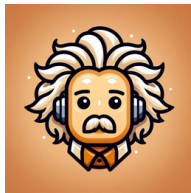


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Ευφυούς Ψηφιακού Βοηθού Βασισμένου σε
Μεγάλα Γλωσσικά Μοντέλα για την Υποστήριξη
Φοιτητών»



Του φοιτητή
Δημητρίου Κουτσέλη
Αρ. Μητρώου: 2019082

Επιβλέπων
Δρ Κωνσταντίνος Γουλιάνας
Καθηγητής

Ημερομηνία 09/05/2025

Τίτλος Δ.Ε. Ανάπτυξη Ευφυούς Ψηφιακού Βοηθού Βασισμένου σε Μεγάλα Γλωσσικά Μοντέλα για
την Υποστήριξη Φοιτητών

Κωδικός Δ.Ε. 25178

Ονοματεπώνυμο φοιτητή Δημήτριος Κουτσέλη
Ονοματεπώνυμο εισηγητή Δρ Κωνσταντίνος Γουλιάνας

Ημερομηνία ανάληψης Δ.Ε. 12-03-2025

Ημερομηνία περάτωσης Δ.Ε. 09-05-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Δημήτριου Κουτσέλη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Στους γονείς μου

Πρόλογος

Για την διπλωματική μου επέλεξα το θέμα, «Ανάπτυξη Ευφυούς Ψηφιακού Βοηθού Βασισμένου σε Μεγάλα Γλωσσικά Μοντέλα για την Υποστήριξη Φοιτητών» διότι πιστεύω ότι ο φυσικός λόγος αποτελεί την καλύτερη διεπαφή χρήστη, καθώς και ότι τα Μεγάλα Γλωσσικά Μοντέλα ελεύθερης διανομής μπορούν πλέον να φέρουν εις πέρας αυτή την πρόκληση με μεγάλη επιτυχία. Δημιούργησα έναν έξυπνο βοηθό ο οποίος αξιοποιώντας την αρχιτεκτονική RAG (Retrieval-Augmented Generation) μπορεί να απαντήσει σε κάθε ερώτηση της οποίας η απάντηση περιέχεται στο [επίσημο site της σχολής](#), καθώς και να απαντήσει σε ερωτήσεις καθημερινότητας – όπως η πρόβλεψη του καιρού ή/και οι τελευταίες ειδήσεις αξιοποιώντας το function calling που διαθέτουν τα σύγχρονα μεγάλα γλωσσικά μοντέλα και κλήσεις σε ελεύθερα APIs (Application Programming Interface).

Σκοπός είναι η δημιουργία μιας καινούργιας διεπαφής χρήστη για τους φοιτητές όπου θα μπορούν με μία ερώτηση να πάρουν, με πολύ μεγάλη ακρίβεια, την απάντηση που επιθυμούν.

Περίληψη

Η «Ανάπτυξη Ευφυούς Ψηφιακού Βοηθού Βασισμένου σε Μεγάλα Γλωσσικά Μοντέλα για την Υποστήριξη Φοιτητών» αφορά έναν βοηθό που αναπτύχθηκε ως discord bot και αξιοποιεί τα μεγάλα γλωσσικά μοντέλα για την υποστήριξη των φοιτητών μέσω ενός Flask (python) backend. Ο βοηθός χρησιμοποιεί το [Mistral-7B-Instruct-v0.3](#) μοντέλο για την κατανόηση των ερωτημάτων των φοιτητών και την δημιουργία απαντήσεων, σε συνδυασμό με την αρχιτεκτονική RAG (Retrieval-Augmented Generation) κάνοντας χρήση της διανυσματικής βάσης δεδομένων ChromaDB, η οποία περιέχει δεδομένα που πάρθηκαν με web scrapping από την επίσημη σελίδα της σχολής με την βιβλιοθήκη BeautifulSoup. Επίσης, ο βοηθός διαθέτει την δυνατότητα να καλεί συναρτήσεις (function calling capabilities of LLMs) όταν εντοπίζει ερώτηση χρήστη της οποίας απάντηση μπορεί να βρει σε κάποιο από τα ελεύθερα APIs.

Για την ανάπτυξη του βοηθού εξετάστηκαν 4 μοντέλα, 2 multilingual και 2 fine tuned εκδοχές των δύο. Μετά τα πειράματα που έγιναν καλύτερα αποτελέσματα βρέθηκαν στην χρήση του Mistral 7B Instruct μοντέλου με την ερώτηση να γίνεται στην αγγλική γλώσσα. Για τον λόγο αυτό, μόλις έρχεται μία ερώτηση, γίνεται ανίχνευση της γλώσσας με την βιβλιοθήκη langdetect και μετάφραση της στα αγγλικά (αν δεν είναι ήδη) με το μοντέλο [Seamlessm4t](#) και μόλις παραχθεί η απάντηση, μεταφράζεται εκ νέου στην αρχική γλώσσα της ερώτησης. Κάποιο μεγαλύτερο μοντέλο με περισσότερη πολυπλοκότητα ίσως να μην απαιτούσε αυτή τη διαδικασία ωστόσο οι περιορισμοί στο hardware για το μέγεθος αυτών των μοντέλων κάνουν την συγκεκριμένη λύση την καλύτερη.

Το Flask Backend επιτρέπει την δυνατότητα κλιμάκωσης της εφαρμογής με την δημιουργία ui οποιασδήποτε μορφής.

Τέλος, η εφαρμογή προσφέρει σημαντικά οφέλη στους φοιτητές, διευκολύνοντας την άμεση πρόσβαση τους σε πληροφορίες κάνοντας μία ερώτηση σε μία υπηρεσία που ήδη χρησιμοποιούν που διαφορετικά θα απαιτούσε αναζήτηση σε διαφορετικές πηγές.

