή, ο χρήστης είναι πλέον σε θέση να αλληλοεπιδράσει με ένα Μεγάλο Γλωσσικό Μοντέλο και να ρωτήσει απορίες σχετικά με τα έγγραφα που ανέβασε. Ταυτόχρονα το σύστημα έχει την δυνατότητα να προστατεύσει τις ερωτήσεις του χρήστη, αλλά και τα δεδομένα του με χρήση αλγορίθμων ιδιωτικότητας. Στις επόμενες υποενότητες θα αναλυθεί ο τρόπος με τον οποίο πραγματοποιείται η αλληλεπίδραση του χρήστη με το μοντέλο.

4.3.1 Εύρεση Συμφραζόμενων

Στην διεπαφή του χρήστη υπάρχει ένα πεδίο το οποίο επιτρέπει στον χρήστη να υποβάλει την ερώτησή του (βλ. Σχήμα 4.6). Μόλις στείλει την ερώτηση ο χρήστης, το πεδίο αυτό καλεί την συνάρτηση [28] “handleUserQuestion” η οποία είναι υπεύθυνη για τον συντονισμό της αλληλεπίδρασης.



Σχήμα 4.6: Πλαίσιο Συνομιλίας

Η συνάρτηση αρχικά λαμβάνει την ερώτηση του χρήστη και την αποθηκεύει ως μεταβλητή. Επίσης, ενημερώνει την διεπαφή του χρήστη εμφανίζοντας την ερώτηση του ως νέο μήνυμα (βλ. Σχήμα 4.6). Στη συνέχεια το σύστημα φορτώνει το “Universal Sentence Encoder” της Google [9][52] για να το χρησιμοποιήσει στην ετοιμασία της ερώτησης που θα στείλει στο μοντέλο.

Έπειτα, το σύστημα χρησιμοποιώντας το “Universal Sentence Encoder” μετατρέπει την ερώτηση του χρήστη σε διάνυσμα, όπως ακριβώς μετέτρεψε και τα τμήματα του κειμένου. Αυτό το βήμα είναι απαραίτητο έτσι ώστε να μπορεί να πραγματοποιηθεί διανυσματική σύγκριση της ερώτησης με τα αποθηκευμένα τμήματα.

Ακολούθως, η συνάρτηση καλεί την βοηθητική συνάρτηση “getAllChunksFromIndexedDB” [29] για να πάρει όλα τα τμήματα κειμένου που θα χρειαστεί. Επιπρόσθετα, το σύστημα αρχικοποιεί τις μεταβλητές στις οποίες θα αποθηκευτούν τα συμφραζόμενα που συσχετίζονται με την ερώτηση του χρήστη. Ακόμη, το σύστημα αγνοεί μικρά τμήματα τα οποία κρίνει ότι δεν θα είναι χρήσιμα για την απάντηση της ερώτησης. Τέτοια τμήματα είναι οι κεφαλίδες του κειμένου ή οι πίνακες.

Αφού μετατρέψει την ερώτηση σε διάνυσμα, και αποκτήσει όλα τα τμήματα κειμένου το σύστημα τώρα θα κάνει σύγκριση του διανύσματος της ερώτησης, με τα υπόλοιπα διανύσματα. Η διανυσματική σύγκριση εκτελείται με την βοηθητική συνάρτηση “cosineSimilarity” η οποία παίρνει ως είσοδο το διάνυσμα ερώτησης, και το κάθε διάνυσμα των τμημάτων ένα προς ένα.

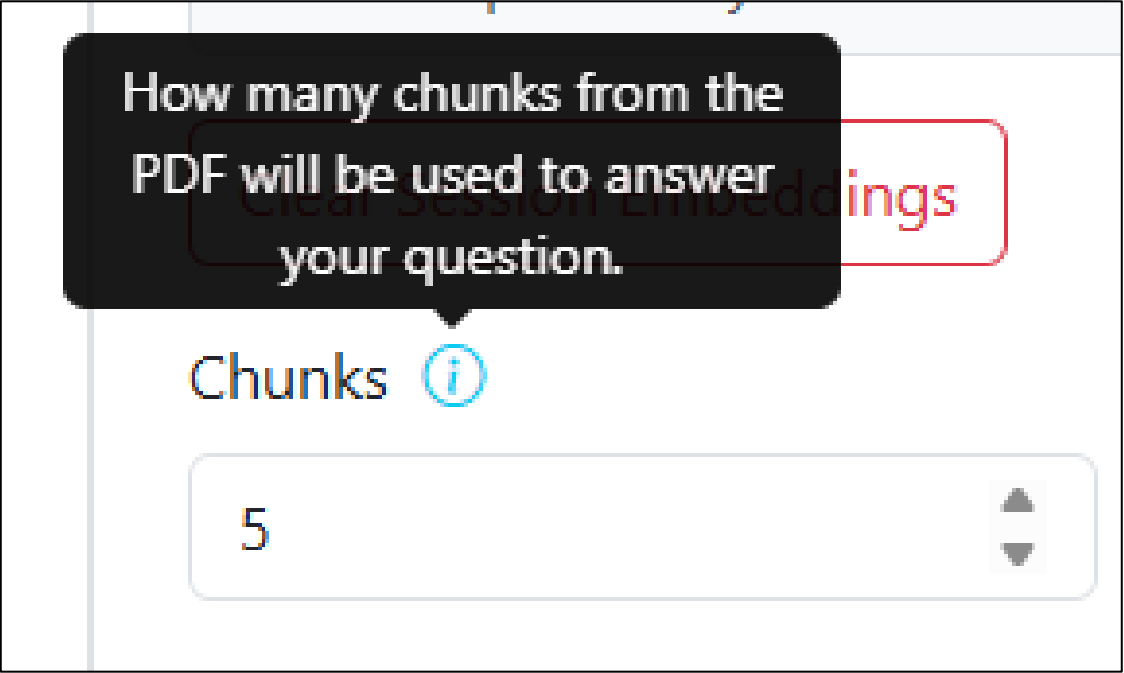
```
function cosineSimilarity(a, b) {  
  const dot = a.reduce((sum, ai, i) => sum + ai * b[i], 0);  
  const magA = Math.sqrt(a.reduce((sum, ai) => sum + ai * ai, 0));  
  const magB = Math.sqrt(b.reduce((sum, bi) => sum + bi * bi, 0));  
  return dot / (magA * magB);  
}
```

Η συνάρτηση αυτή ακολουθεί ακριβώς την εξίσωση της συνημιτονοειδούς ομοιότητας. Αρχικά βρίσκει το εσωτερικό γινόμενο των δύο διανυσμάτων που δέχεται ως είσοδο. Στη συνέχεια, βρίσκει το μέτρο των δύο διανυσμάτων ξεχωριστά. Στο τέλος διαιρεί το εσωτερικό γινόμενο με το γινόμενο των δύο μέτρων. Η τιμή που υπολογίζει αντιπροσωπεύει την σημασιολογική ομοιότητα των δύο διανυσμάτων όπως εξηγήθηκε και στο κεφάλαιο 2 [65]. Αξίζει να σημειωθεί ότι η τιμή αυτή κυμαίνεται από 1 μέχρι -1, αλλά η πιθανότητα η ομοιότητα των δύο διανυσμάτων να είναι αρνητική, δηλαδή τα δύο διανύσματα να έχουν αντίθετη σημασία, είναι αρκετά μικρή σε καλά εκπαιδευμένα μοντέλα όπως το “Universal Sentence Encoder” [9][52].

Η τιμή ομοιότητας μεταξύ των δύο διανυσμάτων επιστρέφεται στην συνάρτηση “handleUserQuestion” η οποία την αποθηκεύει σε ένα αντικείμενο μαζί με το κείμενο του τμήματος, και το έγγραφο στο οποίο ανήκει. Όταν υπολογιστεί η ομοιότητα για κάθε τμήμα με την ερώτηση, ταξινομούνται τα διανύσματα σε φθίνουσα σειρά με βάση το

cosine similarity τους. Αυτό έχει ως αποτέλεσμα, τμήματα που είναι σχετικά με την ερώτηση, και άρα έχουν cosine similarity κοντά στην μονάδα, να εμφανίζονται πρώτα [65].

Ο χρήστης πριν στείλει την ερώτηση στο σύστημα, έχει την επιλογή να επιλέξει πόσα συμφραζόμενα θα δεχθεί το γλωσσικό μοντέλο ως είσοδο (βλ. Σχήμα 4.7). Το σύστημα ως προεπιλεγμένη τιμή έχει το 5.



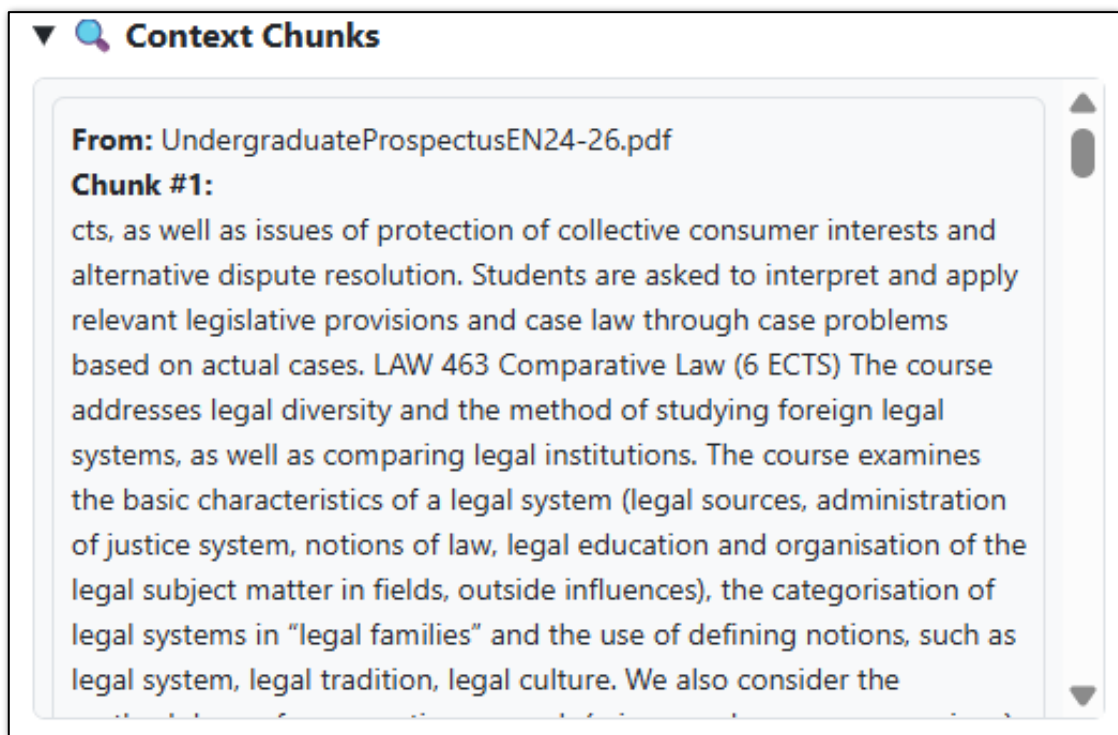
How many chunks from the PDF will be used to answer your question.

Chunks ⓘ

5

Σχήμα 4.7: Επιλογή Αριθμού Τμημάτων

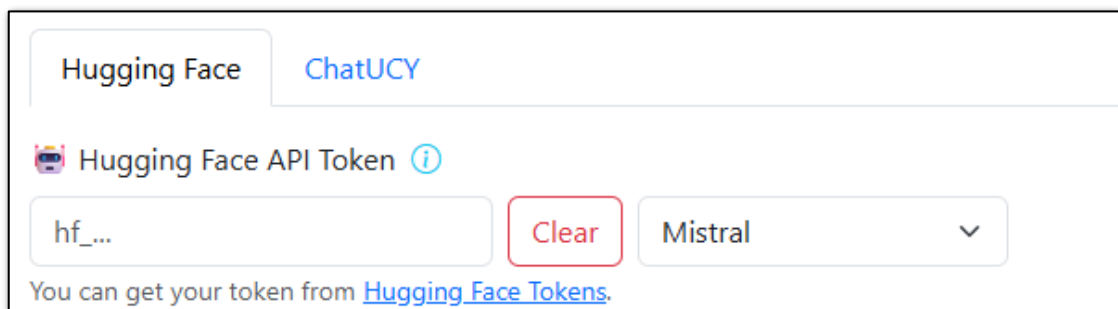
Με βάση αυτό τον αριθμό το σύστημα επιλέγει τα k πιο σχετικά τμήματα για να χρησιμοποιήσει ως συμφραζόμενα. Τέλος, ενώνει όλα τα συμφραζόμενα μαζί. Όσον αφορά την διεπαφή του χρήστη, το σύστημα εμφανίζει τα τμήματα τα οποία θα στείλει στο γλωσσικό μοντέλο (βλ. Σχήμα 4.8).



Σχήμα 4.8: Εμφάνιση Συμφραζόμενων

4.3.2 Επικοινωνία με Γλωσσικά Μοντέλα

Το σύστημα προσφέρει την δυνατότητα στον χρήστη να επιλέξει με ποιο γλωσσικό μοντέλο θα αλληλοεπιδράσει, επιλέγοντας το με την χρήση μιας γραμμής πλοήγησης (βλ. Σχήμα 4.9 και Σχήμα 4.10). Ο χρήστης έχει την επιλογή να επικοινωνήσει με μοντέλα μέσω του HuggingFace Inference [17], αλλά και με μοντέλα τα φιλοξενούνται στον ιστότοπο του "ChatUCY" [53].

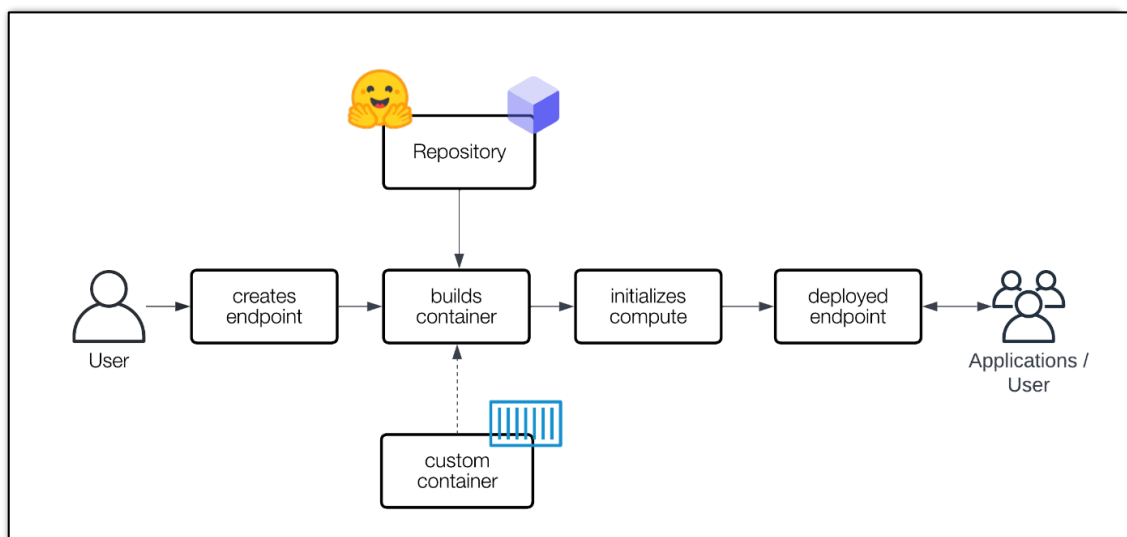


Σχήμα 4.9: Επιλογή HuggingFace Μοντέλου



Σχήμα 4.10: Επιλογή ChatUCY Μοντέλου

Η επικοινωνία με το Hugging Face επιτυγχάνεται με την χρήση του Hugging Face Inference [17] το οποίο ενεργοποιείται με την χρήση API calls. Το Hugging Face Inference προσφέρει μεγάλα γλωσσικά μοντέλα τα οποία είναι ήδη εκπαιδευμένα και φιλοξενούνται στο Hugging Face Hub (βλ. Σχήμα 4.11).

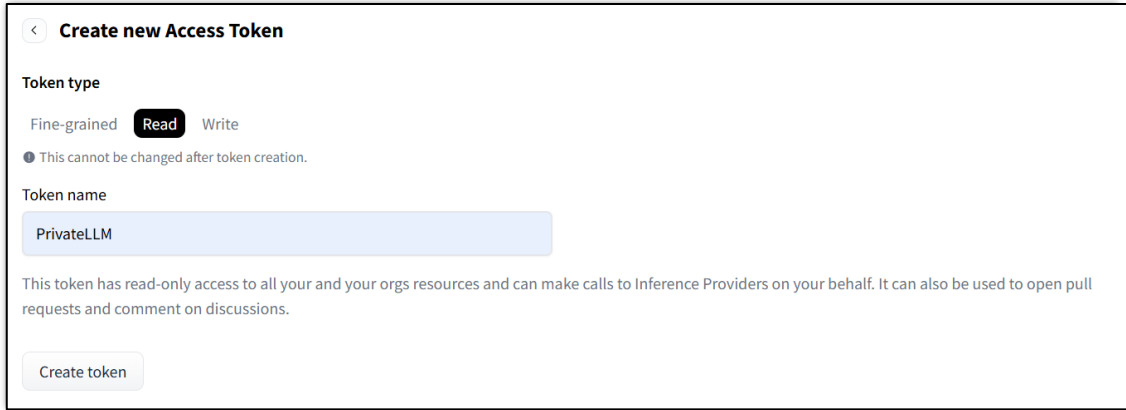


Σχήμα 4.11: Διάγραμμα Λειτουργίας Hugging Face Inference [17]

Η χρήση του Hugging Face Inference είναι δωρεάν αλλά έχει περιορισμούς στη χρήση του κάθε μοντέλου. Για την κλήση μοντέλων μέσω του API χρειάζεται η απόκτηση ενός API token. Η διαδικασία για την απόκτηση ενός token είναι εύκολη και πραγματοποιείται μέσω της επίσημης ιστοσελίδας του Hugging Face [16].

Για τον συγκεκριμένο σκοπό, αρκεί ο τύπος του κλειδιού να είναι μόνο για ανάγνωση (βλ. Σχήμα 4.12). Η επικοινωνία δεν απαιτεί την εγγραφή σε γλωσσικό μοντέλο, παρά μόνο την ανάγνωση και το κατέβασμα μοντέλων. Σε περίπτωση που ο χρήστης επιθυμεί

να ανεβάσει δικό του μοντέλο στην ιστοσελίδα, ή να επεξεργαστεί περιεχόμενο, απαιτείται η δημιουργία ενός κλειδιού εγγραφής.