Dimitrios Koutselis

Abstract

“Development of an Intelligent Digital Assistant Based on Large Language Models for Student Support” is about an assistant developed as a discord bot that leverages large language models to support students through a Flask (python) backend. The assistant uses the Mistral-7B-Instruct-v0.3 model to understand student queries and generate answers, combined with the RAG (Retrieval-Augmented Generation) architecture making use of the vector database ChromaDB, which contains data obtained by web scraping from the official faculty page with the BeautifulSoup library. Also, the assistant has the function calling capabilities of LLMs when it detects a user question whose answer can be found in one of the free APIs.

For the development of the helper 4 models were considered, 2 multilingual and 2 fine tuned versions of both. After the experiments were done better results were found in using the Mistral 7B Instruct model with the question being asked in English. For this reason, once a question comes in, the language is detected with the langdetect library and translated into English (if not already) with the SeamlessM4t model and once the answer is produced, it is re-translated into the original language of the question. Some larger model with more complexity might not require this process however the hardware limitations on the size of these models make this the best solution.

The Flask Backend allows the ability to scale the application by creating ui's of any format.

Finally, the application offers significant benefits to students by facilitating their immediate access to information by asking a question to a service they already use that would otherwise require searching different sources.

Ευχαριστίες

Θέλω να ευχαριστήσω την οικογένεια μου, η οποία με στήριξε απόλυτα σε όλη την διάρκεια της φοίτησής μου στο Τμήμα. Επίσης θέλω να ευχαριστήσω τον Δρ. Κωνσταντίνο Χατζησάββα (MSENSIS A.E.) για την πολύτιμη βοήθειά του, χωρίς την οποία αυτή η διπλωματική εργασία δε θα μπορούσε να έχει γίνει, καθώς και τον Καθηγητή κ. Κωνσταντίνο Γουλιάνα για την ευκαιρία που μου έδωσε να αναλάβω ένα τόσο ενδιαφέρον θέμα.

Περιεχόμενα

Πρόλογος	v	
Περίληψη	vi	
Abstract	vii	
Ευχαριστίες	viii	
Περιεχόμενα	ix	
Κατάλογος Σχημάτων	xi	
Κατάλογος Πινάκων	xi	
Συνομογραφίες	xii	
Κεφάλαιο 1ο:	1	
1.1	1	
1.2	1	
1.3	22	
Κεφάλαιο 2ο:	3	
2.1	3	
2.2	3	
Κεφάλαιο 3ο:	56	
3.1	56	
3.2	Μηχανική Μάθηση56	
3.3	Αρχιτεκτονική Transformers78	
3.4	Αρχιτεκτονική RAG89	
3.5	Μεγάλα Γλωσσικά Μοντέλα (Large Language Models, LLMs) Error! Bookmark not defined. 10	
3.6	Αναγνώριση Γλώσσας και Μετάφραση1112	
3.7	Pytorch και Cuda	13
3.8	Διεπαφή Προγραμματισμού Εφαρμογών	13
3.9	Flask και Gunicorn	13
3.10	Huggingface και Transformers	14
3.11	Καθαρή αρχιτεκτονική	14
3.12	Discord	16
3.13	Embeddings	16
3.14	Web Scraping και BeautifulSoup	17
3.15	GitHub	17

Κεφάλαιο 4ο:		1819
4.1	519	
4.2	Διεπαφή Χρήστη - Discord	19
4.3	Flask Backend	20
4.4	Υπηρεσίες Μηχανικής Μάθησης	22
4.5	Διανυσματική Βάση Δεδομένων	38
4.6	Υπηρεσίες Διαθέσιμες για Function Calling	42
4.7	Σύνοψη	50
Κεφάλαιο 5ο:	Διερεύνηση του καλύτερου Μοντέλου και των Υπερπαραμέτρων	51
5.1	551	
5.2	Dataset	51
5.3	Μέθοδος Αξιολόγησης	52
5.4	Αναζήτηση Πλέγματος και Υπερ Παράμετροι	53
5.5	Αξιολόγηση των Μεγάλων Γλωσσικών Μοντέλων	55
5.6	Σύνοψη	59
Κεφάλαιο 6ο:	Συμπεράσματα και Βελτιώσεις	60
6.1	560	
6.2	Προτάσεις για βελτίωση και αδυναμίες	60
6.3	Σύνοψη	60
BIBΛΙΟΓΡΑΦΙΑ		62

Κατάλογος Σχημάτων

Σχήμα 1: Αρχιτεκτονική HyDE	4
Σχήμα 2: Αποτελέσματα της μεθόδου HyDE	4
Σχήμα 3: Τύποι Μηχανικής Μάθησης	6
Σχήμα 4: Η Ανάπτυξη των δημοσιεύσεων με Μηχανική Μάθηση στη Χημεία	7
Σχήμα 5: Οι έννοιες Τεχνητή Νοημοσύνη, Μηχανική Μάθηση και Βαθιά Μηχανική Μάθηση	8
Σχήμα 6: Η αρχιτεκτονική Transformers	9
Σχήμα 7: Η αρχιτεκτονική RAG	10
Σχήμα 8: Παράδειγμα λειτουργίας Μεγάλου Γλωσσικού μοντέλου	11
Σχήμα 9: Διαδικασία εκπαίδευσης ενός Μεγάλου Γλωσσικού Μοντέλου	12
Σχήμα 10: Επίπεδα του Clean Architecture	15
Σχήμα 11: Παράδειγμα Embeddings	16
Σχήμα 12: Αρχιτεκτονική Υλοποίησης διασύνδεσης διεπαφής χρήστη με το backend	20
Σχήμα 13: Διάγραμμα Ροής των Endpoints	21
Σχήμα 14: Επιδόσεις του Mistral 7B μοντέλου	22
Σχήμα 15: Αρχιτεκτονική Mistral 7B σε αντιπαράθεση με transformer	23
Σχήμα 16: Αρχιτεκτονική Llama 3.1 μοντέλου	26
Σχήμα 17: Παράδειγμα function calling	31
Σχήμα 18: Διάγραμμα ροής του function calling service	32
Σχήμα 19: Διάγραμμα ροής του RAG service	33
Σχήμα 20: Screenshot από την εφαρμογή κατά την χρήση RAG	33
Σχήμα 21: Διάγραμμα ροής του Machine Learning Service	35
Σχήμα 22: Αρχιτεκτονική seamless4t μοντέλου	37
Σχήμα 23: Επιδόσεις seamless4t μοντέλου	37
Σχήμα 24: Αρχιτεκτονική της ChromaDB	38
Σχήμα 25: Παράδειγμα του sliding window	39
Σχήμα 26: Web crawler vs Web Scraping	40
Σχήμα 27: Heatmap πειραμάτων για Υπερπαραμέτρους διανυσματικής βάσης	41
Σχήμα 28: Διάγραμμα ράβδων για Υπερπαραμέτρους διανυσματικής βάσης	42
Σχήμα 29: Αρχιτεκτονική function calling	43
Σχήμα 30: Απόσπασμα κώδικα υλοποίησης few shot examples	44
Σχήμα 31: Παράδειγμα συνάρτησης διαθέσιμης στο Μεγάλο Γλωσσικό Μοντέλο	44
Σχήμα 32: Screenshot από την εφαρμογή με κλήση συνάρτησης για τωρινό καιρό	45
Σχήμα 33: Screenshot από την εφαρμογή με κλήση συνάρτησης για πρόβλεψη καιρού	46
Σχήμα 34: Screenshot από την απάντηση του μοντέλου κατα την αναζήτηση ειδήσεων	48
Σχήμα 35: Screenshot από την αυτόματη διαχείριση των συνδέσμων από το discord	49
Σχήμα 36: Οπτικοποίηση της αναζήτησης πλέγματος	53
Σχήμα 37: Παράδειγμα των Υπερπαραμέτρων top_p και temperature	54
Σχήμα 38: Αποτελέσματα Grid Search σε δεδομένα σε Ελληνικά και Αγγλικά	55
Σχήμα 39: Αποτελέσματα Grid Search σε μεταφρασμένα δεδομένα	56
Σχήμα 40: Διάγραμμα Bleu Score προς Top_p, Temperature και τύπου dataset	57
Σχήμα 41: Heatmap Μέσου όρου τιμών του Mistral 7B μοντέλου	57
Σχήμα 42: Heatmap μέσου όρου τιμών LLama μοντέλου	58
Σχήμα 43: Heatmap μέσου όρου τιμών Meltimi μοντέλου	58
Σχήμα 44: Heatmap μέσου όρου τιμών Krikri μοντέλου	58

Κατάλογος Πινάκων

Πίνακας 1: Δεδομένα Επανεκπαίδευσης μοντέλου Meltemi	23
Πίνακας 2: Επιδόσεις μοντέλου Meltemi	24
Πίνακας 3: Επιδόσεις Llama 3.1 8B Instruct	26-27
Πίνακας 4: Δεδομένα επανεκπαίδευσης του μοντέλου Krikri	29
Πίνακας 5: Επιδόσεις μοντέλου Krikri	30
Πίνακας 6: Παραδείγματα από το dataset	51

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
RAG	Retrieval Augmented Generation
LLM	Large Language Model
API	Application Programming Inference

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η εξέλιξη του διαδικτύου και των ηλεκτρονικών υπολογιστών έφερε επανάσταση στον τρόπο που η ανθρωπότητα αντιλαμβάνεται, επεξεργάζεται, χρησιμοποιεί και προσπελάζει την πληροφορία. Δεν ήταν μόνο η διαθεσιμότητα της πληροφορίας και η διασύνδεση όλων των χρηστών του διαδικτύου με μεγάλες βάσεις δεδομένων αυτό που έφερε την επανάσταση αλλά το γεγονός ότι άλλαξε η διεπαφή που το δίνει πρόσβαση στην πληροφορία, από την αναζήτηση σε εγκυκλοπαίδειες πήγαμε στη αναζήτηση με μερικά κλικ. Το επόμενο επίπεδο στο οποίο εισέρχεται η ανθρωπότητα όσον αφορά τις διεπαφές στην πληροφορία είναι ο φυσικός λόγος.

Τα Μεγάλα Γλωσσικά Μοντέλα αποτελούν μία από τις σημαντικότερες δημιουργίες στην πληροφορική, καθώς έχουν φέρει επανάσταση και δραματικές αλλαγές στον τρόπο με τον οποίο μία μηχανή είναι ικανή να κατανοήσει, να επεξεργαστεί και να παράξει κείμενο. Η δημιουργία αυτών των έδωσε την δυνατότητα σύνδεσης μεγάλων βάσεων γνώσης με ένα μεγάλο γλωσσικό μοντέλο με σκοπό την μετάβαση στην εποχή όπου ο φυσικός λόγος αποτελεί τον τρόπο αναζήτησης γνώσης.