The screenshot shows a web interface for creating a new access token. At the top, there's a back arrow and the title 'Create new Access Token'. Below this, the 'Token type' section has two options: 'Fine-grained' and 'Read' (which is selected and highlighted in black). A note below says 'This cannot be changed after token creation.' The 'Token name' section has a text input field containing 'PrivateLLM'. At the bottom, there's a 'Create token' button. A descriptive text below the name field states: 'This token has read-only access to all your and your orgs resources and can make calls to Inference Providers on your behalf. It can also be used to open pull requests and comment on discussions.'

Σχήμα 4.12: Δημιουργία Κλειδιού Ανάγνωσης Μέσω Hugging Face

Ο κάθε χρήστης ο οποίος επιθυμεί να χρησιμοποιήσει κάποιο μοντέλο από το Hugging Face είναι απαραίτητο να δημιουργήσει το προσωπικό του κλειδί και να το εισάγει στο ειδικό πλαίσιο της διεπαφής (βλ. Σχήμα 4.9) ώστε να μπορέσει το σύστημα να το δηλώσει στην επικοινωνία.

Μόλις ο χρήστης εισάγει το κλειδί το στο ειδικό πλαίσιο θα λάβει μήνυμα ότι πέτυχε η εισαγωγή. Σε αντίθετη περίπτωση, ο χρήστης ενημερώνεται ότι η διαδικασία υποβολής του κλειδιού του απέτυχε. Το κλειδί του χρήστη αποθηκεύεται στην τοπική βάση του ιστότοπου του. Αυτό εμπλουτίζει την προστασία της ιδιωτικότητας του κάθε χρήστη [33].

```
function storeHuggingFaceToken(token) {  
  if (token && token.startsWith("hf_")) {  
    userHuggingFaceToken = token;  
    sessionStorage.setItem("hfToken", token);  
    showCustomAlert("☑ Hugging Face token stored.");  
  } else {  
    showCustomAlert("✗ Invalid Hugging Face token format.");  
  }  
}
```

Ουσιαστικά ο έλεγχος ο οποίος πραγματοποιείται για την εγκυρότητα του κλειδιού είναι αν αρχίζει από "hf_", όπως όλα τα κλειδιά που παράγει το Hugging Face. Στη συνέχεια

της διαδικασίας υπάρχουν επιπρόσθετοι έλεγχοι όσον αφορά την επιτυχία της επικοινωνίας.

Επιπρόσθετα, ο χρήστης έχει την δυνατότητα να διαγράψει το κλειδί του από την τοπική του βάση. Η διαγραφή πραγματοποιείται με την επιλογή του κουμπιού “Clear” το οποίο καλεί την συνάρτηση “clearHuggingFaceToken”. Η συνάρτηση αυτή είναι υπεύθυνη να διαγράψει τελείως το κλειδί από την βάση, και να ενημερώσει τον χρήστη για την διαγραφή αυτή.

Όταν ο χρήστης στείλει μια ερώτηση σε ένα μοντέλο μέσω του Hugging Face Inference καλείται η συνάρτηση “queryHuggingFaceLLM” η οποία αναλαμβάνει την επικοινωνία μεταξύ του Hugging Face και του συστήματος. Η συνάρτηση αρχικά ελέγχει αν ο χρήστης έχει εισάγει το κλειδί του, σε περίπτωση που το κλειδί δεν είναι αποθηκευμένο ενημερώνει τον χρήστη ότι για να προχωρήσει πρέπει να παραχωρήσει κλειδί.

Ακολούθως η συνάρτηση δημιουργεί την ερώτηση την οποία θα στείλει στο μοντέλο. Η διαδικασία της δημιουργίας της ερώτησης στην σωστή της μορφή θα εξηγηθεί στην επόμενη υποενότητα. Αφού, δημιουργηθεί το σωστό prompt ακολουθεί η επικοινωνία με το μοντέλο. Σε αυτό το στάδιο, η συνάρτηση καλεί το μοντέλο που έχει επιλέξει ο χρήστης και στέλνει POST request.

```
const selectedModel =
  document.getElementById("hfModelSelect")?.value ||
  "mistralai/Mistral-7B-Instruct-v0.3";

const response = await fetch(
  `https://api-inference.huggingface.co/models/${selectedModel}`,
  {
    method: "POST",
    headers: {
      Authorization: `Bearer ${tokenToUse}`,
      "Content-Type": "application/json",
    },
    body: JSON.stringify({
      inputs: prompt,
      parameters: {
```

```
        max_new_tokens: 750,  
        temperature: 0.7,  
    },  
}),  
}  
);
```

Στο σύστημα το προεπιλεγμένο μοντέλο είναι το Mistral με 7 δισεκατομμύρια παραμέτρους [27]. Έπειτα, η συνάρτηση αρχίζει την διαδικασία της δημιουργίας ενός αιτήματος POST. Το POST request είναι μια μέθοδος του πρωτοκόλλου HTTP η οποία χρησιμοποιείται για την αποστολή δεδομένων από τον πελάτη προς τον διακομιστή [46].

Στη συνέχεια, αρχικοποιεί τις κεφαλίδες του αιτήματος οι οποίες περιέχουν το κλειδί του χρήστη και τον τύπο των δεδομένων που θα στείλει. Στην προκειμένη περίπτωση ο τύπος του αντικειμένου που θα στείλει είναι JSON [10]. Επόμενο βήμα είναι η δημιουργία του JSON αντικειμένου που θα σταλεί. Ως είσοδο δηλώνεται το prompt το οποίο κατασκευάστηκε πριν, ενώ δηλώνονται και συγκεκριμένες παράμετροι.

Οι παράμετροι οι οποίες περιέχονται στο JSON αντικείμενο είναι το “max_new_tokens” και το “temperature”. Η πρώτη παράμετρος δηλώνει τον μέγιστο αριθμό χαρακτήρων τον οποίο θα περιέχει η απάντηση του μοντέλου. Η δεύτερη παράμετρος δηλώνει πόση ελευθερία για εξερεύνηση έχει το μοντέλο.

Αφού επιστρέψει απάντηση το μοντέλο, η συνάρτηση ελέγχει την κατάσταση της απάντησης. Συγκεκριμένα ελέγχει αν η κατάσταση της απάντησης περιέχει το σφάλμα 402 το οποίο εμφανίζεται όταν το κλειδί του χρήστη ξεπεράσει το επιτρεπόμενο όριο [17] και τον ενημερώνει για την κατάσταση αυτή. Διαφορετικά, αν υπάρχει οποιοδήποτε άλλο σφάλμα, η συνάρτηση ενημερώνει τον χρήστη ότι υπήρχε πρόβλημα στην επικοινωνία με το Hugging Face. Τέλος, η συνάρτηση είναι υπεύθυνη να επιστρέψει την απάντηση, αν αυτή έρθει χωρίς σφάλματα.

Εναλλακτικά ο χρήστης έχει την δυνατότητα να επικοινωνήσει με μοντέλα τα οποία φιλοξενούνται σε εικονικό μηχάνημα στο εργαστήριο Διαχείρισης Συστημάτων Βάσεων

Δεδομένων στο Πανεπιστήμιο Κύπρου [54]. Τα γλωσσικά μοντέλα αυτά χρησιμοποιούνται στην ιστοσελίδα ChatUCY [53], στην οποία ο χρήστης έχει την δυνατότητα να επικοινωνήσει με αυτά.

Στο σύστημα PrivateLLM ο χρήστης έχει την δυνατότητα να επιλέξει ένα από τα μοντέλα αυτά, με την χρήση του αναπτυσσόμενου μενού (βλ. Σχήμα 4.10), και να στείλει σε εκείνα την ερώτηση που επιθυμεί.

Η επικοινωνία με το ChatUCY πραγματοποιείται με την χρήση της συνάρτησης “sendToChatUCY” η οποία καλείται από την συνάρτηση “handleUserQuestion” υπό την προϋπόθεση ότι ο χρήστης έχει επιλέξει να επικοινωνήσει με το ChatUCY αντί με το Hugging Face. Η συνάρτηση αυτή παίρνει ως είσοδο την ερώτηση του χρήστη, και τα συμφοραζόμενα.

```
const body = {  
  model: selectedModel + ":latest",  
  messages: messages,  
  websearch: false,  
  clientSideRag: false,  
};
```

Η συνάρτηση δημιουργεί το αντικείμενο “body” το οποίο ορίζει το αίτημα που αποστέλλεται στο API του ChatUCY [53]. Το αντικείμενο αυτό περιέχει τέσσερα στοιχεία. Το πρώτο δηλώνει το όνομα του μοντέλου ακριβώς με τον τρόπο που είναι αποθηκευμένα στο εικονικό μηχάνημα του εργαστηρίου. Η δεύτερη παράμετρος περιέχει την ερώτηση με την σωστή μορφή για τα μοντέλα του ChatUCY. Ο τρόπος με τον οποίο δημιουργείται η συγκεκριμένη ερώτηση θα αναλυθεί στην επόμενη υποενότητα. Οι τελευταίες δύο παράμετροι αφορούν λειτουργίες του ChatUCY τις οποίες δεν αξιοποιεί το σύστημα.

```
const response = await  
fetch("https://chatucy.cs.ucy.ac.cy/send_message", {  
  method: "POST",  
  headers: {
```

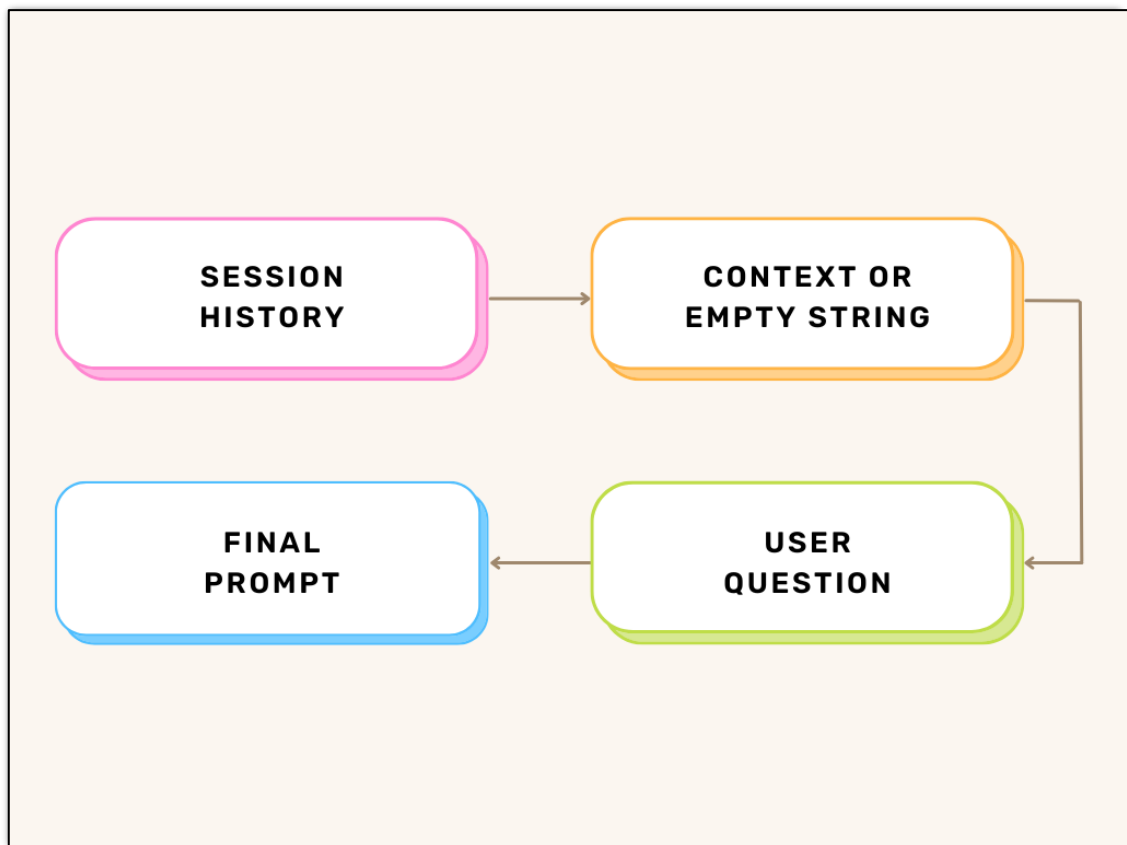
```
    "Content-Type": "application/json",  
  },  
  body: JSON.stringify(body),  
  credentials: "omit",  
});
```

Η συνάρτηση δημιουργεί ένα Fetch API το οποίο είναι υπεύθυνο για την ανάκτηση δεδομένων από ένα διακομιστή. Εκεί δηλώνεται το endpoint, το οποίο είναι η διαδρομή URL του διακομιστή στον οποίο αποστέλλονται τα αιτήματα. Το συγκεκριμένο endpoint δέχεται μηνύματα και επιστρέφει απάντηση.

Η επικοινωνία με τα μοντέλα στο ChatUCY πραγματοποιείται με αίτημα POST, όπως και η επικοινωνία με το Hugging Face. Στην κεφαλίδα δηλώνεται ότι τα δεδομένα που θα σταλούν στο μοντέλο έχουν JSON μορφή και έπειτα μετατρέπει το “body” σε JSON. Τέλος, η επικοινωνία δεν απαιτεί στοιχεία ή οποιασδήποτε μορφής ταυτοποίηση. Τέλος, η συνάρτηση διαβάζει και επεξεργάζεται την απάντηση που επιστρέφει ο διακομιστής.

4.3.3 Επεξεργασία Ερωτήματος

Η λογική της δημιουργίας της ερώτησης είναι ίδια και για τους δύο τρόπους επικοινωνίας. Η ερώτηση αρχικά περιλαμβάνει την ιστορία της συνομιλίας έτσι ώστε το μοντέλο να γνωρίζει όσα ειπώθηκαν πριν. Στη συνέχεια, προστίθενται τα συμφοραζόμενα τα οποία ανακτήθηκαν με την μέθοδο RAG. Τέλος, το prompt συμπληρώνεται με την ερώτηση του χρήστη. Έτσι, δημιουργείται μια ευανάγνωστη ερώτηση με καθαρή δομή για καλύτερη καθοδήγηση του γλωσσικού μοντέλου (βλ. Σχήμα 4.13).



Σχήμα 4.13: Ροή Κατασκευής Τελικής Ερώτησης

Η συνομιλία αποθηκεύεται συνεχώς στην τοπική βάση του ιστότοπου του κάθε χρήστη [33]. Αυτό πραγματοποιείται αυτόματα κάθε φορά που ο χρήστης στέλνει ένα μήνυμα στο σύστημα. Η συνάρτηση που είναι υπεύθυνη για την αποθήκευση των μηνυμάτων είναι η “saveChatMessage”.

```
function saveChatMessage(sender, message) {  
  const key = `chat_history_${currentSession}`;  
  const history = JSON.parse(localStorage.getItem(key) || "[]");  
  
  history.push({ sender, message });  
  localStorage.setItem(key, JSON.stringify(history));  
}
```

Η είσοδος της συνάρτησης είναι η μεταβλητή “sender” η οποία υποδηλώνει αν το μήνυμα στάλθηκε από τον χρήστη ή από το μοντέλο, και η μεταβλητή “message” η οποία περιέχει το μήνυμα. Αρχικά, η συνάρτηση δημιουργεί ένα μοναδικό κλειδί για την κάθε συνομιλία. Έπειτα, ελέγχει αν το κλειδί υπάρχει ήδη, αν υπάρχει φορτώνει τα μηνύματα

από την βάση την οποία αντιπροσωπεύει [33], διαφορετικά σημαίνει ότι δεν υπάρχουν μηνύματα στην συνομιλία. Ακολούθως, μετατρέπει την συνομιλία σε πίνακα και προσθέτει το μήνυμα σε αυτόν. Το μήνυμα έχει την ακόλουθη μορφή:

```
{ sender: "user", message: "What is AI?" }
```

Τέλος, επαναφέρει την ιστορία της συνομιλίας σε JSON αντικείμενο για να μπορεί να αποθηκευτεί στην τοπική βάση του χρήστη [33].

```
const key = `chat_history_${currentSession}`;  
fullHistory = JSON.parse(localStorage.getItem(key) || "[]");  
recentTurns = fullHistory  
    .slice(-6)  
    .map((m) => `${m.sender === "user" ? "You" : "PrivateLLM"}:  
    ${m.message}`)  
    .join("\n")
```

Ο τρόπος που προστίθεται η ιστορία της συνομιλίας στις ερωτήσεις που δέχονται τα μοντέλα του Hugging Face και τα μοντέλα του ChatUCY διαφέρουν. Έτσι, είναι απαραίτητη η επεξεργασία της ιστορίας πριν την πρόσθεσή της στην ερώτηση. Η χρήση της προηγούμενης συνομιλίας στην ερώτηση είναι σημαντική ώστε το μοντέλο να έχει περισσότερες πληροφορίες για ότι ειπώθηκε προηγουμένως, χωρίς να χρειάζεται να επαναληφθεί στην ερώτηση του χρήστη.

Για την δημιουργία της ιστορίας η οποία θα σταλεί στο Hugging Face Inference η συνάρτηση “handleUserQuestion” φορτώνει όλη την συνομιλία η οποία είναι αποθηκευμένη στην τοπική βάση [33] και την μετατρέπει σε JSON αντικείμενο. Στη συνέχεια, βρίσκει τα έξι πιο πρόσφατα μηνύματα, δηλαδή τρία από τα μηνύματα του χρήστη και τρία από τα μηνύματα του γλωσσικού μοντέλου.

Ακολούθως, μετατρέπει τα μηνύματα στην σωστή μορφή ώστε να είναι αξιοποιήσιμα από το μοντέλο. Μετατρέπει τα μηνύματα του “user” ως “You” και τα μηνύματα του “bot” σε “PrivateLLM”. Με αυτό το τρόπο, το μοντέλο μπορεί να μάθει πώς

ανταποκρίνεται το σύστημα και έτσι να απαντήσει με παρόμοιο τρόπο. Τέλος, η συνάρτηση συρράπτει όλα τα μηνύματα της συνομιλίας σε μια συμβολοσειρά.

Η διαδικασία εξαγωγής της ιστορίας της συνομιλίας διαφέρει όταν αποστέλλεται στο ChatUCY σε σχέση με το Hugging Face API. Ο λόγος είναι ο τρόπος με τον οποίο διαβάζει τα POST αιτήματα το σύστημα διαφέρει ως προς την ανάθεση του ρόλου του μοντέλου και του ρόλου του χρήστη [53]. Για την μετατροπή στην σωστή μορφή πραγματοποιείται με την βοηθητική συνάρτηση “convertHistoryForChatUCY”. Η συνάρτηση αυτή μετατρέπει το “bot” σε “assistant”, ενώ το “user” το αφήνει όπως είναι.

Στη συνέχεια, ανεξάρτητα με το τρόπο επικοινωνίας με το γλωσσικό μοντέλο, η ιστορία συνομιλίας μαζί με τα συμφραζόμενα τα οποία αποκτήθηκαν πριν ενώνονται μαζί. Έπειτα πραγματοποιείται μια εκτίμηση για τον αριθμό των tokens τα οποία θα σταλούν στο γλωσσικό μοντέλο. Η εκτίμηση αυτή είναι απαραίτητη ώστε να μην ξεπεραστεί το όριο.

```
const encoder = window.GPT3Encoder;
function estimateTokenCount(text) {
  if (!encoder || !text) return 0;
  return encoder.encode(text).length;
}
```

Για τον σκοπό της εκτίμησης χρησιμοποιείται η βιβλιοθήκη “GPT3Encoder” η οποία προσομοιάζει τον τρόπο με τον οποίο χωρίζει ένα γλωσσικό μοντέλο το κείμενο σε tokens [51]. Η συνάρτηση η οποία κάνει την εκτίμηση παίρνει ως είσοδο το κείμενο, σε αυτή την περίπτωση τη συνομιλία, μαζί με τα συμφραζόμενα, ως συμβολοσειρά.

Έπειτα, με την χρήση του encoder η συμβολοσειρά μετατρέπεται σε ένα πίνακα ο οποίος περιέχει token ids. Τα tokens μπορεί να είναι είτε λέξεις, είτε κομμάτια λέξεων, ή ακόμη και σημεία στίξης. Στο τέλος η συνάρτηση επιστρέφει το πλήθος των token ids ο οποίος αντιπροσωπεύει μια εκτίμηση για τον αριθμό των tokens που παράγονται με την συνομιλία και τα συμφραζόμενα.

Το όριο για κάθε prompt είναι κοινό για την ερώτηση και την απάντηση. Αυτό σημαίνει ότι αν η ερώτηση περιέχει 2000 tokens και το όριο είναι 4000 tokens, για την απάντηση του μοντέλου περισσεύουν μόνο 2000 tokens. Ουσιαστικά ο καθορισμός ενός ορίου ελέγχου διασφαλίζει ότι δεν θα υπάρξουν σφάλματα για το μέγεθος της ερώτησης. Έτσι για σκοπούς πρόληψης σφαλμάτων στην συνάρτηση υπάρχει το όριο των 2000 για τα συμφραζόμενα και την ιστορία της συνομιλίας μαζί.

Ακολούθως, πραγματοποιείται μια δυναμική επεξεργασία του μεγέθους των συμφραζόμενων και της συνομιλίας. Η επεξεργασία πραγματοποιείται μόνο αν το συνδυασμένο κείμενο ξεπερνάει το όριο των 2000 tokens. Αν χρειάζεται επεξεργασία τότε διαγράφει τα 2 πιο παλιά μηνύματα από την συνομιλία, κρατώντας τα 4 πιο πρόσφατα. Έπειτα, πραγματοποιεί ξανά μια εκτίμηση του αριθμού των tokens του συνδυασμένου κείμενου.

Ακολούθως, αν το συνδυασμένο κείμενο ξεπερνάει ακόμη το μέγιστο αριθμό των tokens, κάνει επανάληψη και αφαιρεί ένα προς ένα τα συμφραζόμενα αρχίζοντας από αυτό με τη λιγότερη σημασιολογική ομοιότητα. Σε κάθε επανάληψη δημιουργείται νέο συνδυασμένο κείμενο και γίνεται νέα πρόβλεψη [51]. Η επανάληψη πραγματοποιείται μέχρι η πρόβλεψη για το μέγεθος του συνδυασμένου κείμενου δεν ξεπερνάει το όριο. Η διαδικασία αυτή ισχύει μόνο όταν καλούνται μοντέλα από το Hugging Face τα οποία έχουν περιορισμό με τα κλειδιά. Όσον αφορά τα μοντέλα στο ChatUCY, παίρνουν όλα τα συμφραζόμενα και ολόκληρη την ιστορία της συνομιλίας.

Όταν δημιουργηθεί το απαραίτητο συμφραζόμενο, η συνάρτηση ελέγχει αν πρέπει να καλέσει το Hugging Face Inference [17] ή το ChatUCY [53]. Για την κλήση του Hugging Face χρησιμοποιείται η συνάρτηση “queryHuggingFaceLLM” η οποία προαναφέρθηκε.

Στη συνάρτηση αυτή πραγματοποιείται η τελική συρραφή του prompt. Το prompt αυτό αλλάζει ανάλογα με τον αν υπάρχουν αποθηκευμένα έγγραφα. Στην περίπτωση που υπάρχουν αποθηκευμένα έγγραφα σημαίνει ότι το μοντέλο θα δεχθεί συμφραζόμενα. Διαφορετικά, το μοντέλο δεν θα δεχθεί συμφραζόμενα.

```

    prompt = `Answer the following question based ONLY on the context
below.
You MUST always answer in exactly the SAME language as the input
question.

Context:
${context}

Question: ${question}
Answer:`;

```

Πιο πάνω φαίνεται η δομή της ερώτησης που θα λάβει το γλωσσικό μοντέλο στο Hugging Face Inference [17]. Αν η ερώτηση δεν περιέχει συμφραζόμενα τότε το πεδίο “Context” δεν υπάρχει.

Με παρόμοιο τρόπο επιτυγχάνεται και η δημιουργία του prompt το οποίο θα σταλεί στο ChatUCY. Το prompt έχει την ακόλουθη δομή:

```

const chatHistory = convertHistoryForChatUCY(currentSession);
const messages = [];

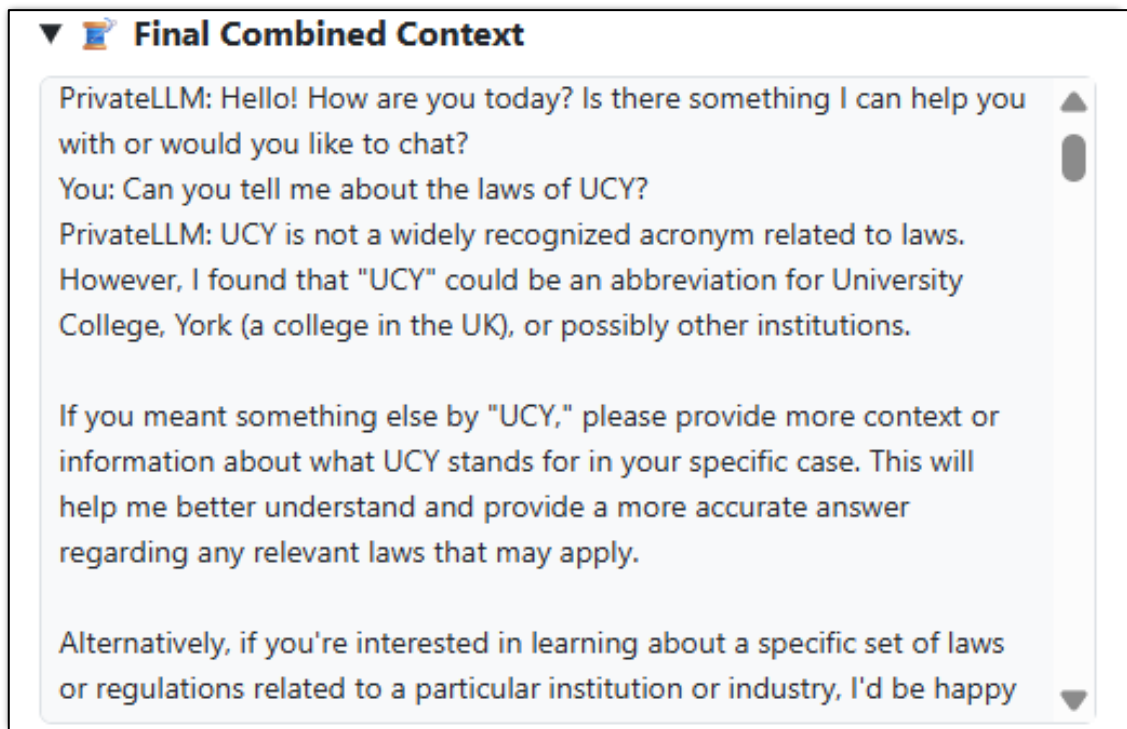
messages.push(...chatHistory); // 1. Previous conversation

if (context && context.trim()) {
    messages.push({
        role: "system",
        content: `Based on the following context, answer the user's next
question.\n\n${context}`,
    }); // 2. System prompt with context
}

messages.push({ role: "user", content: question }); // 3. Actual
question

```

Πρώτα ανακτά την ιστορία της συνομιλίας με τον τρόπο που προαναφέρθηκε. Στη συνέχεια προσθέτει στο μοντέλο την εντολή να απαντήσει την ερώτηση του user που ακολουθεί με βάση τα πιο κάτω. Έπειτα, προσθέτει τα συμφραζόμενα και τέλος προσθέτει την ερώτηση που έστειλε ο χρήστης. Το τελικό prompt παρουσιάζεται στην διεπαφή του χρήστη (βλ. Σχήμα 4.14).



Σχήμα 4.14: Παρουσίαση Τελικού Prompt

Μόλις τα γλωσσικά μοντέλα επιστρέψουν την απάντηση η κάθε συνάρτηση είναι υπεύθυνη να εξάγει την απάντηση από το JSON αντικείμενο.

```
const data = await response.json();  
  
const generated = data?.[0]?.generated_text || "";  
const answerOnly = generated.split("Answer:").pop().trim();  
  
return answerOnly;
```

Η συνάρτηση που πραγματοποιεί την επικοινωνία με το Hugging Face API λαμβάνει την απάντηση και έπειτα από τον πίνακα με τα δεδομένα παίρνει το πρώτο αντικείμενο το οποίο έχει την ακόλουθη μορφή:

```
generated = "Context:\n...\nQuestion: Who is Alan Turing?\nAnswer: Alan Turing was a British mathematician."
```

Η συνάρτηση “queryHuggingFaceLLM” αναλαμβάνει να κρατήσει μόνο το κείμενο μετά από το “Answer” το οποίο και επιστρέφει. Με παρόμοιο τρόπο λειτουργεί και η

συνάρτηση “sendToChatUCY”. Η συνάρτηση διαβάζει την απάντηση όπως έρχεται αντί να περιμένει ολόκληρο το JSON object. Επίσης, χρησιμοποιεί αποκωδικοποιητή για να μετατρέψει το κείμενο από “UTF-8” σε απλό κείμενο [32].

Ακολούθως, κάθε τμήμα του κειμένου που διαβάζει το προσθέτει στην τελική απάντηση. Τέλος, φεύγει επιπρόσθετο κείμενο στην απάντηση που επιστρέφει το API call από το ChatUCY [53] και επιστρέφει την απάντηση μαζί με ολόκληρο το prompt που στάλθηκε αρχικά.

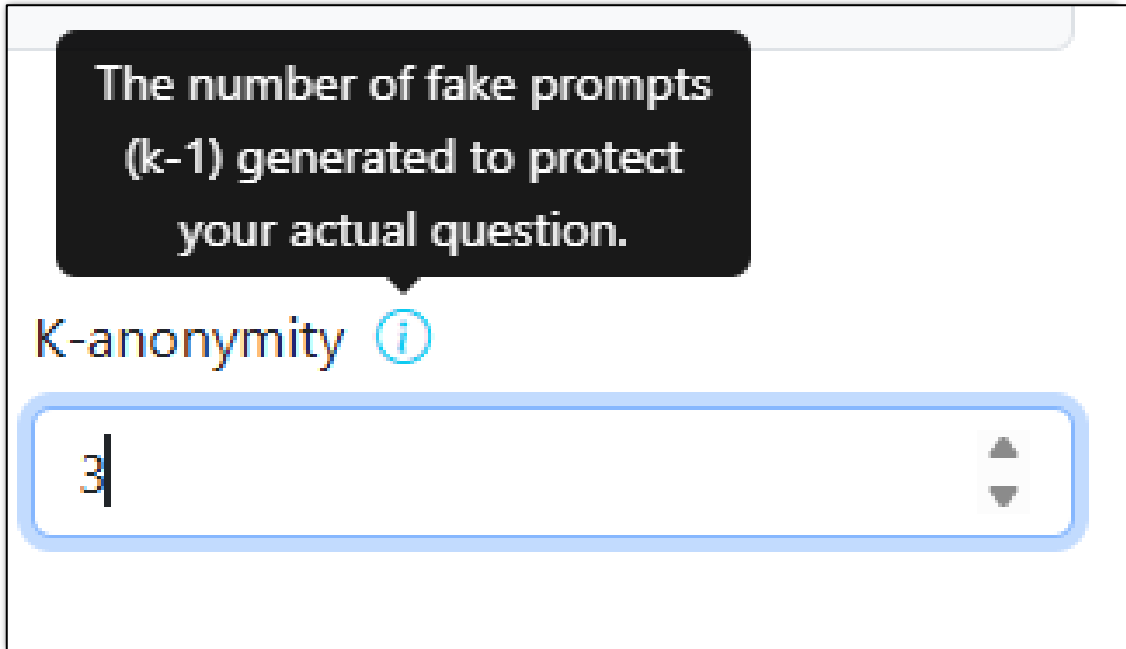
4.3.4 Μέθοδοι Προστασίας Ιδιωτικότητας & Υλοποίηση Κ-Ανωνυμίας

Το σύστημα έχει ως σκοπό την προστασία της ιδιωτικότητας του χρήστη. Ένας τρόπος με τον οποίο επιτυγχάνεται η προστασία της ιδιωτικότητας του χρήστη είναι η χρήση της τοπικής βάσης του ιστότοπου του κάθε χρήστη. Στην τοπική βάση είναι αποθηκευμένο το κλειδί του χρήστη, οι συνομιλίες του, αλλά και τα τμήματα κειμένου από τα έγγραφα που φορτώνει. Επίσης, το σύστημα τρέχει και εκτελεί τις λειτουργίες στην πλευρά του χρήστη.

Αυτό επιτυγχάνεται με την χρήση της κ-ανωνυμίας η οποία παρουσιάστηκε πρώτη φορά από την L. Sweeney [21]. Η κ-ανωνυμία είναι ένας τρόπος προστασίας ευαίσθητων δεδομένων των χρηστών. Για παράδειγμα με τη χρήση γενικών δημογραφικών στοιχείων όπως φύλο, ηλικία, και ταχυδρομικό κώδικα, είναι πιθανό να αποκαλυφθούν πιθανά ευαίσθητα δεδομένα ενός ατόμου με αυτά τα στοιχεία από ένα αντίστοιχο σύνολο δεδομένων. Η κ-ανωνυμία στο συγκεκριμένο παράδειγμα δηλώνει ότι αν υπάρχουν 3 άτομα με ακριβώς τα ίδια δημογραφικά στοιχεία, τότε τα ευαίσθητα δεδομένα τους είναι προστατευμένα με 3-ανωνυμία.

Η λογική αυτή θα εφαρμοστεί στις ερωτήσεις του χρήστη με παρόμοιο τρόπο στο σύστημα. Ο τρόπος εφαρμογής του συγκεκριμένου μοντέλου προστασίας πραγματοποιείται με την δημιουργία κ διαφορετικών ερωτήσεων τα οποία θα στέλνονται στο μοντέλο. Οι κ-1 ερωτήσεις θα περιέχουν ψεύτικη ερώτηση με τυχαία συμφραζόμενα, ενώ η μια θα είναι αληθινή. Με αυτό τον τρόπο το μοντέλο δεν θα γνωρίζει τι πραγματικά

ισχύει. Στον χρήστη επιτρέπεται να διαλέξει τον αριθμό των ψεύτικων ερωτήσεων που θα δημιουργηθούν. Έχει επίσης την δυνατότητα να επιλέξει να μην δημιουργηθούν ψευδείς ερωτήσεις, επιλέγοντας το k αν είναι ίσο με 1 (βλ. Σχήμα 4.15).



The number of fake prompts
($k-1$) generated to protect
your actual question.

K-anonymity ⓘ

3

Σχήμα 4.15: Επιλογή K Ψευδών Ερωτήσεων

Η συνάρτηση “handleUserQuestion” είναι υπεύθυνη να αναγνωρίσει αν πρέπει να παραχθούν ψευδείς ερωτήσεις και να καλέσει την βοηθητική συνάρτηση “generateFakePrompts” η οποία δέχεται ως είσοδο το $k-1$ και με την σειρά της καλεί την συνάρτηση “generateFakeWikipediaPrompt”.

```
async function generateFakePrompts(count) {
  const results = [];
  while (results.length < count) {
    const fake = await fetchFakeWikipediaPrompt();
    if (!results.some((p) => p.question === fake.question)) {
      results.push(fake); // Store both question and context
    }
  }
  return results;
}

async function fetchFakeWikipediaPrompt() {
  try {
```

```

const response = await fetch(
  "https://en.wikipedia.org/api/rest_v1/page/random/summary"
);
const data = await response.json();
const title = data.title;
const context = data.extract || "No summary available.";
const question = `What is ${title}?`;
return { question, context };
} catch (err) {
  console.error("Wikipedia fetch failed:", err);
  return {
    question: "What is quantum physics?",
    context:
      "Quantum physics is a fundamental branch of physics concerned
with processes involving, for example, atoms and photons.",
  };
}
}

```

Η συνάρτηση “generateFakePrompts” είναι υπεύθυνη να δημιουργήσει κ-1 ψεύτικες ερωτήσεις και να τις επιστρέψει. Για την δημιουργία των ψεύτικων ερωτήσεων καλείται η συνάρτηση “generateFakeWikipediaPrompt”. Η συνάρτηση αυτή καλεί το Wikipedia API [63] για να πάρει μια τυχαία σελίδα από την Wikipedia. Από την σελίδα παίρνει τον τίτλο και δημιουργεί μια ερώτηση τύπου “What is..” έτσι ώστε η ερώτηση να αφορά τον τίτλο της σελίδας. Ως συμπραζόμενο η συνάρτηση παίρνει την περίληψη της σελίδας. Άρα δημιουργούνται αντικείμενα με ερώτηση και συμπραζόμενα από την Wikipedia ως ψεύτικες ερωτήσεις. Τέλος η συνάρτηση “generateFakePrompts” επιστρέφει ένα πίνακα με κ-1 τυχαίες ερωτήσεις. Στην περίπτωση που το κ είναι ίσο με ένα, η συνάρτηση επιστρέφει άδειο πίνακα.