Τα μεγάλα γλωσσικά μοντέλα επίσης διαθέτουν δυνατότητες κλήσης συναρτήσεων (function calling). Η δυνατότητα των μοντέλων αυτών να χρησιμοποιούν εργαλεία μπορεί να τους διαμορφώσει σε πράκτορες (agents), ψηφιακών βοηθών δηλαδή οι οποίοι με βάση την κατανόηση του προβλήματος που έχει εκφράσει σε φυσικό λόγο ο χρήστης τους, μπορούν να επιλέξουν το κατάλληλο εργαλείο για να το λύσουν.

Η αρχιτεκτονική RAG (Retrieval Augmented Generation) [1] η οποία συνδέει ένα μεγάλο γλωσσικό μοντέλο με μία βάση γνώσης, έχει δώσει δείγματα βελτίωσης, ειδικότερα στα μικρότερα LLMs (Large Language Models), όταν το πρόβλημα αφορά κάποιο πολύ μικρό και συγκεκριμένο πεδίο. Η αρχιτεκτονική αυτή επιβάλλει την χρήση μίας διανυσματικής βάσης δεδομένων καθώς και έναν αυτοματοποιημένο τρόπο γεμίσματος της από την πηγή στην οποία επιθυμούμε να έχει πρόσβαση το μεγάλο γλωσσικό μοντέλο.

Ο λόγος είναι το σύγχρονο μέσω διασύνδεσης του χρήστη με την πληροφορία, χάρη στα μεγάλα γλωσσικά μοντέλα, ωστόσο εφαρμογές όπως το discord οι οποίες επιτρέπουν την ανάπτυξη και την χρήση εφαρμογών μέσω της δικής τους υποδομής καθώς και η δημοφιλία του στους νέους, το καθιστούν ιδανική πλατφόρμα μέσω της οποίας οι φοιτητές θα έχουν πρόσβαση στον ψηφιακό βοηθό.

1.2 Στόχος Διπλωματικής

Σε αυτήν την εργασία, ο κύριος στόχος μου ήταν να αναπτύξω μία scalable υποδομή μέσω της οποίας οι φοιτητές του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου Ελλάδος θα έχουν ευκολότερη πρόσβαση στην πληροφορία που τους ενδιαφέρει καθώς και λύσεις σε καθημερινά προβλήματα. Οι φοιτητές θα έχουν πρόσβαση σε πληροφορίες όπως (α) στοιχεία επικοινωνίας καθηγητών, (β) προγράμματα μαθημάτων και εξεταστικής, (γ) σημαντικές πληροφορίες για το τμήμα και (δ) διαθέσιμες υποδομές του τμήματος, απλώς ρωτώντας σε ένα chat που ήδη, κατα πλειοψηφία, χρησιμοποιούν, χωρίς πλέον να υπάρχει η ανάγκη να ψάχνουν την πληροφορία στο site της σχολής. Επίσης, θα έχουν πρόσβαση σε πληροφορίες που δίνονται από ελεύθερα APIs

(Application Programming Inference) όπως προβλέψεις καιρού, δημοφιλοί νέα και άλλα. Σκοπός της εφαρμογής είναι να κάνει την φοίτηση των φοιτητών της σχολής πιο εύκολη.

Ο τρόπος σχεδιασμού της εφαρμογής, την καθιστά scalable καθώς ακολουθεί τις πρακτικές του clean architecture και το rython backend κομμάτι έχει σχεδιαστεί με τέτοιο τρόπο, ώστε να μπορεί συμπληρωματικά να υποστηρίξει και οποιασδήποτε άλλης μορφής διεπαφής χρήστη π.χ. android εφαρμογή, website κ.ο.κ.

Τέλος, είναι σημαντικό να αναφέρω τις διάφορες προκλήσεις που αντιμετώπισα, τα μοντέλα, αν και αποτελούν μια κατηγορία «μικρών» μοντέλων αυτού του είδους, απαιτούν πολύ καλή υποδομή hardware καθώς και μεγάλους χρόνους απόκρισης που έκαναν το testing και το debugging αρκετά χρονοβόρο. Επίσης, η απόδοση των μοντέλων στα ελληνικά ήταν αρκετά μέτρια, ακόμα και των μοντέλων που είχαν γίνει fine-tuned στην ελληνική γλώσσα, για αυτό επιλέχθηκε η λύση μετάφρασης των ερωταπαντήσεων με κατάλληλο μοντέλο

1.3 Δομή Διπλωματικής

Η δομή της διπλωματικής η οποία ακολουθήθηκε είναι αρχικά μία βιβλιογραφική ανασκόπηση του θέματος της εργασίας, με την εύρεση και την ανάλυση της. Η ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν καθώς και του θεωρητικού υποβάθρου όλων όσων αναφέρονται μετέπειτα στην εργασία. Ακολουθεί μία ανάλυση της αρχιτεκτονικής η οποία ακολουθήθηκε κατά την υλοποίηση της εφαρμογής. Έπειτα, είναι η ανάλυση του τρόπου αξιολόγησης των μοντέλων και τέλος τα συμπεράσματα και οι προτεινόμενες βελτιώσεις.

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση

2.1 Εισαγωγή

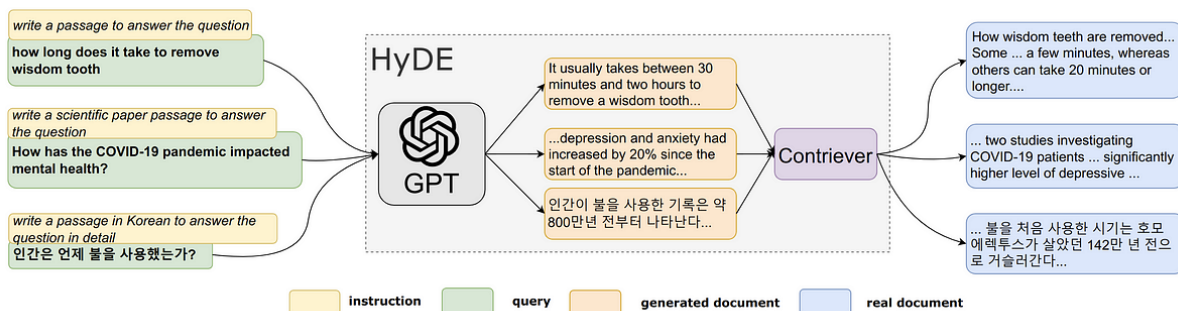
Η αρχιτεκτονική RAG και τα μεγάλα γλωσσικά μοντέλα αποτελούν πεδίο μεγάλου ενδιαφέροντος για την παγκόσμια βιβλιογραφία. Η δημοφιλία εφαρμογών με πράκτορες (agents) τεχνητής νοημοσύνης ανέδειξαν την χρησιμότητα που μπορεί να αποτελέσει ένας πράκτορας σε οποιαδήποτε καθημερινή δραστηριότητα, το υπολογιστικό κόστος, ωστόσο, για την χρήση ενός τέτοιου μοντέλου είναι πολύ μεγάλο και εκεί φάνηκε η χρησιμότητα της αρχιτεκτονικής RAG όπου επιτρέπει την χρήση πολύ μικρών μοντέλων, τα οποία αναλογικά έχουν μικρότερο κόστος, για την δημιουργία τέτοιων εφαρμογών σε ένα πολύ συγκεκριμένο πεδίο.

2.2 Βιβλιογραφική Ανασκόπηση

Η χρήση της αρχιτεκτονικής RAG σε συστήματα που αποτελούν πράκτορες τεχνητής νοημοσύνης δείχνουν να έχουν πολύ καλή απόδοση τόσο στην ανάκτηση πληροφορίας όσο και στην ικανοποίηση του χρήστη, ώντας συστήματα τα οποία χρησιμοποιούνται σε συστήματα ερωταπαντήσεων (question answering systems, QA systems), chatbots και διαλογικούς πράκτορες Ajit Singh et al. [2]

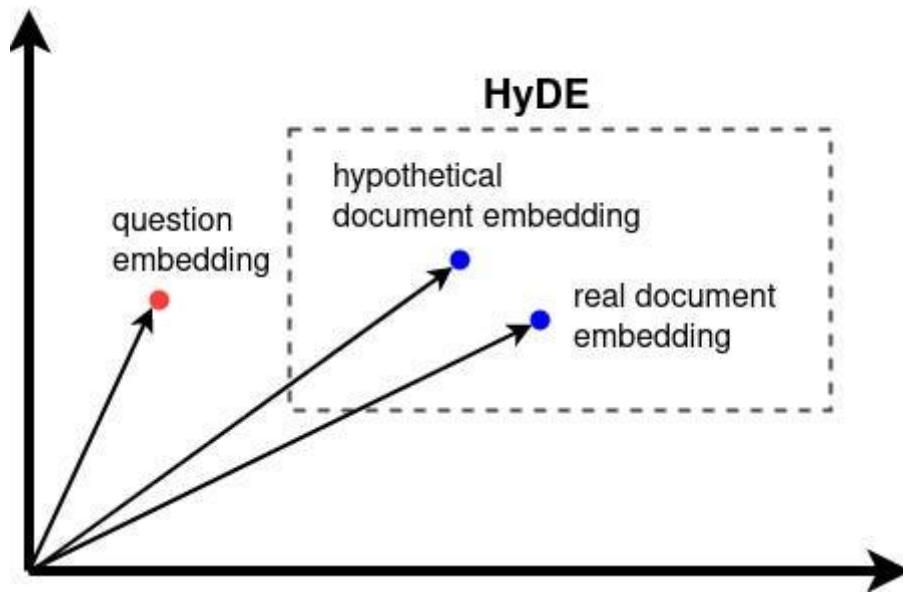
Στη σύγχρονη βιβλιογραφία, αναφερόμαστε σε μοντέλα όπως το Mistral 7B, το Gemini κ.ο.κ. ως μεγάλα γλωσσικά μοντέλα, ωστόσο η αρχιτεκτονική RAG προϋπάρχει αυτών και είχε αναπτυχθεί από την Facebook AI Research [1] με μοντέλα τύπου BART [3] και T5-11b [4]. Τα αποτελέσματα της χρήσης της αρχιτεκτονικής αυτής ήταν εφάμιλλα με αυτά των state-of-the-art μοντέλων, ωστόσο ο κόσμος προτιμούσε τα αποτελέσματα με την μέθοδο RAG ως πιο ακριβή και βασισμένα στην πραγματικότητα.