Αφού επιστρέψει τον πίνακα, η συνάρτηση “handleUserQuestion” μετατρέπει τις ερωτήσεις αυτές σε αντικείμενα τα οποία έχουν δύο παραμέτρους. Η πρώτη παράμετρος είναι η ερώτηση ως συμβολοσειρά, και η δεύτερη είναι μια δυαδική τιμή που υποδηλώνει αν η ερώτηση είναι η αληθινή ή όχι. Έπειτα, ανακατεύει τα αντικείμενα με τυχαίο τρόπο ώστε να μην μπορεί το μοντέλο να αναγνωρίσει που βρίσκεται η πραγματική ερώτηση.

Πέρα την K-Ανωνυμία, εφαρμόζεται ακόμη μια μέθοδος για την προστασία των δεδομένων του χρήστη. Η μέθοδος αυτή είναι η προσθήκη θορύβου ο οποίος θα οδηγήσει σε επιπρόσθετη σύγχυση του γλωσσικού μοντέλου.

Η προσθήκη του θορύβου πραγματοποιείται με την χρήση της βοηθητικής συνάρτησης “generateNoisyContext”. Η συνάρτηση αυτή δέχεται ως είσοδο τα συμφραζόμενα που ανακτήθηκαν στα προηγούμενα στάδια, και μια μεταβλητή που δηλώνει τον λόγο του θορύβου ως προς τα πραγματικά δεδομένα. Η μεταβλητή αυτή είναι 0.8, δηλαδή το 80% είναι πραγματικά δεδομένα, ενώ 20% είναι θόρυβος.

```
async function generateNoisyContext(realChunks, ratio = 0.8) {
  const fullText = realChunks.join(" ").trim();
  if (!fullText) return "[FALLBACK] No real context available.";

  // Split real context into sentences
  const sentences = fullText
    .split(/(?<=[.?!])\s+/)
    .filter((s) => s.trim().length > 0);

  const totalSentences = sentences.length;
  const numToReplace = Math.floor(totalSentences * (1 - ratio));

  // Fetch noise sentences from Wikipedia
  const noiseSentences = [];
  while (noiseSentences.length < numToReplace) {
    const fake = await fetchFakeWikipediaPrompt();
    const sentence = fake.context
      .split(/(?<=[.?!])\s+/)
      .find((s) => s.length >= 30 && s.length <= 150);

    if (sentence) noiseSentences.push(sentence);
  }

  // Replace random real sentences with noise
  const indices = new Set();
  while (indices.size < numToReplace) {
    indices.add(Math.floor(Math.random() * totalSentences));
  }

  const mixed = sentences.map((s, i) =>
    indices.has(i) ? noiseSentences.pop() : s
  );
}
```

```

return mixed.join(" ");
}

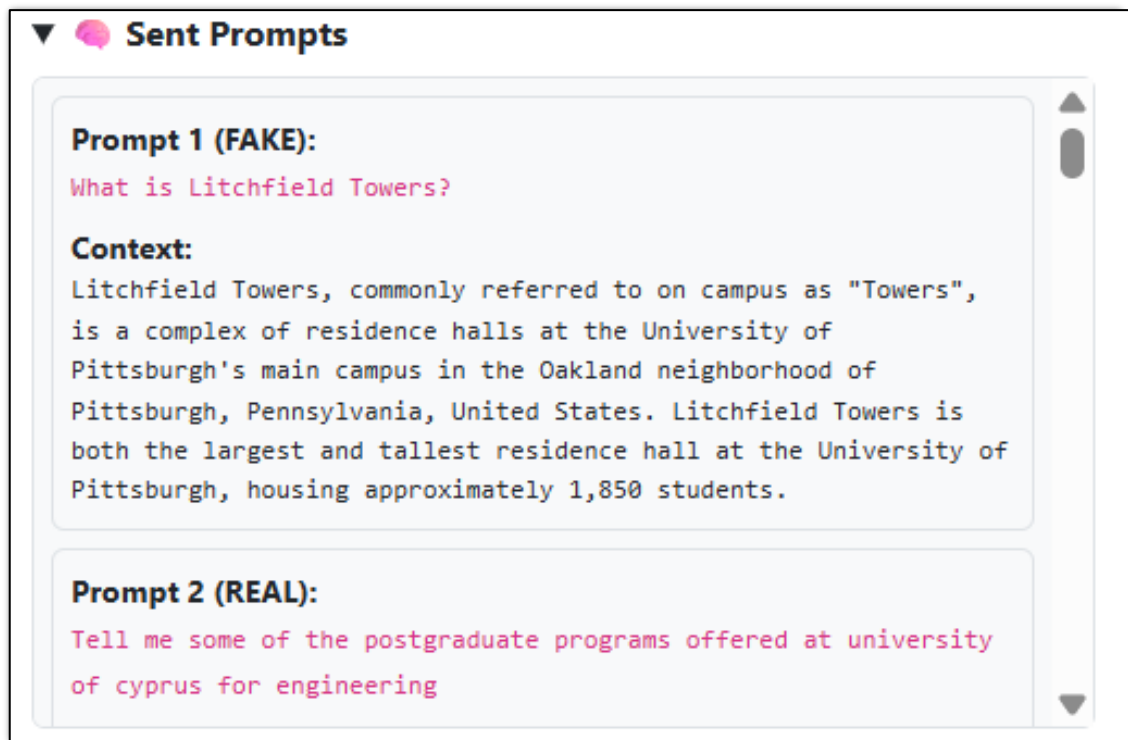
```

Αρχικά, η συνάρτηση μετρά πόσες προτάσεις υπάρχουν στα συμφραζόμενα που δέχτηκε ως είσοδο. Με βάση των αριθμό των προτάσεων και την μεταβλητή του λόγου, υπολογίζει πόσες ψεύτικες προτάσεις πρέπει να προσθέσει. Στη συνέχεια καλεί την συνάρτηση “fetchFakeWikipediaPrompt” για να λάβει ψεύτικα δεδομένα από την Wikipedia [63]. Από τα δεδομένα αυτά παίρνει μερικές προτάσεις και έπειτα επιλέγει τυχαία προτάσεις από τα συμφραζόμενα για να τις αντικαταστήσει. Τέλος επιστρέφει τα συμφραζόμενα με θόρυβο. Ο λόγος των καθαρών με τα θορυβώδη δεδομένα επηρεάζει τόσο την ακρίβεια όσο και την προστασία των δεδομένων (βλ. Πίνακα 4.1).

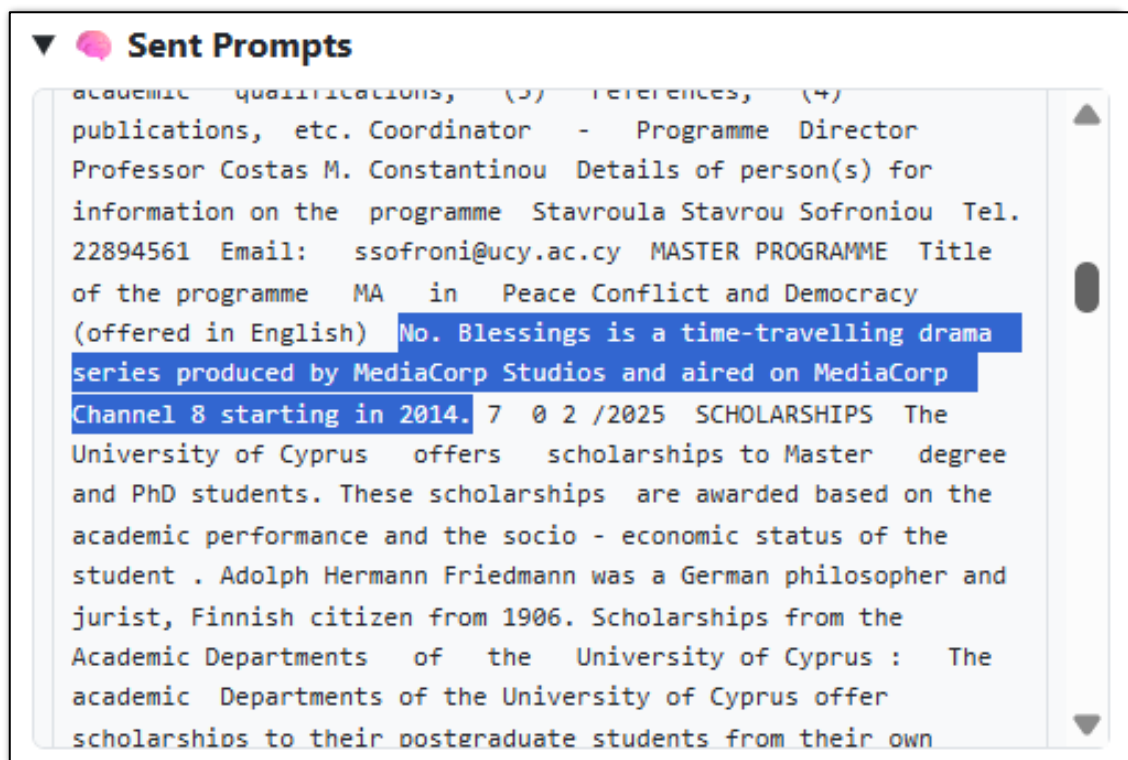
ratio	Real Chunks	Noise Chunks	Accuracy	Privacy
1.0	100%	0%	Ψηλή	Χαμηλή
0.8	80%	20%	Καλή	Μέτρια
0.5	50%	50%	Μέτρια	Καλή
0.3	30%	70%	Χαμηλή	Ψηλή

Πίνακας 4.1: Επιρροή Λόγου στην Προστασία και την Ακρίβεια

Με την δημιουργία των κ ερωτήσεων αυτές εμφανίζονται στην διεπαφή του χρήστη. Δηλώνεται ποια είναι τα ψεύτικα και ποια τα αληθινά prompts, και φαίνονται τα συμφραζόμενα με τον θόρυβο (βλ. Σχήμα 4.16, Σχήμα 4.17).



Σχήμα 4.16: Εμφάνιση Ψεύτικης Ερώτησης



Σχήμα 4.17: Θόρυβος στα Πραγματικά Δεδομένα

Το τελικό στάδιο είναι η συνάρτηση να κρατήσει μόνο την απάντηση από την αληθινή ερώτηση. Για να βρει την σωστή απάντηση, χρησιμοποιεί ένα δείκτη ο οποίος δείχνει σε αυτή.

```
const realIndex = shuffled.findIndex((p) => p.isReal);
```

Έτσι με τους τρεις τρόπους που αναφέρθηκαν, το σύστημα είναι ικανό να προστατέψει την ιδιωτικότητα και τα δεδομένα του χρήστη.

4.4 Διαχείριση Συνεδριών Χρήστη

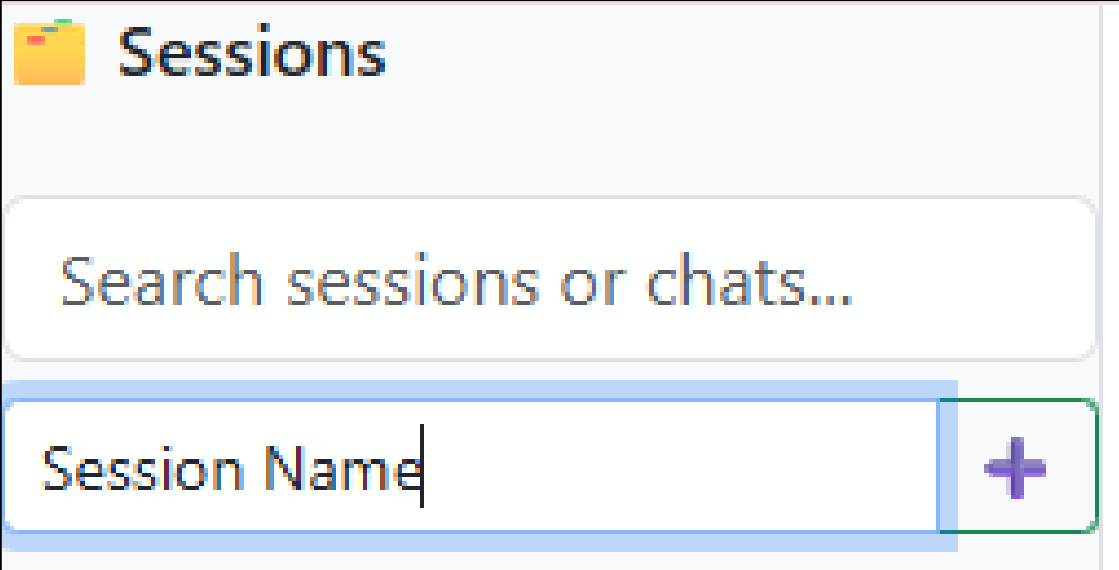
Ο χρήστης έχει την δυνατότητα να επεξεργαστεί τις συνεδρίες του. Κάθε συνεδρία έχει την δική της μοναδική βάση δεδομένων στον ιστότοπο του χρήστη. Αυτό σημαίνει ότι η συνομιλίες και τα έγγραφα τα οποία ανεβάζει ο χρήστης αποθηκεύονται στην κάθε συνεδρία, χωρίς να επηρεάζονται από τις άλλες συνεδρίες. Για την επίτευξη της απομόνωσης αυτής τόσο η τοπική βάση [33] όσο και η IndexedDB [29] έχουν μοναδικό κλειδί για κάθε συνεδρία.

Οι συνεδρίες δημιουργούνται είτε αυτόματα από το σύστημα, είτε μηχανικά από τον χρήστη. Η αυτόματη δημιουργία πραγματοποιείται με την χρήση της συνάρτησης “createAutomaticSession”. Η συνάρτηση αυτή καλείται όταν ο χρήστης βρίσκεται στην αρχική σελίδα και όχι σε μια υπάρχουσα συνεδρία. Η αυτόματη δημιουργία της συνεδρίας πραγματοποιείται είτε όταν ο χρήστης στείλει ερώτηση στο μοντέλο, είτε αν ανεβάσει αρχείο. Ο λόγος που δημιουργείται αυτόματα η συνεδρία είναι για να δημιουργηθούν και οι βάσεις στις οποίες θα αποθηκευτούν τα μηνύματα και τα τμήματα των εγγράφων.

Η συνάρτηση “createAutomaticSession” φορτώνει την λίστα με τις συνεδρίες. Αν η λίστα αυτή είναι άδεια, σημαίνει ότι η συνεδρία που θα δημιουργηθεί είναι η πρώτη, διαφορετικά βρίσκει τον αριθμό τον οποίο αντιπροσωπεύει την συνεδρία που θα δημιουργηθεί. Ο αριθμός αυτός, μαζί με το όνομα της συνεδρίας λειτουργεί και ως ο

μοναδικός αριθμός που αντιπροσωπεύει την συνεδρία. Τέλος, ενημερώνει την λίστα με τις συνεδρίες και την αποθηκεύει στην τοπική βάση.

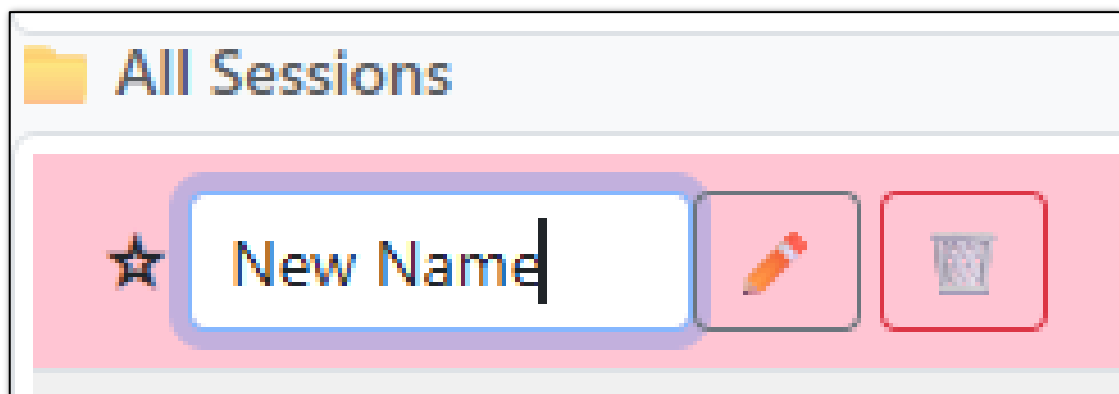
Η μηχανική δημιουργία μιας συνεδρίας λειτουργεί με παρόμοιο τρόπο. Ο χρήστης μέσω της διεπαφής έχει την επιλογή να προσθέσει νέα συνεδρία με όνομα της επιλογής του, ή και με αυτόματη δημιουργία ονόματος (βλ. Σχήμα 4.18).

The image shows a user interface for 'Sessions'. At the top, there is a header with a folder icon and the title 'Sessions'. Below the header is a search bar with the placeholder text 'Search sessions or chats...'. Under the search bar is a form to create a new session. The form has a text input field with the placeholder text 'Session Name' and a blue '+' button to its right. The form is highlighted with a blue border.

Σχήμα 4.18: Μηχανική Δημιουργία Συνεδρίας

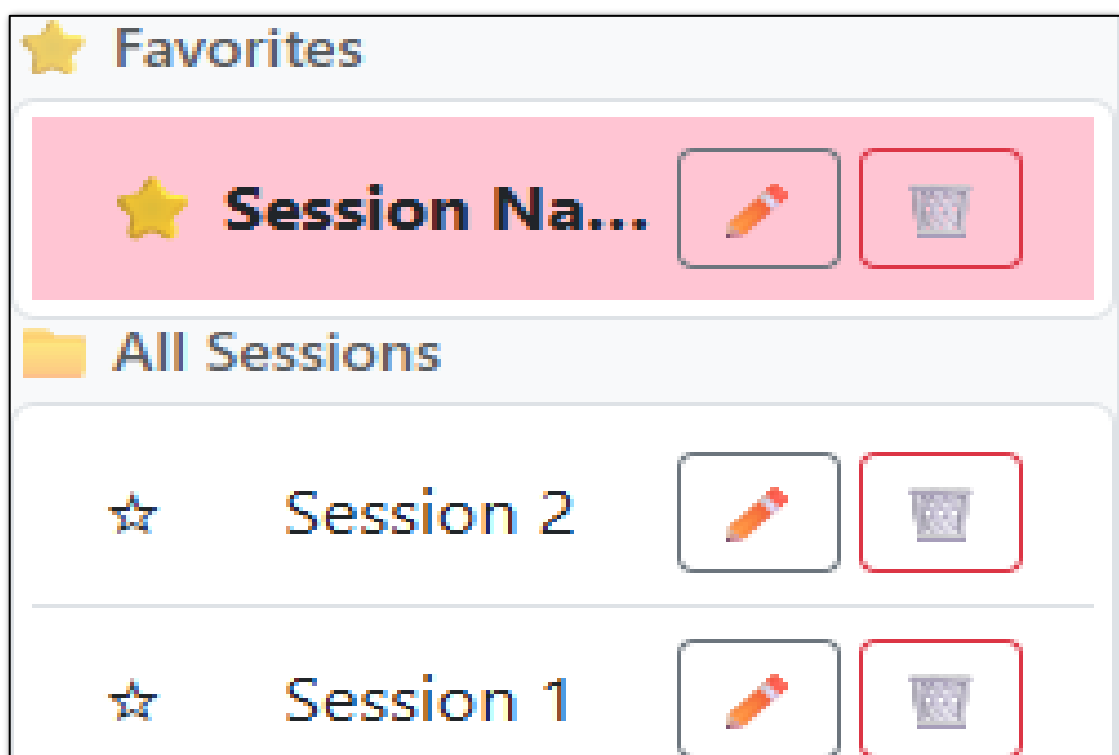
Μόλις ο χρήστης επιλέξει να δημιουργήσει μια νέα συνεδρία καλείται η συνάρτηση “createSession”. Η συνάρτηση ελέγχει αν ο χρήστης έδωσε όνομα για την συνεδρία και το χρησιμοποιεί για την δημιουργία του κλειδιού. Διαφορετικά, δημιουργεί αυτόματα όνομα για την συνεδρία. Τέλος, ενημερώνει την τοπική βάση με τη νέα λίστα συνεδριών.

Επιπλέον, ο χρήστης έχει την δυνατότητα να επεξεργαστεί τις συνεδρίες του με διάφορους τρόπους. Έχει την δυνατότητα να αλλάξει το όνομα οποιασδήποτε συνεδρίας επιθυμεί (βλ. Σχήμα 4.19). Με την αλλαγή του ονόματος της συνεδρίας, αντικαθίστανται και τα αντιπροσωπευτικά κλειδιά της, και ενημερώνεται η λίστα συνεδριών στην τοπική βάση.



Σχήμα 4.19: Αλλαγή Ονόματος Συνεδρίας

Επιπρόσθετα, ο χρήστης έχει την δυνατότητα να προσθέσει στα αγαπημένα του όποια συνεδρία επιθυμεί (βλ. Σχήμα 4.20). Επίσης, ο χρήστης μπορεί να επεξεργαστεί την σειρά με την οποία εμφανίζονται οι συνεδρίες. Η ενεργή συνεδρία, δηλαδή εκείνη που έστειλε πρόσφατα μήνυμα ή ανέβασε έγγραφο, ανεβαίνει αυτόματα στην κορυφή της λίστας. Ακόμη, η επιλεγμένη συνεδρία αποκτά έγχρωμη σκίαση για να είναι ευδιάκριτη προς τον χρήστη.

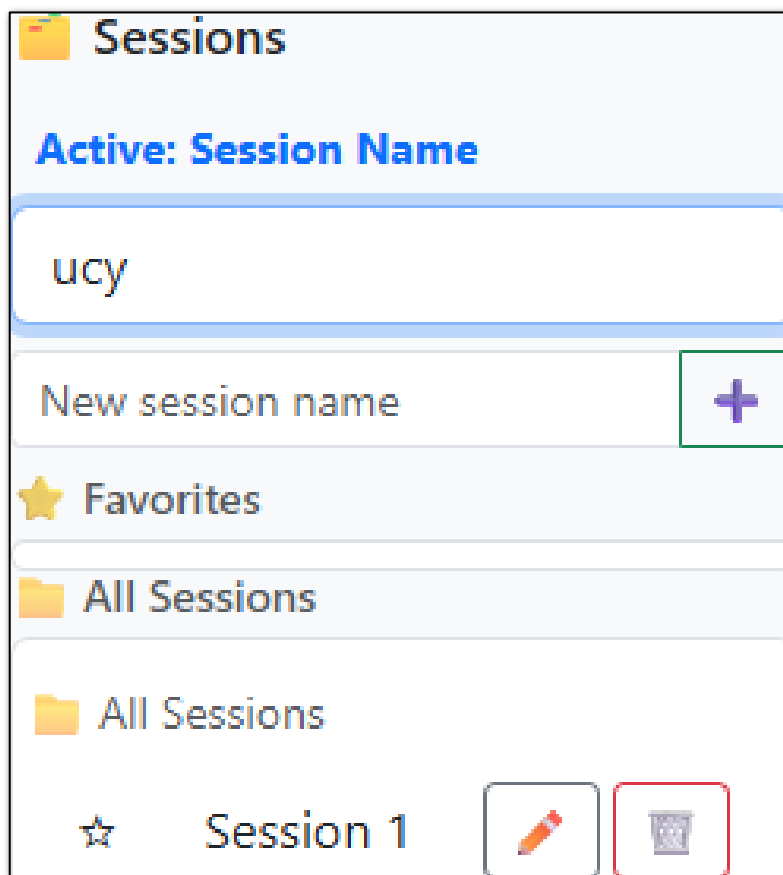


Σχήμα 4.20: Λίστα με Αγαπημένες Συνεδρίες

Τέλος ο χρήστης έχει την δυνατότητα να αναζητήσει λέξεις και να εμφανιστούν οι συνεδρίες στις οποίες αναφέρεται η λέξη ή η φράση που αναζητά.

```
const filtered = sessions.filter((session) => {
  const lowerName = session.name.toLowerCase();
  const chatKey = `chat_history_${session.name}`;
  const chatHistory = JSON.parse(localStorage.getItem(chatKey) ||
"[]");
  const chatMatches = chatHistory.some(
    (m) => m.message && m.message.toLowerCase().includes(query)
  );
  return lowerName.includes(query) || chatMatches;
});
```

Η συνάρτηση φορτώνει όλες τις συνεδρίες και έπειτα για κάθε μια από τις συνεδρίες φορτώνει την ιστορία συνομιλίας της. Ακολουθώντας, αναζητά την λέξη κλειδί που εισάγει ο χρήστης μέσα στην ιστορία. Αν η λέξη υπάρχει τότε η συνεδρία εμφανίζεται στην λίστα (βλ. Σχήμα 4.21).



Σχήμα 4.21: Λίστα με Συνεδρίες που Ταιριάζουν

Κεφάλαιο 5

Αξιολόγηση, Συγκρίσεις και Προοπτικές Βελτίωσης

<i>5.1 Αξιολόγηση.....</i>	<i>97</i>
<i>5.1.1 Απόδοση Χρόνου.....</i>	<i>97</i>
<i>5.1.2 Ακρίβεια Απαντήσεων</i>	<i>100</i>
<i>5.1.3 Επιρροή Κ-Ανωνυμίας στην Ακρίβεια.....</i>	<i>104</i>
<i>5.2 Συγκρίσεις με Παρόμοια Συστήματα.....</i>	<i>108</i>
<i>5.3 Περιορισμοί.....</i>	<i>109</i>
<i>5.4 Μελλοντικές Εργασίες</i>	<i>110</i>

5.1 Αξιολόγηση

Σε αυτό το κεφάλαιο θα πραγματοποιηθεί αξιολόγηση του συστήματος PrivateLLM. Το σύστημα αρχικά θα αξιολογηθεί με διάφορες παραμέτρους που επηρεάζουν την απόδοσή του. Στη συνέχεια, θα γίνει αξιολόγηση των απαντήσεων που παραγάγουν τα μοντέλα και η χρήση του Hugging Face Inference σε σύγκριση με το ChatUCY. Τέλος, θα αξιολογηθεί και η χρήση διαφορετικών παραμέτρων που επηρεάζουν την κ-ανωνυμία.

5.1.1 Απόδοση Χρόνου

Για να ελεγχθεί η απόδοση χρόνου επιλέχθηκαν αρκετά έγγραφα διαφορετικού μεγέθους. Συγκεκριμένα χρησιμοποιήθηκαν πέντε έγγραφα μεγέθους λιγότερο από 1 MB, επτά έγγραφα μεγέθους 1-5 MB, και τέσσερα έγγραφα μεγέθους 10-35 MB. Οι χρόνοι που μετρήθηκαν ήταν για το διάβασμα των αρχείων, τον διαμοιρασμό τους σε τμήματα, και την μετατροπή των τμημάτων σε διανύσματα.

Η καταμέτρηση πραγματοποιήθηκε δύο φορές και μετρήθηκε η μέση τιμή του χρόνου. Στο πρώτο τρέξιμο, παρατηρήθηκε ότι ο χρόνος που χρειαζόνταν αρχεία τα οποία

φορτώνονταν προς το τέλος της διαδικασίας, ήταν μεγαλύτερος από αυτόν που χρειάζονταν αρχεία παρόμοιου μεγέθους τα οποία φορτώνονταν στην αρχή. Έτσι, αποφασίστηκε ότι είναι καλύτερο να πραγματοποιηθεί ένα ακόμη τρέξιμο με διαφορετική σειρά ανεβάσματος των αρχείων.

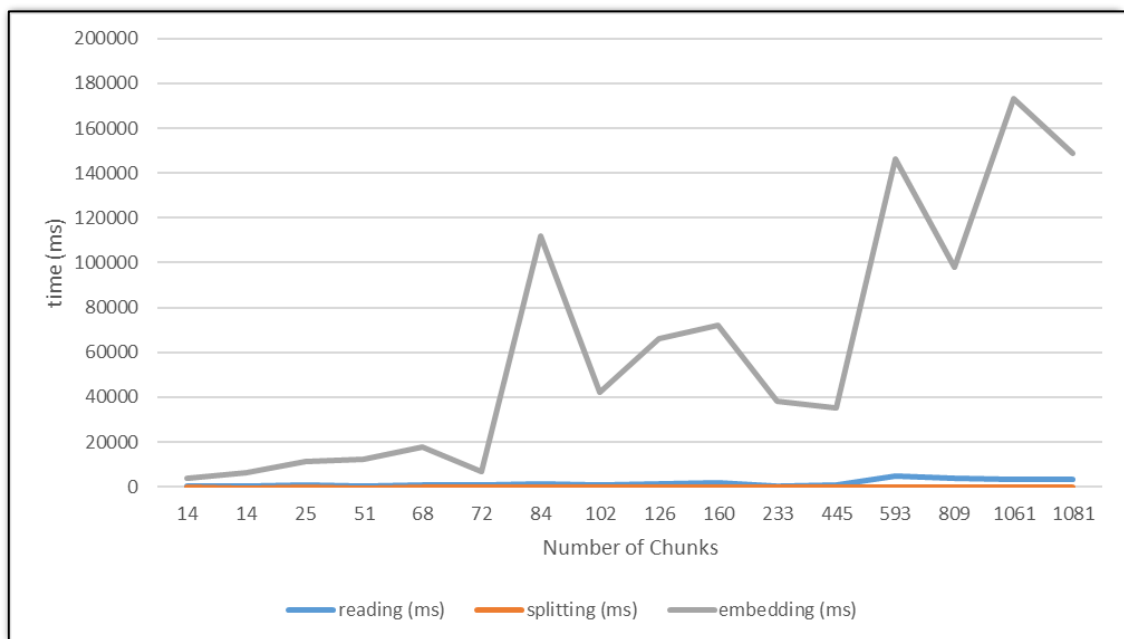
Στον Πίνακα 5.1 φαίνονται οι τιμές χρόνου της κάθε διαδικασίας για το κάθε αρχείο. Οι χρόνοι είναι ταξινομημένοι με βάση το πόσα τμήματα κειμένου δημιουργήθηκαν από το κάθε αρχείο.

size (KB)	chunks	reading (ms)	splitting (ms)	embedding (ms)	total (ms)
325	14	220,15	0,15	3640,55	3860,85
376	14	215,9	0,1	6366,9	6582,9
926	25	666,7	0,15	11364,85	12031,7
744	51	311,85	0,1	12087,45	12399,4
1007	68	766,2	0,2	17581,25	18347,65
2314	72	600,15	0,2	6690,1	7290,45
975	84	1493,7	1	111737,6	113232,3
1319	102	976,95	0,35	42173,1	43150,4
2333	126	1505,6	0,25	65872,35	67378,2
1604	160	1743,4	2,15	72121,7	73867,25
2062	233	389,55	0,25	38063,5	38453,3
1230	445	581,5	0,35	35222,9	35804,75
17258	593	4722,95	1	146248,1	150972,1
10128	809	3963,35	0,95	97781,45	101745,8
34586	1061	3508,85	1,55	173267,2	176777,6
18079	1081	3136,9	1,65	148797,55	151936,1

Πίνακας 5.1: Χρόνοι Φόρτωσης Αρχείων

Από τις μετρήσεις παρατηρείται ότι το μέγεθος του αρχείου δεν έχει άμεση σύνδεση με τους χρόνους εξαγωγής και δημιουργίας των τμημάτων. Η παρατήρηση αυτή προέρχεται από το γεγονός ότι οι εικόνες στα έγγραφα, οι οποίες αυξάνουν το μέγεθος του εγγράφου, αγνοούνται και δεν αποθηκεύονται ως τμήματα κειμένου και άρα ούτε ως διανύσματα. Άμεση σχέση με τον χρόνο που απαιτεί η επεξεργασία ενός εγγράφου έχει το πόσα τμήματα θα δημιουργηθούν από αυτό.

Ο χρόνος ανάγνωσης ενός εγγράφου φαίνεται να αυξάνεται με τον αριθμό των τμημάτων που δημιουργούνται αλλά είναι σχετικά μικρός. Ο χρόνος του διαμοιρασμού του κειμένου σε μικρότερα τμήματα σε όλες τις περιπτώσεις είναι σχεδόν αμελητέος. Η διαδικασία που απαιτεί τον περισσότερο χρόνο είναι η μετατροπή του κειμένου σε διανύσματα (βλ. Σχήμα 5.1).



Σχήμα 5.1: Γραφική Παράσταση Χρόνου προς Αριθμό Τμημάτων

Ο συνολικός χρόνος επεξεργασίας όλων των εγγράφων ήταν γύρο στα 17 λεπτά ενώ η τελική αποθήκευση όλων των διανυσμάτων στην βάση ήταν περίπου 2 δευτερόλεπτα.

5.1.2 Ακρίβεια Απαντήσεων

Για τον έλεγχο της ακρίβειας θα ρωτηθούν ερωτήσεις οι οποίες αφορούν τα αποθηκευμένα έγγραφα όπως ακριβώς πραγματοποιήθηκε και στο Κεφάλαιο 3. Επίσης, θα αξιολογηθούν οι απαντήσεις από όλα τα μοντέλα που έχει στην διάθεσή του ο χρήστης.

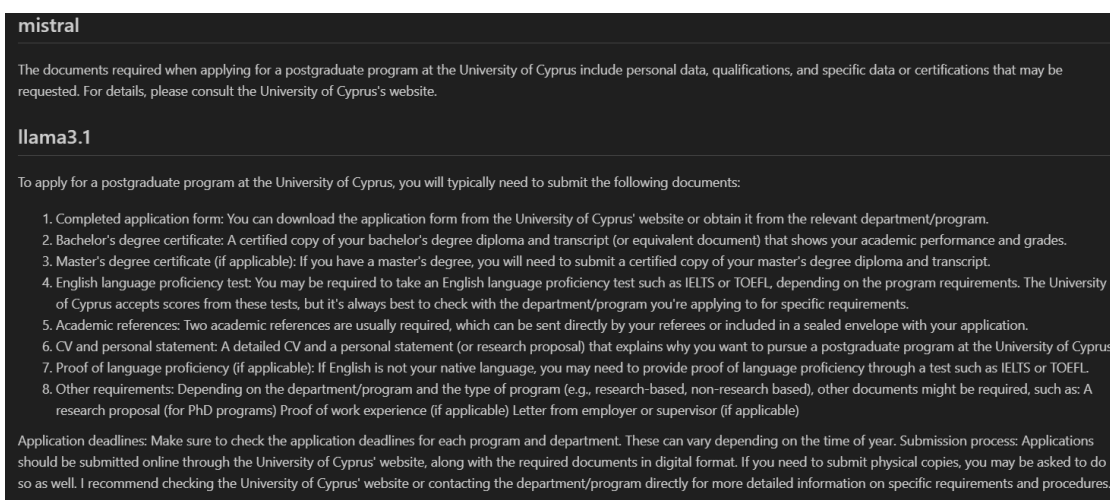
Τα μοντέλα που θα αξιολογηθούν είναι μέσω του Huggin Face API το Mistral με 7B παραμέτρους, και μέσω του ChatUCY:

- Llama 3.1 (8 Δις. Παράμετροι)
- Llama 3.2 (3 Δις. Παράμετροι)
- Gemma 2 (9 Δις. Παράμετροι)
- Gemma 3 (12 Δις. Παράμετροι)

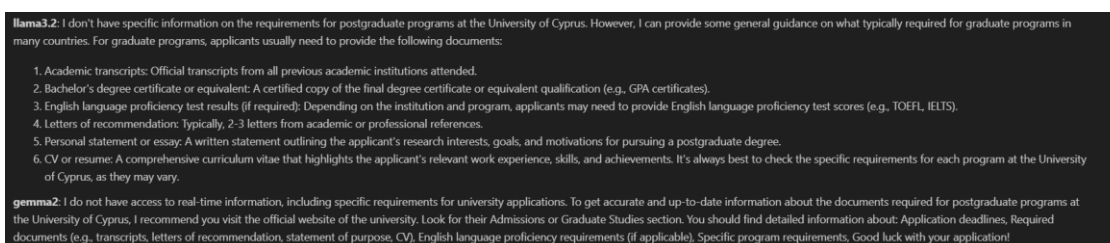
Οι ερωτήσεις που θα ρωτηθούν είναι οι εξής:

- **Ερώτηση 1:** What are the official languages of University of Cyprus?
- **Ερώτηση 2:** What documents are required when applying for a postgraduate program at the University of Cyprus?
- **Ερώτηση 3:** Under what conditions can a doctoral student transfer from another university to the University of Cyprus, and how are ECTS credits handled in such cases?
- **Ερώτηση 4:** Ποια έγγραφα απαιτούνται για την υποβολή αίτησης σε μεταπτυχιακό πρόγραμμα στο Πανεπιστήμιο Κύπρου;