Η αρχιτεκτονική RAG απαιτεί την ρύθμιση πολλών Υπερπαραμέτρων τόσο των μεγάλων γλωσσικών μοντέλων όσο και των μοντέλων διανυσματοποίησης (embedding models) καθώς και τη διανυσματικής βάσης δεδομένων (vector database). Μια σειρά διεργασιών (pipeline) της RAG αρχιτεκτονικής μπορεί γρήγορα να γίνει περίπλοκη προκειμένου να βελτιστοποιηθεί η ποιότητα των απαντήσεων. Οι X. Wang et al. [5] χρησιμοποιούν έναν κατηγοριοποιητή (classifier) για να αποφασίσει το είδος και αν θα γίνει αναζήτηση δεδομένων ή αν η ικανότητα του μοντέλου αρκεί για να απαντήσει την ερώτηση. Έπειτα, χρησιμοποιούν μία υβριδική αναζήτηση (Hybrid Search) με την τεχνολογία HyDE (Hypothetical Document Embeddings). Η υβριδική αναζήτηση αποτελείται από δύο μέρη την αραιή (sparse) και την πυκνή (dense) ανάκτηση, η αραιή ανάκτηση αναζητά και επιστρέφει κείμενα βασισμένα σε λέξεις ή keywords ενώ η πυκνή ανάκτηση το κάνει με την χρήση των embeddings. Η μέθοδος HyDE χρησιμοποιεί ένα μεγάλο γλωσσικό μοντέλο για να δημιουργήσει υποθετικά δεδομένα και έπειτα η αναζήτηση στη βάση γίνεται με αυτά, και όχι με το input που έδωσε ο χρήστης.



Σχήμα 1: Αρχιτεκτονική HyDE

Με την μέθοδο HyDE τα αποτελέσματα είναι πολύ καλύτερα ωστόσο απαιτεί μεγάλη υπολογιστική δύναμη και πολύ μεγαλύτερο χρόνο αναμονής του χρήστη μέχρι την απάντηση του συστήματος.



Σχήμα 2: Αποτελέσματα της μεθόδου HyDE

Τέλος, χρησιμοποιούν το monoT5 [6], ένα transformer-based μοντέλο (ακολουθεί την αρχιτεκτονική transformers [7]) ώστε να κάνει reranking αξιολογώντας ποια έγγραφα από αυτά που ανακτήθηκαν θα είναι πιο χρήσιμα.

Τα συστήματα πρακτόρων με LLM μπορούν να χρησιμοποιηθούν συνεργατικά μεταξύ τους, δημιουργώντας συστήματα πολλών πρακτόρων (multi-agent systems). Οι Guo et al. [8] έκαναν μια ενδελεχή έρευνα στα πολυπρακτορικά συστήματα με μεγάλα γλωσσικά μοντέλα, αναδεικνύοντας τις ικανότητες τους να αναλύουν και να προσομοιώνουν σύνθετα, συνεργατικά περιβάλλοντα. Χρησιμοποιώντας συστήματα με πολλούς εξειδικευμένους πράκτορες να συνεργάζονται μπορούν να καλύψουν ένα πολύ ευρύ φάσμα προβλημάτων στα οποία μπορούν να δώσουν λύσεις.

Στην έρευνα τους, οι A. Singh et al. [9] ερευνούν τους διαφορετικούς τύπους της αρχιτεκτονικής RAG χωρίζοντας τους σε 5 κατηγορίες, αφελής (naive) RAG είναι ένας διαφορετικός τίτλος για το sparse retrieval όπου η ανάκτηση εγγράφων γίνεται με την χρήση keywords. Προχωρημένη (advanced) RAG είναι αντίστοιχα ο τίτλος της πυκνής (dense) ανάκτησης εγγράφων βασισμένη στην σημασία και το context με την χρήση των embeddings. Αρθρωτή (Modular) RAG όπου είναι η υβριδική μέθοδος που χρησιμοποιεί τις δύο προηγούμενες μεθόδους. RAG με την χρήση γράφου (Graph RAG) όπου τα δεδομένα στη βάση είναι αποθηκευμένα σε μορφή γράφου με τις συνδέσεις μεταξύ τους να αποτελούν έξτρα πληροφορία πέρα από τα ίδια τα έγγραφα. Τέλος, το agentic RAG μέσω του οποίου το ίδιο το σύστημα αποφασίζει ποια είναι η καλύτερη μέθοδος ανάκτησης και αξιολογεί την ποιότητα της αποκτηθείσας πληροφορίας. Είναι εμφανές στην έρευνα ότι η τελευταία μέθοδος αποδίδει καλύτερα, ωστόσο απαιτεί μεγάλη υπολογιστική ισχύ και αυξάνει αρκετά τον χρόνο που απαιτείται για να απαντήσει το σύστημα στον χρήστη.