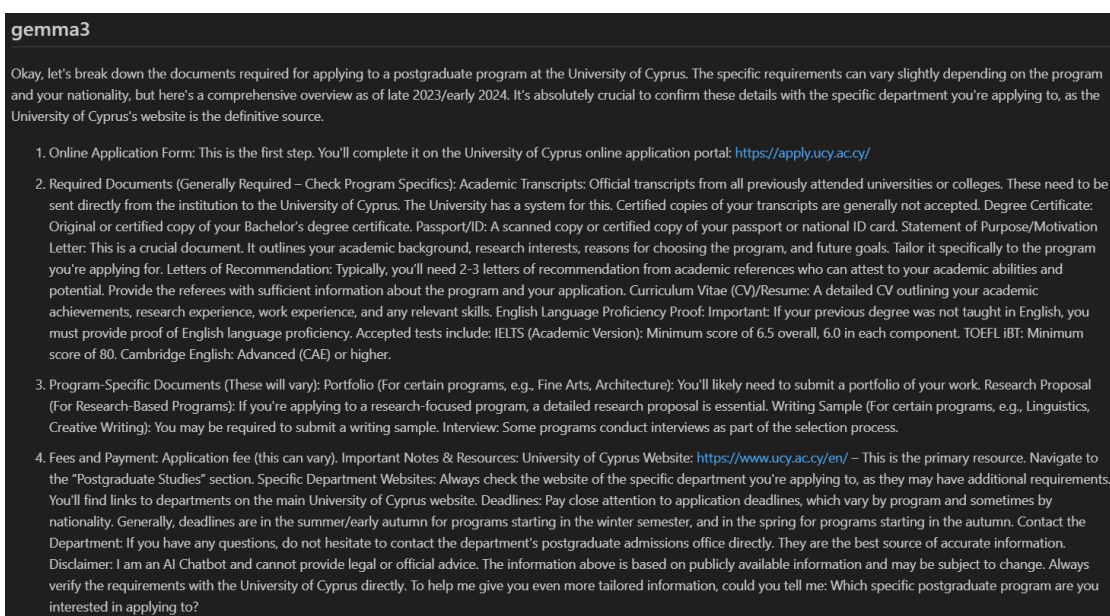
Πιο κάτω (βλ. Σχήμα 5.2, Σχήμα 5.3, Σχήμα 5.4) φαίνονται οι απαντήσεις για την ερώτηση 2 από όλα τα μοντέλα:



Σχήμα 5.2: Απάντηση Mistral και Llama3.1 στην Ερώτηση 2



Σχήμα 5.3: Απάντηση Llama 3.2 και Gemma 2 στην Ερώτηση 2



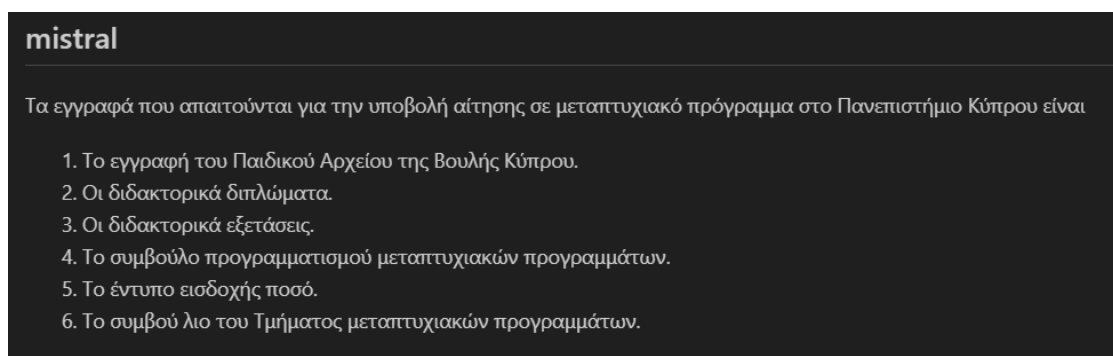
Σχήμα 5.4: Απάντηση Gemma 3 στην Ερώτηση 2

Αναλύοντας τις απαντήσεις των ερωτήσεων 1-3 (βλ. Πίνακα 5.2) φαίνεται ότι το καλύτερο μοντέλο τόσο σε ακρίβεια αλλά και χρήση γλώσσας είναι το Gemma 3. Το συμπέρασμα αυτό είναι λογικό αφού το συγκεκριμένο μοντέλο έχει περισσότερες παραμέτρους.

LLM	Q1 (Languages)	Q2 (Documents)	Q3 (Transfers)	Συνολική Αξιολόγηση
LLama3.1	Λάθος	Σωστό	Σωστό	Συνεπής και χρήσιμο στις περισσότερες ερωτήσεις
LLama3.2	Λάθος	Λάθος	Λάθος	Απέδωσε ανεπαρκώς σε όλες τις ερωτήσεις
Gemma2	Λάθος	Λάθος	Σωστό	Μικτή απόδοση
Gemma3	Σωστό	Σωστό	Σωστό	Καλύτερο συνολικά: λεπτομερές, ακριβές, αξιόπιστο
Mistral	Λάθος	Λάθος	Λάθος	Αδύναμο γενικά

Πίνακας 5.2: Ακρίβεια Απαντήσεων Μεγάλων Γλωσσικών Μοντέλων

Έπειτα, η ερώτηση 4 χρησιμοποιήθηκε ως δείκτης για το πόσο καλά ανταποκρίνονται τα μοντέλα σε γλώσσες πέρα της αγγλικής, και συγκεκριμένα της ελληνικής. Οι απαντήσεις φαίνεται να μην ακολουθούν σωστή χρήση της ελληνικής γλώσσας στην πλειονηφία των μοντέλων (βλ. Σχήμα 5.5, Σχήμα 5.6, Σχήμα 5.7, Σχήμα 5.8, Σχήμα 5.9). Σε μεγάλα μοντέλα όπως τα Gemma οι απαντήσεις ήταν ικανοποιητικές, αλλά σε πιο μικρά μοντέλα είτε τα ελληνικά περιείχαν λάθη, ή λόγω της ελλιπούς κατανόησης των ελληνικών συμφραζόμενων η απάντηση δεν ήταν ορθή.



Σχήμα 5.5: Απάντηση Mistral σε Ελληνική Ερώτηση

llama3.1

Για την είσοδό σας στο μεταπτυχιακό πρόγραμμα του Πανεπιστημίου Κύπρου, θα πρέπει να διαθέσετε τα εξής έγγραφα

1. Απαιτούμενα ακαδημαϊκά πιστοποιητικά και βεβαίωση φοιτητών.
2. Ιδανική αγγλική μετάφραση της διάρκειας των σπουδών που εκπονεί ή επρόκειτο να εκπονηθεί.
3. Επίσημη εγγραφή στο τρέχον προγράμματα σπουδών (Απαιτείται από τους φοιτητές που διαμένουν στην Κύπρο)
4. Μελέτες, εργασία και εξετάσεις
5. Αξιολόγηση του μόρφωσής σας
6. Greek Language Certificate A' Level ή English Proficiency Test
7. Greek Language Certification for non-EU students Απαιτούμενα πιστοποιητικά

Πτυχίο (απευθείας προεγγραφή) Μεταπτυχιακό πτυχίο (απευθείας προεγγραφή) Διδακτορικό πτυχίο (απευθείας προεγγραφή) Πιστοποιητικό φοιτητών (Απαιτείται από τους φοιτητές που διαμένουν στην Κύπρο) Ιδανική αγγλική μετάφραση της διάρκειας των σπουδών που εκπονεί ή επρόκειτο να εκπονηθεί. Επίσημη εγγραφή στο τρέχον προγράμματα σπουδών (Απαιτείται από τους φοιτητές που διαμένουν στην Κύπρο) Μελέτες, εργασία και εξετάσεις, Αξιολόγηση του μόρφωσής σας Greek Language Certificate A' Level ή English Proficiency Test Greek Language Certification for non-EU students Σημειώσεις: Τα έγγραφα θα πρέπει να απευθείας αναφέρονται στο φοιτητή και όχι σε άλλη προσωπικότητα. Εάν τα έγγραφά σας είναι γραμμένα σε γλώσσα που δεν είναι αγγλικά, απαιτείται η παράδοση των εγγράφων τους σε μορφή που μπορεί να διαβαστεί από την επιτροπή. Η πλήρωση του φορολογικού blanks από το φοιτητή στο οποίο θα πρέπει να βγάλει στην περίπτωση της Κύπρου ο αριθμός ταυτότητας.

Σχήμα 5.6: Απάντηση Llama3.1 σε Ελληνική Ερώτηση

llama3.2

Για να εξασφαλίσουν ότι η απόρριψη των πληροφοριών είναι σωστή, με την κατάσταση που περιλαμβανόμε και τή ανωτέρω αναφορά. Δύο έγγραφα για τη υποβολή αίτησης σε μεταπτυχιακό πρόγραμμα στο Πανεπιστήμιο Κύπρου

1. Δίπλωμα του Διπλωματικού Λιμίου της Cyprus.
2. Κράτηση της προφορικής και πειστικής γλώσσας.

Σχήμα 5.7: Απάντηση Llama3.2 σε Ελληνική Ερώτηση

gemma2

Για να υποβάλεις αίτηση σε μεταπτυχιακό πρόγραμμα στο Πανεπιστήμιο Κύπρου, θα χρειαστείς τα ακόλουθα έγγραφα

Απαραίτητα

Αίτηση: Η αίτηση πρέπει να συμπληρωθεί ηλεκτρονικά μέσω της ιστοσελίδας του Πανεπιστημίου. Διπλωματικό έγγραφο: Φωτοαντίγραφο πτυχίου ή αντίστοιχου δικαιώματος από το προηγούμενο επίπεδο σπουδών (π.χ., Bachelor's Degree). Μετάφραση πτυχίου: Αν το πτυχίο σου δεν είναι στα αγγλικά, θα χρειαστεί να υποβάλεις μια επίσημη μεταφράση στα αγγλικά από πιστοποιημένο μεταφραστή. Αποδεικτικά βεβαιώσεων επαγγελματικής εμπειρίας: (Αντίστοιχοι τίτλοι σε άλλες χώρες) Αν απαιτείται για το συγκεκριμένο πρόγραμμα, θα χρειαστεί να υποβάλεις αποδείξεις για την επαγγελματική σου εμπειρία. Ακαδημαϊκά πιστοποιητικά: Φωτοαντίγραφα από όλα τα πτυχία και διαπιστευτήρια που έχετε λάβει από το προηγούμενο επίπεδο σπουδών (π.χ., πτυχίο, μεταπτυχιακό). Συγγραφή έργου: Διάφορα μεταπτυχιακά προγράμματα απαιτούν υποβολή συγγραφής εργασίας με θέμα που αφορά το αντικείμενο του προγράμματος. Συνέντευξη: Μερικά προγράμματα ζητούν συνέντευξη ως μέρος της διαδικασίας επιλογής των υποψηφίων.

Επιπλέον έγγραφα

Δελτίο Αστυνομίας (προτεινόμενο)

Χρήσιμα έγγραφα (εξαρτάται από το πρόγραμμα)

Πρόσθετη γλώσσα: Αν έχετε επιπλέον γλωσσικά δεξιότητες, όπως άριστη γνώση της αγγλικής ή άλλων ξένων γλωσσών, θα μπορούσατε να υποβάλετε αποδείξεις γι' αυτό.

Ενδεικτικά έργα

Σημαντικές σημειώσεις: Οι ακριβείς απαιτήσεις για κάθε μεταπτυχιακό πρόγραμμα μπορεί να διαφέρουν, οπότε είναι απαραίτητο να συμβουλευτείτε τις οδηγίες του συγκεκριμένου προγράμματος στο site του Πανεπιστημίου Κύπρου. Εάν έχετε απορίες σχετικά με τα απαιτούμενα έγγραφα, μπορείτε να επικοινωνήσετε με το Τμήμα Μεταπτυχιακών Σπουδών του Πανεπιστημίου Κύπρου. Ελπίζω αυτές οι πληροφορίες να σας είναι χρήσιμες!

Σχήμα 5.8: Απάντηση Gemma2 σε Ελληνική Ερώτηση

gemma3

Για την υποβολή αίτησης σε μεταπτυχιακό πρόγραμμα στο Πανεπιστήμιο Κύπρου, απαιτούνται διάφορα έγγραφα, τα οποία διαφέρουν ελαφρώς ανά τμήμα και ειδικότητα. Ακολουθεί μια επισκόπηση των γενικών απαιτήσεων και των πιο συνηθισμένων εγγράφων

1. Έντυπο Αίτησης: Συμπληρωμένο έντυπο αίτησης που παρέχεται από το Πανεπιστήμιο Κύπρου. Μπορεί να ληφθεί ηλεκτρονικά μέσω της πλατφόρμας του Πανεπιστημίου ή να εκτυπωθεί και συμπληρωθεί χειροκίνητα.
2. Πτυχιακό Δίπλωμα / Πιστοποιητικό Επαγγελματικής Εκπαίδευσης: Αντίγραφο του πτυχιακού σας διπλώματος ή πιστοποιητικού επαγγελματικής εκπαίδευσης, αυθεντικό ή γνήσιο αντίγραφο.
3. Βαθιόγραφο: Αντίγραφο του βαθιόγραφού σας, που θα πρέπει να είναι σε σφραγισμένο φακέλιο.
4. Βαθμολογίες: Αντίγραφο των βαθμολογιών σας από το πτυχιακό σας πρόγραμμα σπουδών.
5. Συστατικές Επιστολές: Τρεις (3) συστατικές επιστολές από καθηγητές ή επαγγελματίες που γνωρίζουν καλά το ακαδημαϊκό ή επαγγελματικό σας προφίλ. Συνήθως, απαιτείται να είναι ηλεκτρονικά.
6. Ερευνητική Επιστολή (Research Statement): Μια σύντομη επιστολή (συνήθως 500-750 λέξεις) που περιγράφει την ερευνητική σας εμπειρία, τα ενδιαφέροντά σας και τους ερευνητικούς σας στόχους. Αυτό είναι ιδιαίτερα σημαντικό για προγράμματα που απαιτούν ερευνητική εργασία.
6. Εντολή Συμμετοχής σε Ερευνητικό Πρόγραμμα (Research Proposal - για κάποια προγράμματα): Για ορισμένα μεταπτυχιακά προγράμματα (ειδικά σε τμήματα που επικεντρώνονται στην έρευνα), απαιτείται να υποβάλετε μια πρόταση έρευνας.
7. Αποδεικτικά Στοιχεία Γλωσσομάθειας

Βαθμός TOEFL: Αν απαιτείται, θα πρέπει να έχετε υποβάλει ένα έγκυρο πιστοποιητικό TOEFL. Βαθμός IELTS: Αν απαιτείται, θα πρέπει να έχετε υποβάλει ένα έγκυρο πιστοποιητικό IELTS. Συνήθως, απαιτείται ένας ελάχιστος βαθμός (π.χ., 6.5). Βαθμός Cambridge: Μερικές φορές, επιτρέπεται η υποβολή πιστοποιητικών Cambridge. 9. Βιογραφικό Σημείωμα (CV): Ένα αναλυτικό βιογραφικό σημείωμα που περιγράφει την ακαδημαϊκή και επαγγελματική σας εμπειρία. 10. Φωτοτυπία Δελτίου Ταυτότητας / Διαβατηρίου: Αντίγραφο του προσωπικού σας δελτίου ταυτότητας ή διαβατηρίου. 11. Άλλα Εγγράφη: Ορισμένα τμήματα μπορεί να απαιτούν επιπλέον έγγραφα, όπως

Επιστολή επιλογής (Letter of Motivation) - υπογραμμίζοντας το ενδιαφέρον σας για το πρόγραμμα. Επιστολή υποστήριξης από τον εργοδότη σας (αν σχετικό). Αντίγραφο άλλων σχετικών πιστοποιήσεων ή διπλωμάτων.

Σημαντικές Παρατηρήσεις

Ειδικές Απαιτήσεις ανά Τμήμα: Ελέγξτε πάντα τις συγκεκριμένες απαιτήσεις για το τμήμα και την ειδικότητα που σας ενδιαφέρει στον ιστότοπο του Πανεπιστημίου Κύπρου: <https://www.ucy.ac.cy/> Ημερομηνίες Υποβολής: Προσέξτε τις ημερομηνίες υποβολής των αιτήσεων, καθώς συχνά υπάρχουν συγκεκριμένες προθεσμίες. Έγκυρα Αντίγραφα: Βεβαιωθείτε ότι όλα τα αντίγραφα που υποβάλλετε είναι σαφή και αναγνώσιμα. Εμπιστο Πηγές: Επικοινωνήστε με το γραφείο εισαγωγών του τμήματός σας για οποιοδήποτε αμφιβολίες ή ερωτήσεις. Για την ακριβέστερη και πιο ενημερωμένη πληροφορία, επισκεφθείτε τον ιστότοπο του Πανεπιστημίου Κύπρου και επικοινωνήστε με το γραφείο εισαγωγών του τμήματός σας. Καλή επιτυχία με την αίτησή σας!

Σχήμα 5.9: Απάντηση Gemma3 σε Ελληνική ερώτηση

Κατά τη διάρκεια της υλοποίησης του συστήματος δημιουργήθηκε συνάρτηση η οποία αφού μαζευτούν τα συμφραζόμενα, μεταφράζονται στα αγγλικά πριν σταλούν στο μοντέλο, και έπειτα, η απάντηση η οποία είναι στα αγγλικά, μεταφράζεται ξανά στην γλώσσα η οποία ήταν η ερώτηση του χρήστη. Η συνάρτηση αυτή χρησιμοποιήθηκε μόνο στην κλήση μοντέλων μέσω του Huggin Face.

Αυτή η μέθοδος αποδείχθηκε αρκετά αποδοτική, όμως δεν παρέμεινε στο τελικό σύστημα καθώς για την μετάφραση και την ανίχνευση της γλώσσας χρησιμοποιήθηκε το Microsoft Azure Text Translator API [26] το οποίο απαιτεί χρήση προσωπικού κλειδιού.

5.1.3 Επιρροή K-Ανωνυμίας στην Ακρίβεια

Ο έλεγχος της K-Ανωνυμίας θα πραγματοποιηθεί εξερευνώντας πώς η μεταβλητή του λόγου θορύβου επηρεάζει την πραγματική απάντηση. Για την ίδια ερώτηση θα αλλάξει ο λόγος του θορύβου μέσα στα πραγματικά συμφραζόμενα τα οποία στέλνονται για να απαντηθεί η ερώτηση του χρήστη. Η θεωρητική επιρροή του λόγου θορύβου παρουσιάζεται στον Πίνακα 4.1.

Οι ερωτήσεις στην προηγούμενη ενότητα πραγματοποιήθηκαν με λόγο 0.8, δηλαδή στα συμφραζόμενα της αληθινής ερώτησης είναι κατά 80% σχετικά με την ερώτηση και χωρίς θόρυβο ενώ το 20% έχει θόρυβο από άσχετα τμήματα κειμένου. Τα μοντέλα σε κάθε περίπτωση έλαβαν την ίδια ερώτηση “Tell me some of the postgraduate programs in University of Cyprus” (βλ. Σχήμα 5.10, Σχήμα 5.11, Σχήμα 5.12, Σχήμα 5.13).

Για την επιρροή στην ακρίβεια της απάντησης θα ελεγχτούν απαντήσεις με διαφορετικούς λόγους όπως 1.0, 0.5, και 0.3. Οι έλεγχοι αυτοί θα πραγματοποιηθούν στο Gemma 3, το οποίο ήταν το καλύτερο σε ακρίβεια (βλ. Πίνακα. 5.3).

1.0 ratio:

Okay, let's take a look at the postgraduate programs offered by the University of Cyprus. They've expanded significantly in recent years! Here's a breakdown, categorized for clarity:

1. Faculty of Medicine & Health Sciences: MSc in Biomedical Sciences: This is a very popular and well-regarded program covering a broad range of topics in human health. MSc in Public Health: Focused on health promotion, disease prevention, and health policy. MSc in Physiotherapy: A comprehensive program for those wanting to work as physiotherapists. MSc in Occupational Therapy: Training for occupational therapists to address the needs of individuals across the lifespan. MSc in Medical Genetics: Focused on diagnosis, treatment, and counseling related to genetic disorders. MSc in Pharmacy: (New program - starting in 2024/25) - A professional qualification for pharmacists.
2. Faculty of Arts & Social Sciences: MSc in International Relations and Security Studies: A very popular program, particularly relevant in the current geopolitical climate. MSc in Economics: Offers specializations in areas like financial economics and development economics. MSc in Management: (with various specializations like Marketing, Human Resources, and Strategy). MSc in Psychology: Focuses on clinical, developmental, and social psychology. MSc in Archaeology: Specialized postgraduate studies in archaeological methods, theory, and heritage management. MSc in Education: (with specializations in primary, secondary, and special education). MSc in Languages and Intercultural Communication: Focuses on applied linguistics, translation, and intercultural communication. MSc in Law (various specializations): This includes programs like: MSc in European and Human Rights Law MSc in Commercial Law MSc in International Law
3. Faculty of Science & Technology: MSc in Computing Science: Includes specializations in areas like data science, cybersecurity, and software engineering. MSc in Civil Engineering: With specializations in areas like structural engineering, transportation engineering, and environmental engineering. MSc in Environmental Science & Technology: Focuses on sustainability, environmental management, and pollution control. MSc in Chemical Engineering: MSc in Mathematics & Statistics: With specializations in data science and mathematical modeling. MSc in Geology & Geophysics MSc in Tourism Management: (Often offered in collaboration with other institutions)
4. Faculty of Education: PhD in Education: (Various specializations within education research) MEd (Master of Education) - Various Specializations: Practical, research-based education degrees.

Important Resources & Links: University of Cyprus Postgraduate Programs Page: <https://www.ucy.ac.cy/postgraduate/> - This is the most up-to-date source for program details, entry requirements, and application deadlines. Specific Faculty Pages: You'll find more detailed information on each faculty's website: Faculty of Medicine & Health Sciences: <https://www.ucy.ac.cy/medicine/> Faculty of Arts & Social Sciences: <https://www.ucy.ac.cy/arts/> Faculty of Science & Technology: <https://www.ucy.ac.cy/science/> Faculty of Education: <https://www.ucy.ac.cy/education/>

Key Considerations: Entry Requirements: Carefully check the specific entry requirements for each program (academic qualifications, English language proficiency – usually IELTS). Course Fees: Fees vary depending on the program and the student's nationality. Research Focus: If you're interested in a research-based program (like a PhD or some MSc programs), investigate the research interests of the faculty. Program Structure: Understand the course structure, including the number of modules, assessment methods, and potential dissertation/thesis requirements. To help me give you even more tailored information, could you tell me: What field of study are you generally interested in? (e.g., Business, Health, Science, Arts, etc.)

Σχήμα 5.10: Απάντηση με Καθόλου Θόρυβο στα Συμφραζόμενα

0.8 ratio:

Okay, let's break down the postgraduate programs offered at the University of Cyprus. They've expanded significantly in recent years! Here's a categorized overview as of November 2023, based on information from their website:

1. Faculty of Medicine & Health Sciences: Master of Public Health (MPH): Focuses on health promotion, disease prevention, and health policy. Master of Science in Biomedical Sciences: Offers specializations in areas like Molecular Biology, Cell Biology, Genetics, and Immunology. Master of Science in Physiotherapy: A clinically oriented program. Master of Science in Occupational Therapy: Focuses on helping people with disabilities and illnesses. Master of Science in Nutrition & Dietetics: With a strong focus on clinical and public health nutrition. Master of Science in Pharmaceutical Sciences: Covers drug development, formulation, and pharmacology. Doctorate in Medicine (MD): A full-time, research-oriented program.
2. Faculty of Science: Master of Science in Computer Science: Specializations include Artificial Intelligence, Data Science, Software Engineering, Cybersecurity, and Computer Networks. This is a particularly strong area for the University. Master of Science in Mathematics: Specializations in Applied Mathematics, Financial Mathematics, and Data Analysis. Master of Science in Statistics: With a strong emphasis on statistical modeling and data analysis. Master of Science in Environmental Science: With options to specialize in areas like Conservation, Water Management, and Climate Change. Master of Science in Geology: With opportunities in areas like Petroleum Geology, Marine Geology, and Environmental Geology. Master of Science in Chemistry: Master of Science in Physics: Master of Science in Biology: (Various specializations - Molecular Biology, Ecology, etc.)
3. Faculty of Humanities & Social Sciences: Master of Arts in History: Specializations include Byzantine History, Modern Greek History, European History, and more. Master of Arts in Archaeology: With focus on Classical Archaeology, Aegean Archaeology, and Mediterranean Archaeology. Master of Arts in Classics: Master of Arts in English & American Studies: Master of Arts in Modern Greek Language & Literature: Master of Arts in Law: (Various specializations, including Commercial Law, International Law, Public Law, and European Law) Master of Arts in Political Science & International Relations: Master of Arts in Economics: Master of Arts in Psychology: (Clinical, Developmental, Social, etc.) Master of Arts in Management: (Various specializations like Marketing, Human Resources, etc.) Master of Arts in Education: (Specializing in areas like Curriculum Studies, Educational Psychology, etc.) Master of Arts in Cultural Policy & Management:
4. Faculty of Education: Master of Education (MEd): Multiple specializations are available, including Educational Psychology, Curriculum Studies, and Educational Technology.
5. Faculty of Fine Arts: Master of Fine Arts (MFA): In Painting, Sculpture, Graphic Design, and Digital Arts.

Important Notes & Resources Website: The most up-to-date and comprehensive information is always on the University of Cyprus website: <https://www.ucy.ac.cy/> Specifically, navigate to the "Postgraduate Studies" section. Program Requirements: Each program has specific entry requirements (e.g., Bachelor's degree, GPA, language proficiency). Check the program page for detailed information. Language of Instruction: Most postgraduate programs are taught in English. Fees: Postgraduate fees vary depending on the program. Research Opportunities: The University of Cyprus is increasingly focused on research. Explore opportunities for theses and research projects. To help me give you even more targeted information, could you tell me: What field of study are you interested in? (e.g., Computer Science, Archaeology, Public Health, etc.)

Σχήμα 5.11: Απάντηση με 20% Θόρυβο στα Συμφραζόμενα

0.5 ratio:

Okay, let's break down the postgraduate programs offered by the University of Cyprus. They have a solid range of options, particularly in the sciences and humanities. Here's a categorized overview, as of today (October 26, 2023). Please note that program details, admission requirements, and fees can change, so ALWAYS check the official University of Cyprus website for the most up-to-date information: <https://www.ucy.ac.cy/en/>

1. Faculty of Medicine & Health Sciences: MSc in Medical Sciences: This is a popular and well-regarded program covering various aspects of medicine, including physiology, pharmacology, and clinical research. MSc in Public Health: Focuses on the promotion of health and well-being through public health interventions. MSc in Physiotherapy: A comprehensive program covering musculoskeletal, neurological, and pediatric physiotherapy. MSc in Biomedical Sciences: Covers molecular biology, genetics, and related areas. MSc in Occupational Therapy: Focuses on helping people with disabilities and illnesses improve their daily lives. MSc in Clinical Psychology: Investigates the theories and practices of clinical psychology.
2. Faculty of Science: MSc in Computer Science: Covers areas like artificial intelligence, data science, cybersecurity, and software engineering. MSc in Mathematics: Advanced mathematical topics, often with specialization options. MSc in Environmental Science: A growing area of focus, covering topics such as climate change, biodiversity, and environmental management. MSc in Geology & Geophysics: Research-oriented, focusing on geological processes and Earth systems. MSc in Statistics: Focused on statistical theory and methods. MSc in Chemistry: Advanced topics in organic, inorganic, and physical chemistry. MSc in Marine Science: Studying the marine environment, ecosystems, and resources. MSc in Astrophysics: Studying the physical universe and celestial objects.
3. Faculty of Humanities & Social Sciences: MSc in Accounting & Finance: Offers specializations in areas such as corporate finance, investment, and taxation. MSc in Business Administration: Focuses on various business disciplines like marketing, human resource management, and strategy. MSc in Law: Various specializations, including International Law, Commercial Law, and Human Rights Law. MSc in Political Science & International Relations: Provides a strong grounding in political theory and contemporary international affairs. MSc in History: Allows for specialization in various historical periods and regions. MSc in Archaeology: Combines archaeological research with theoretical and methodological skills. MSc in Translation & Interpreting: For aspiring translators and interpreters. MSc in Languages (English, Greek, French, German): Intensive language studies for those seeking professional qualifications. MSc in Tourism Management: Focuses on the management of tourism operations and destinations. MSc in Education: With specializations in areas like primary education, special education, and educational technology.
4. Other Programs: MSc in Renewable Energy: Examining the technologies and policies related to renewable energy sources. MSc in Urban Planning & Development: Focusing on sustainable urban development and planning.

Important Notes & Resources: Research Focus: The University of Cyprus has a growing research community. Many postgraduate programs are research-based. Fees: Fees vary depending on the program and your nationality. Check the university website for the most current fee information. Admission Requirements: Generally, you'll need a Bachelor's degree (or equivalent) in a relevant field. Some programs have specific entry requirements (e.g., minimum GPA, specific coursework). Application Deadlines: Deadlines vary by program and academic year. Be sure to check them carefully. Scholarships: The University of Cyprus offers scholarships to both domestic and international students. Explore the scholarship opportunities on their website. To help me give you even more targeted information, could you tell me: What field of study are you most interested in? (e.g., Computer Science, Business, Medicine, etc.)

Σχήμα 5.12: Απάντηση με 50% Θόρυβο στα Συμφραζόμενα

0.3 ratio:

Okay, let's break down the postgraduate programs offered at the University of Cyprus. They have a good range of Masters and Doctoral programs across various disciplines. Here's a categorized overview as of today, October 26, 2023. Please note that program details, including entry requirements and fees, are subject to change. Always check the official University of Cyprus website for the most up-to-date information.

1. Faculty of Medicine & Life Sciences MSc in Biomedical Sciences: (Focus on research and clinical aspects of medicine) PhD in Biomedical Sciences: Research-focused doctoral program. MSc in Public Health: (With options for specialization - Epidemiology, Health Promotion) MSc in Medical Genetics: (Focus on clinical and research aspects of genetics) PhD in Medical Sciences (various specializations - Pathology, Immunology, Pharmacology etc.)
2. Faculty of Education MSc in Educational Studies: (Focuses on contemporary educational challenges and research) MSc in Special Educational Needs: (Focuses on the needs of children and adults with disabilities) PhD in Education: Research-based doctoral program.
3. Faculty of Arts & Social Sciences MSc in Archaeology: (Various specializations, including Ancient Greek, Mediterranean Archaeology) MSc in Cultural Management: (Focus on the management and promotion of cultural heritage) MSc in International Relations & Security Studies MSc in Demography: (Population studies, analysis, and policy) MSc in Economic and Social Policy: MSc in Psychology: (Clinical, Developmental, Social, Organizational) MSc in Languages and Intercultural Communication PhD in Social Sciences: (Various specializations - Sociology, Anthropology, Political Science, History, etc.)
4. Faculty of Science & Technology MSc in Computer Science: (Software Engineering, Data Science, Cybersecurity, Artificial Intelligence) MSc in Engineering Sciences: (Various specializations - Civil, Mechanical, Electrical, etc.) MSc in Environmental Science & Technology MSc in Mathematics & Statistics MSc in Finance MSc in Business Administration: (Various specializations - Marketing, Human Resources, Logistics, etc.) MSc in Sustainable Energy Systems PhD in Science & Technology: (Various specializations - Physics, Chemistry, Biology, Computer Science, etc.)
5. Faculty of Humanities MSc in Classics: (Ancient Greek and Roman studies) MSc in Art History MSc in Linguistics PhD in Humanities: (Various specializations - History, Archaeology, Literature, etc.)

Important Resources & Links: University of Cyprus Postgraduate Programs Website: <https://www.ucy.ac.cy/postgraduate> (This is the best place to find detailed information) University of Cyprus Official Website: <https://www.ucy.ac.cy/>

Key Considerations for Prospective Students: Entry Requirements: These vary significantly depending on the program and your prior qualifications (e.g., undergraduate degree, relevant experience). Language of Instruction: Most postgraduate programs are taught in English. Research Focus: Doctoral programs are heavily research-oriented. Funding Opportunities: Investigate scholarships and grants offered by the University of Cyprus and external organizations. To help me give you even more targeted information, could you tell me: What field of study are you interested in? (e.g., Computer Science, Archaeology, Business, etc.)

Σχήμα 5.13: Απάντηση με 70% Θόρυβο στα Συμφραζόμενα

Λόγος Θορύβου	Περιγραφή	Ακρίβεια	Συμπέρασμα
1.0	Η πιο πλήρης και καλά κατηγοριοποιημένη απάντηση, περιλαμβάνει σωστά προγράμματα όπως Βιοϊατρικές Επιστήμες, Διοίκηση, ΤΝ, Νομική κ.ά.	Ακριβής	Πλησιέστερη στην πραγματικότητα
0.8	Ελαφρώς λιγότερο αναλυτική, αλλά αναφέρει σωστά και πρόσφατα προγράμματα όπως Υγεία, Γνωστικά Συστήματα κ.λπ.	Ακριβής	Πολύ καλή
0.5	Μέτριος βαθμός λεπτομέρειας, λείπουν αρκετά σύγχρονα ή διατμηματικά προγράμματα	Ελλιπής	Μέτρια
0.3	Πολύ γενική απάντηση, παραλείπει σχεδόν όλα τα διατμηματικά και εξειδικευμένα προγράμματα	Μη επαρκής	Ανεπαρκής

Πίνακας 5.3: Επιρροή Θορύβου στην Ακρίβεια Απαντήσεων

5.2 Συγκρίσεις με Παρόμοια Συστήματα

Ακολούθως, το σύστημα θα συγκριθεί με παρόμοια συστήματα όπως το NotebookLM [11] και το RunLLM [48] (βλ. Πίνακα 5.4). Αρχικά, για την αποθήκευση των εγγράφων το NotebookLM είναι αρκετά πιο γρήγορο από το PrivateLLM, ενώ το RunLLM είναι αρκετά πιο αργό επειδή λειτουργεί ως βελτιστοποίηση (fine tune) του γλωσσικού μοντέλου.

Τα δύο συστήματα δεν απαιτούν οποιαδήποτε εισαγωγή κλειδιού για την επικοινωνία με τα γλωσσικά μοντέλα. Επίσης, τα γλωσσικά μοντέλα τα οποία χρησιμοποιούν αποτελούνται από πολλές παραμέτρους και άρα οι απαντήσεις τους είναι πιο σωστές και υποστηρίζουν σωστά την χρήση ξένων γλωσσών.

Ακόμη, τα δύο συστήματα δέχονται ως είσοδο αρχεία πολλών μορφών, όπως κείμενο, εικόνα, και ήχο. Επιπρόσθετα, παρατηρήθηκε ότι το NotebookLM παρουσιάζει με μεγάλη ακρίβεια τα έγγραφα τα οποία χρησιμοποίησε για την εξαγωγή της απάντησης (βλ. Σχήμα 2.7). Αντιθέτως, το RunLLM δεν εμφανίζει καθόλου τις πηγές (βλ. Σχήμα 2.8).

Όσον αφορά τους περιορισμούς στη χρήση, το NotebookLM έχει περιορισμό δημιουργίας 100 συνεδριών, με μέγιστο αριθμό 50 εγγράφων και 50 ερωτήσεων ανά συνεδρία [11]. Το RunLLM δεν αναφέρει συγκεκριμένα τον περιορισμό στα δεδομένα που μπορεί να ανεβάσει ο χρήστης.

Ωστόσο, κανένα από τα δύο συστήματα δεν προσφέρει άμεσα τεχνικές προστασίας της ιδιωτικότητας των χρηστών. Αυτό αποτελεί ένα πλεονέκτημα στο PrivateLLM αφού το σύστημα προσφέρει μια αίσθηση διαφάνειας προς τον χρήστη σε σχέση με τα δεδομένα από τα έγγραφα του.

Χαρακτηριστικό	PrivateLLM	NotebookLM	RunLLM
Ταχύτητα αποθήκευσης εγγράφων	Μέτρια	Πολύ γρήγορη	Αργή (λόγω fine-tuning)
Απαιτείται API key	Απαιτείται από τον χρήστη (για Hugging Face)	Δεν απαιτείται	Δεν απαιτείται
Ποιότητα απαντήσεων	Καλή (μικρότερο μοντέλο)	Πολύ καλή (μεγάλο μοντέλο)	Πολύ καλή (fine-tuned μοντέλο)
Υποστήριξη Ελληνικής γλώσσας	Μερική (ανάλογα το μοντέλο)	Πλήρης	Πλήρης
Τύποι εισόδου	Μόνο κείμενο PDF	Κείμενο, εικόνα, ήχος	Κείμενο, εικόνα, ήχος
Παρουσίαση πηγών στην απάντηση	Εμφάνιση στο πλάι	Εμφανίζει ξεκάθαρα	Δεν εμφανίζει
Όρια χρήσης	Εξαρτάται από τον browser/συσκευή	100 συνεδρίες / 50 έγγραφα / 50 ερωτήσεις [11]	Δεν αναφέρονται
Μηχανισμοί προστασίας ιδιωτικότητας	Μέσω τοπικής αποθήκευσης, K-Ανωνυμίας	Δεν αναφέρονται	Δεν αναφέρονται

Πίνακας 5.4: Σύγκριση PrivateLLM με Άλλα Συστήματα

5.3 Περιορισμοί

Στην παρούσα φάση το PrivateLLM παρουσιάζει πολλούς περιορισμούς. Αρχικά, το σύστημα καλεί συγκεκριμένα μοντέλα, τα οποία δεν αποτελούνται από πολλές παραμέτρους. Αυτό κάνει τις απαντήσεις του συστήματος ελλείψεις ή όχι αρκετά ικανοποιητικές.