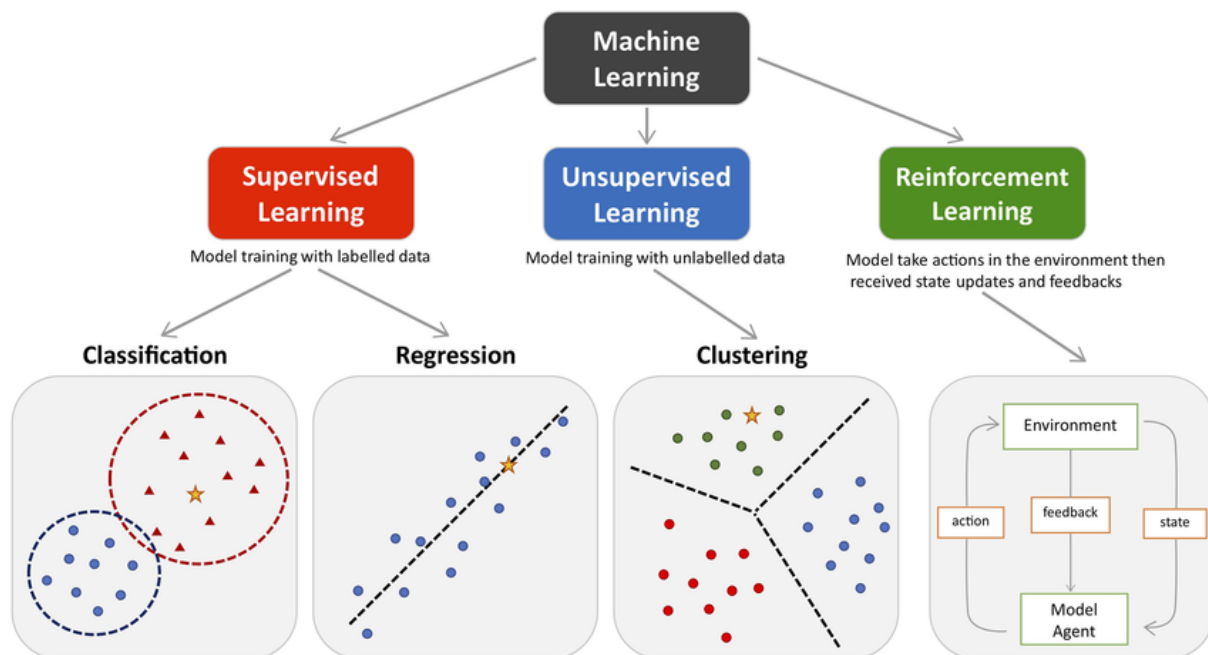
Κεφάλαιο 3ο: Θεωρητικό Υπόβαθρο

3.1 Εισαγωγή

Για την υλοποίηση του πράκτορα τεχνητής νοημοσύνης απαιτείται η υλοποίηση και η διασύνδεση πολλών τμημάτων τα οποία οφείλουν να συνεργάζονται αρμονικά. Τα τμήματα αυτά αποτελούνται και από μοντέλα μηχανικής μάθησης τα οποία απαιτούν εις βάθος γνώση των αρχών που τα διέπουν ώστε να μπορεί να γίνει μεγιστοποίηση της αξιοποίησης των ικανοτήτων τους. Είναι σημαντικό να αναλυθούν εις βάθος όχι μόνο οι χρησιμοποιούμενες τεχνολογίες αλλά και οι θεμελιώδεις έννοιες οι οποίες διέπουν αυτές.

3.2 Μηχανική Μάθηση

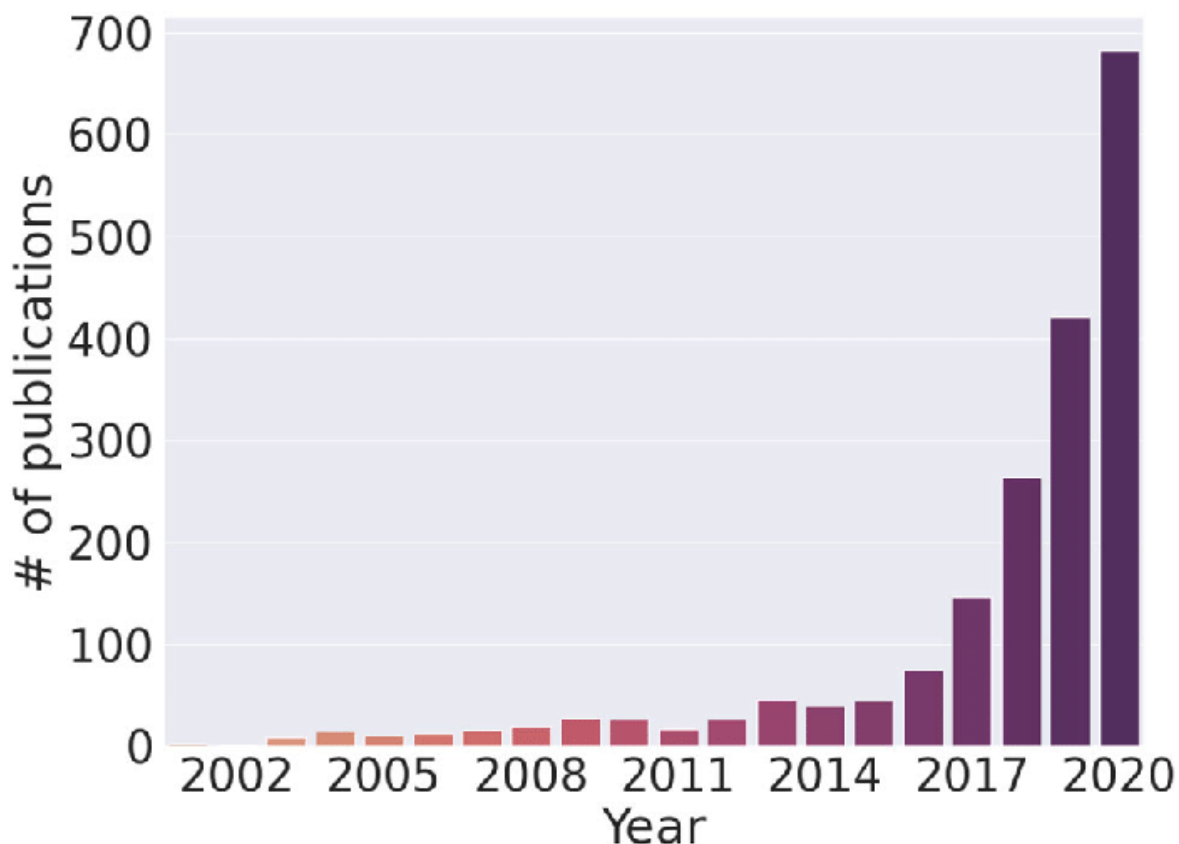
Μηχανική μάθηση ονομάζουμε την χρήση και την ανάπτυξη υπολογιστικών συστημάτων, τα οποία, μπορούν να μάθουν και να προσαρμοστούν χωρίς να λάβουν επιπλέον συγκεκριμένες οδηγίες, χρησιμοποιώντας αλγόριθμους και στατιστικά μοντέλα για να αναλύσουν και να κατανοήσουν μοτίβα στα δεδομένα (Ορισμός από το Oxford Languages). Η μηχανική μάθηση, επιτρέπει στα συστήματα που την χρησιμοποιούν, να μαθαίνουν τα μοτίβα των δεδομένων εκμάθησης (training data) και να γενικεύουν, αξιοποιώντας την γνώση και την αναγνώριση μοτίβων σε πραγματικά δεδομένα. Οι κύριες κατηγορίες τις οποίες μπορούμε να διακρίνουμε στη μηχανική μάθηση είναι η εποπτευόμενη (supervised), η μη-εποπτευόμενη (unsupervised) και η ενισχυτική (reinforcement) μάθηση [10].