Επίσης, αν ο χρήστης παραχωρήσει Hugging Face Token το οποίο είναι μέρος του δωρεάν πλάνου, η χρήση του είναι περιορισμένη. Αυτό καθώς θα έχει περιορισμένο αριθμό από ερωτήσεις και απαντήσεις οι οποίες μπορούν να ανταλλαχθούν με το

Hugging Face Inference ειδικά στην κ-ανωνυμία όπου χρησιμοποιείται μεγάλος αριθμός από ψεύτικες ερωτήσεις. Συγκεκριμένα, με ένα δωρεάν token ο χρήστης έχει μόνο 0,10 δολάρια στο Hugging Face Inference και μπορεί να χρησιμοποιήσει περιορισμένο αριθμό από μοντέλα.

Ακόμη, η χρήση μιας τοπικής βάσης ως προς τον φυλλομετρητή ιστού έχει ως αποτέλεσμα την καθυστέρηση στην ανταπόκριση, ειδικά όταν υπάρχουν πολλά δεδομένα αποθηκευμένα στην βάση.

Επιπρόσθετα, το σύστημα ανταποκρίνεται μόνο σε δεδομένα που έχουν την μορφή PDF. Τέλος, το σύστημα χρειάζεται διαδίκτυο για να λειτουργήσει αφού η επικοινωνία μεταξύ του PrivateLLM και των γλωσσικών μοντέλων πραγματοποιείται με POST requests.

5.4 Μελλοντικές Εργασίες

Η υλοποίηση του συστήματος αποτελεί ένα πρώτο βήμα ως προς την ασφαλή επικοινωνία με γλωσσικά μοντέλα. Υπάρχουν ακόμη αρκετές βελτιώσεις οι οποίες μπορούν να προστεθούν στο σύστημα. Μια κύρια αναβάθμιση που χρειάζεται το σύστημα είναι η ικανότητα να διαβάζει διάφορων μορφών δεδομένα και όχι μόνο PDF αρχεία.

Επιπρόσθετα, εξίσου σημαντική είναι η προσθήκη νέων μοντέλων τα οποία έχουν περισσότερες παραμέτρους έτσι ώστε να βελτιωθεί η παραγωγή φυσικής γλώσσας. Ακόμη, μπορεί να υλοποιηθεί ένας τρόπος μετάφρασης της ερώτησης και των συμφραζόμενων με τρόπο όπου δεν θα απαιτείται κλειδί για την χρήση του.

Πέρα από την συμπλήρωση περισσότερων μοντέλων μια καλή βελτίωση θα ήταν η προσθήκη της δυνατότητας επικοινωνίας με τα GPTs της OpenAI. Επίσης, ακόμη μια επιλογή θα μπορούσε να είναι η κλήση του RunLLM μοντέλου.

Τέλος, εκτός από την κ-ανωνυμία και την προσθήκη θορύβου, μπορούν να υλοποιηθούν διαφορετικές τεχνικές για την προστασία των δεδομένων του χρήστη.

Γενικά το σύστημα χρειάζεται αρκετή βελτιστοποίηση έτσι ώστε να είναι πιο φιλικό προς τον χρήστη. Συμπερασματικά, υπάρχουν πολλές ακόμη ευκαιρίες για την ανάπτυξη του PrivateLLM.

Κεφάλαιο 6

Συμπεράσματα

<i>6.1 Τελικά Συμπεράσματα.....</i>	<i>112</i>
-------------------------------------	------------

6.1 Τελικά Συμπεράσματα

Με την ραγδαία ανάπτυξη της τάσης των ανθρώπων προς την χρήση της Τεχνητής Νοημοσύνης αλλά και την αυξημένη συνείδηση των χρηστών ως προς τα προσωπικά δεδομένα και την ιδιωτικότητα τους στο διαδίκτυο είναι βέβαιο ότι η ανάγκη για συστήματα όπως το PrivateLLM θα αυξάνεται παράλληλα.

Μέσω της μελέτης και της ανάπτυξης το συστήματος PrivateLLM είναι φανερό ότι η προστασία των δεδομένων ενός χρήστη κατά την διάρκεια της αλληλεπίδρασης του με μεγάλα γλωσσικά μοντέλα βρίσκεται σε καλό δρόμο. Αδιαμφισβήτητα, υπάρχει ακόμη μεγάλος δρόμος για την απόλυτη προστασία των δεδομένων από την Παραγωγική Τεχνητή Νοημοσύνη και τα Μεγάλα Γλωσσικά Μοντέλα.

Ακόμη, η διπλωματική εργασία ανάδειξε την δύναμη της μεθόδου Retrieval Augmented Generation, η οποία μπορεί να παραγάγει ακριβείς απαντήσεις σε ερωτήσεις του χρήστη που αφορούν νέα προς το μοντέλο δεδομένα, χωρίς την χρήση των μεγάλων υπολογιστικών πόρων που απαιτεί η μέθοδος fine tuning.

Η ανάπτυξη τέτοιων συστημάτων είναι απαραίτητη, ειδικά με την γρήγορη πορεία προς την αυτόματη εκπαίδευση των γλωσσικών μοντέλων. Συνοπτικά, το PrivateLLM είναι ένα πρώτο βήμα προς την ασφαλή χρήση της Τεχνητής Νοημοσύνης με βάση προσωπικά δεδομένα, με απλό και προσβάσιμο τρόπο για τον χρήστη.

Βιβλιογραφία

- [1] A. Al-Shaikh and A. Sleit, “Evaluating IndexedDB performance on web browsers,” pp. 1–4, 2017, [Online]. Available: https://www.researchgate.net/publication/320614525_Evaluating_IndexedDB_Performance_on_Web_Browsers
- [2] A. Bick, A. Blandin, and D. J. Deming, “The rapid adoption of generative AI,” NBER Working Paper No. 32966, National Bureau of Economic Research, Cambridge, MA, pp. 10-11,13, 2024. [Online]. Available: <http://www.nber.org/papers/w32966>
- [3] Bootstrap, Bootstrap v5.3 Documentation. [Online]. Available: <https://getbootstrap.com/docs/5.3/>
- [4] Bootstrap Icons, Bootstrap Icons. [Online]. Available: <https://icons.getbootstrap.com/>
- [5] C.-H. Lin and P.-J. Cheng, “Legal documents drafting with fine-tuned pre-trained large language model,” Comput. Sci. Inf. Technol. (CS & IT) - CSCP 2024, pp. 4-8, 2024. [Online]. Available: <https://doi.org/10.5121/csit.2024.140819>.
- [6] C. Smith, “The History of Artificial Intelligence”, CSEP 590A History of Computing, University of Washington, pp. 4, 2006. [Online]. Available: <https://courses.cs.washington.edu/courses/csep590/06au/projects/history-ai.pdf>
- [7] Chroma. Chroma: An open-source embedding database, [Online]. Available: <https://www.trychroma.com/>
- [8] CyLaw, “CyLaw – Cyprus Legal Information,” CyLaw, [Online]. Available: <https://www.cylaw.org/>
- [9] D. Cer et al., "Universal Sentence Encoder," arXiv preprint arXiv:1803.11175, 2018. [Online]. Available: <https://arxiv.org/abs/1803.11175>
- [10] D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), IETF, RFC 8259, Dec. 2017. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc8259>
- [11] Google Cloud, “NotebookLM Enterprise overview,” [Online]. Available: <https://cloud.google.com/agentspace/notebooklm-enterprise/docs/overview>

- [12] Google, SentencePiece Tokenizer Library, [Online]. Available: <https://github.com/google/sentencepiece>
- [13] H. Naveed et al., "A Comprehensive Overview of Large Language Models," arXiv preprint arXiv:2307.06435, pp. 2-8, 2024. [Online]. Available: <https://arxiv.org/pdf/2307.06435>
- [14] H. Soudani, E. Kanoulas, and F. Hasibi, "Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge," *arXiv preprint* arXiv:2403.01432, pp. 1-2, 2024. [Online]. Available: <https://arxiv.org/abs/2403.01432>
- [15] Harvey AI, "Harvey AI – AI Solutions for Legal Professionals," Harvey AI, [Online]. Available: <https://www.harvey.ai/>
- [16] Hugging Face, Access Tokens – Hugging Face Hub Documentation, [Online]. Available: <https://huggingface.co/docs/hub/security-tokens>
- [17] Hugging Face, Hugging Face Transformers and Inference API, [Online]. Available: <https://huggingface.co/docs/api-inference/>
- [18] Hugging Face, safetensors – Fast and Safe Tensor Storage Format, [Online]. Available: <https://github.com/huggingface/safetensors>
- [19] J. Manchanda, L. Boettcher, M. Westphalen, and J. Jasser, "The Open Source Advantage in Large Language Models (LLMs)," *arXiv preprint* arXiv:2412.12004, pp. 1, 2025. [Online]. Available: <https://arxiv.org/abs/2412.12004>
- [20] L. Papaleo, "Introduction to XML and Its Applications," in *Handbook of Research on Text and Web Mining Technologies*, A. Kao and S. R. Poteet, Eds. Singapore: World Scientific Publishing Co., pp. 3-1, 2014, [Online]. Available: https://www.researchgate.net/publication/304823432_Introduction_to_XML_and_its_applications
- [21] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002. [Online]. Available: <https://www.worldscientific.com/doi/abs/10.1142/S021848850200165X>
- [22] LangChain. LangChain Community: PyPDFDirectoryLoader Module, [Online]. Available: <https://github.com/langchain-ai/langchain>

- [23] LangChain, LangChain Python Framework. [Online]. Available: <https://docs.langchain.com/>
- [24] LangChain. LangChain Text Splitters, [Online]. Available: <https://docs.langchain.com/docs/components/text-splitters/>
- [25] M. Ali, A. Arunasalam, and H. Farrukh, “Understanding users’ security and privacy concerns and attitudes towards conversational AI platforms,” pp. 5-6, 2025. [Online]. Available: <https://arxiv.org/abs/2504.06552>
- [26] Microsoft, Translator Text API – Azure Cognitive Services, [Online]. Available: <https://learn.microsoft.com/en-us/azure/cognitive-services/translator/>
- [27] Mistral AI, Mistral-7B-Instruct-v0.3, Hugging Face. [Online]. Available: <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>
- [28] Mozilla Developer Network, Document Object Model (DOM) – Web APIs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
- [29] Mozilla Developer Network, IndexedDB API Documentation, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API
- [30] Mozilla Developer Network (MDN), Promise – JavaScript Reference, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise
- [31] Mozilla Developer Network (MDN), Set – JavaScript Reference, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set
- [32] Mozilla Developer Network, TextDecoder – Encoding API, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/TextDecoder>
- [33] Mozilla Developer Network, Web Storage API – localStorage and sessionStorage, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
- [34] Mozilla Developer Network, WindowOrWorkerGlobalScope.setTimeout(), [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/setTimeout>

- [35] Mozilla, PDF.js: Portable Document Format Viewer, [Online]. Available: <https://mozilla.github.io/pdf.js/>
- [36] N. Razavian, F. Knoll, and K. J. Geras, "Artificial intelligence explained for nonexperts," *Semin. Musculoskelet. Radiol.*, vol. 24, pp. 1-2, 2020. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7380586/>
- [37] N. Satterfield, P. Holbrook, and T. Wilcox, "Fine-tuning Llama with Case Law Data to Improve Legal Domain Performance," OSF Preprints, pp. 2-5, 2024. [Online]. Available: https://osf.io/preprints/osf/e6mjs_v1
- [38] Nomic AI, nomic-embed-text model. [Online]. Available: <https://ollama.com/library/nomic-embed-text>
- [39] Ollama, Ollama Documentation. [Online]. Available: <https://docs.ollama.com/>
- [40] Ollama, Ollama: Run LLMs Locally. [Online]. Available: <https://ollama.com/>
- [41] OpenStreetMap contributors, "Map data for University of Cyprus area," [Online]. Available: <https://www.openstreetmap.org>
- [42] P. Colombo *et al.*, "SaulLM-7B: A pioneering Large Language Model for Law," *arXiv preprint arXiv:2403.03883*, pp. 1-5, 2024. [Online]. Available: <https://arxiv.org/abs/2403.03883>
- [43] pixegami, "An Improved Langchain RAG Tutorial (v2) with local LLMs, database updates, and testing," GitHub repository, <https://github.com/pixegami/rag-tutorial-v2>.
- [44] Python Software Foundation, Python Language Reference, version 3.x. [Online]. Available: <https://www.python.org>
- [45] PyTorch, PyTorch Documentation, [Online]. Available: <https://pytorch.org>
- [46] R. Fielding and J. Reschke, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, IETF, RFC 7231, June 2014. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7231>
- [47] Republic of Cyprus, "The University of Cyprus Laws 1989 to 2024: English Translation and Consolidation", Office of the Law Commissioner, Nicosia, Cyprus, Apr. 2024. [Online]. Available: <https://www.ucy.ac.cy/legislation/wp-content/uploads/The-University-of-Cyprus-Laws-1989-to-2024.pdf>.

- [48] RunLLM, “RunLLM Documentation,” [Online]. Available: <https://docs.runllm.com/>
- [49] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," *arXiv preprint* arXiv:2410.12837, pp. 2-3, 2024. [Online]. Available: <https://arxiv.org/abs/2410.12837>
- [50] S. Yue *et al.*, "DISC-LawLLM: Fine-tuning Large Language Models for Intelligent Legal Services," *arXiv preprint* arXiv:2309.11325, pp. 1-4, 2023. [Online]. Available: <https://arxiv.org/abs/2309.11325>
- [51] T. Watson, gpt-3-encoder JavaScript Library, [Online]. Available: <https://github.com/latitudegames/GPT-3-Encoder>
- [52] TensorFlow.js Team, TensorFlow.js: A WebGL Accelerated Machine Learning Library for the Web, [Online]. Available: <https://www.tensorflow.org/js>
- [53] University of Cyprus, ChatUCY – Interactive LLM Chat Service, [Online]. Available: <https://chatucy.cs.ucy.ac.cy/>
- [54] University of Cyprus, Data Management Systems Laboratory (DMSL), [Online]. Available: <https://dmsl.cs.ucy.ac.cy>
- [55] University of Cyprus, Department of Computer Science, “Study Guide 2024–2025”, Nicosia, Cyprus, 2024. [Online]. Available: <https://www.cs.ucy.ac.cy/docs/prospectus.pdf>.
- [56] University of Cyprus, “Framework for the Composition of the Examining Committees,” [Online]. Available: <https://www.ucy.ac.cy/graduateschool/wp-content/uploads/sites/45/2023/10/POLITIKI-3MELIS-5MELIS-ENGLISH-FINAL-October-2023.pdf>
- [57] University of Cyprus, “Postgraduate Programmes Winter Semester 2025–2026 (English version),” Feb. 2025. [Online]. Available: <https://www.ucy.ac.cy/graduateschool/wp-content/uploads/sites/45/2025/02/Agglikli-Prokirixi-xeimerino-2025-2026.pdf>
- [58] University of Cyprus, “Postgraduate Programmes Winter Semester 2025–2026 (Greek version),” Feb. 2025. [Online]. Available:

- <https://www.ucy.ac.cy/graduateschool/wp-content/uploads/sites/45/2025/02/Elliniki-Prokirixi-xeimerino-2025-2026.pdf>
- [59] University of Cyprus, “Postgraduate Prospectus 2023–2025”, Nicosia, Cyprus, Dec. 2023. [Online]. Available: https://www.ucy.ac.cy/publications/wp-content/uploads/sites/54/2023/12/Postgr.Prosp_.En_23-25.pdf.
 - [60] University of Cyprus, “Postgraduate Studies Rules, English” Mar. 2021 [Online]. Available: <https://www.ucy.ac.cy/graduateschool/wp-content/uploads/sites/45/2023/02/kanones-dimosiefsimi-morfi-14.02.23-english-1.pdf>
 - [61] University of Cyprus, “Postgraduate Studies Rules, Greek” Mar. 2021 [Online]. Available: <https://www.ucy.ac.cy/graduateschool/wp-content/uploads/sites/45/2022/05/Kanones-dimosiefsimi-morfi-22.04.22-greek-2.pdf>
 - [62] University of Cyprus, “Undergraduate Prospectus 2024–2026”, Nicosia, Cyprus, Jul. 2024. [Online]. Available: <https://www.ucy.ac.cy/publications/wp-content/uploads/sites/54/2024/07/UndergraduateProspectusEN24-26.pdf>
 - [63] Wikimedia Foundation, Wikipedia REST API – Summary Endpoint, [Online]. Available: https://en.wikipedia.org/api/rest_v1/
 - [64] X. Ye, Y. Yan, J. Li, and B. Jiang, “Privacy and personal data risk governance for generative artificial intelligence: A Chinese perspective,” vol. 48, no. 8, pp. 4, 2024. [Online]. Available: <https://doi.org/10.1016/j.telpol.2024.102851>
 - [65] Y. Han, C. Liu, and P. Wang, “A comprehensive survey on vector database: Storage and retrieval technique, challenge,” arXiv:2310.11703, pp. 1, 2023. [Online]. Available: <https://arxiv.org/abs/2310.11703>
 - [66] Y. Liu *et al.*, “Understanding LLMs: A Comprehensive Overview from Training to Inference,” *arXiv preprint* arXiv:2401.02038, pp. 6-18, 2024. [Online]. Available: <https://arxiv.org/abs/2401.02038>
 - [67] Z. Liu, Y. Zhu, and M. Lu, “Enhancing legal expertise in large language models through composite model integration: The development and evaluation of Law-Neo,” in Proc. Natural Legal Language Processing Workshop 2024, pp. 1-4, Nov. 2024. [Online]. Available: <https://aclanthology.org/2024.nllp-1.3.pdf>

- [68] Z. Proser, “How to create a custom Alpaca instruction dataset for fine-tuning LLMs,” Modern Coding MC, 2024. [Online]. Available: <https://zackproser.com/blog/how-to-create-a-custom-alpaca-dataset>
- [69] Z. Xia, L. Zhu, B. Li, F. Chen, Q. Li, C. Liao, F. Wang, and H. Liu, “Analyzing 16,193 LLM papers for fun and profits,” pp. 2, 2025. [Online]. Available: <https://arxiv.org/abs/2504.08619>