Σχήμα 3: Τύποι Μηχανικής Μάθησης

Η μηχανική μάθηση υλοποιείται σε πολλά διαφορετικά συστήματα και πεδία, επιλύοντας προβλήματα με μεγάλη ποικιλία στην μορφή και το είδος τους. Προβλήματα κατηγοριοποίησης (classification problems) τόσο σε δεδομένα κειμένου, αριθμητικά, όσο και εικόνας, προβλήματα πρόβλεψης (forecasting) όπως είναι οι προβλέψεις σε χρονοσειρές (time-series forecasting), καθώς και η επεξεργασία φυσικής γλώσσας (Natural Language Processing, NLP) ένας κλάδος με τεράστια άνθηση χάρη στην αρχιτεκτονική μοντέλου Transformer και τον μηχανισμό “προσοχή” (attention) [7]. Οι εφαρμογές μηχανικής μάθησης μπορούν να εμφανιστούν σε οποιοδήποτε πεδίο έχει πολλά καλώς

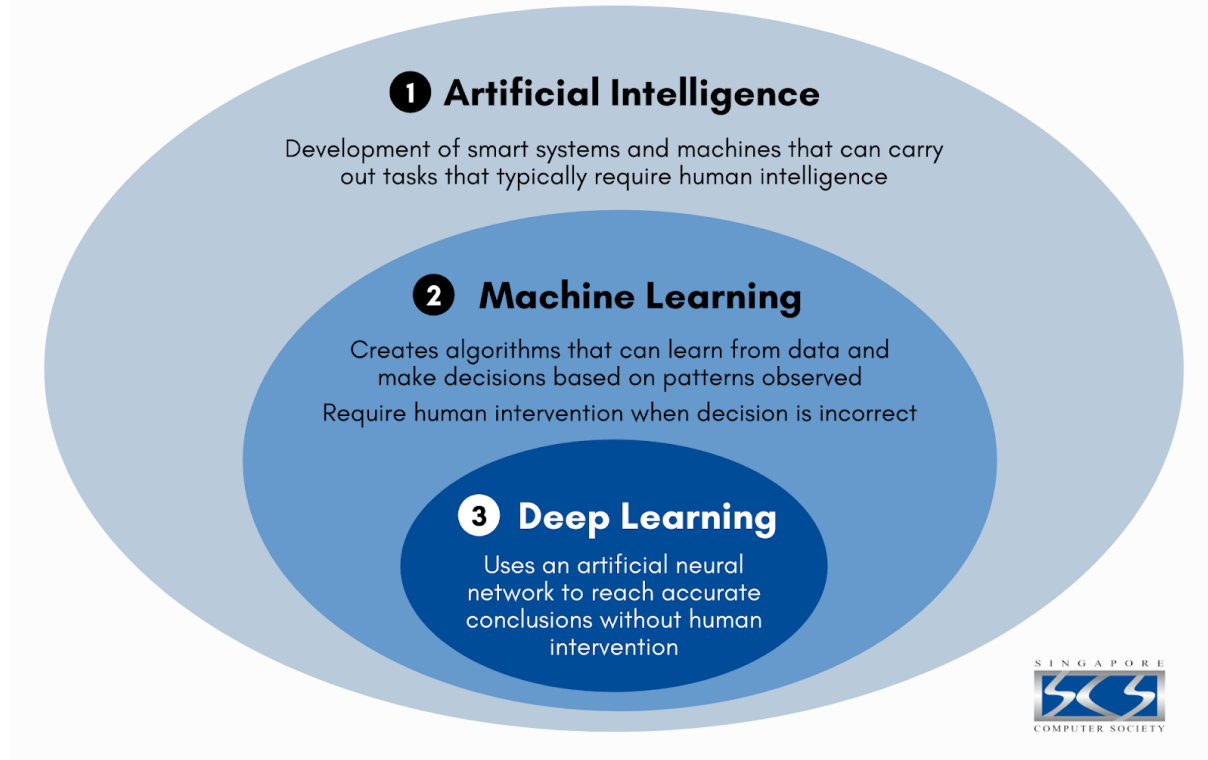
δομημένα δεδομένα και προβλήματα που ακολουθούν μοτίβα. Παράδειγμα αποτελεί η άνοδος της χρήσης της μηχανικής μάθησης τα τελευταία χρόνια σε έρευνες που δημοσιεύτηκαν στο “American Chemical Society”, οι οποίες αγγίζουν τις 700 το 2020 [11].



Σχήμα 4: Η Ανάπτυξη των δημοσιεύσεων με Μηχανική Μάθηση στη Χημεία

Στη σύγχρονη εποχή, μηχανική μάθηση και τεχνητή νοημοσύνη συχνά συγχέονται, με την μηχανική μάθηση να αποτελεί υποσύνολο της τεχνητής νοημοσύνης. Η αύξηση των διαθέσιμων δεδομένων σε συνδυασμό με τις ικανότητες του σύγχρονου υλισμικού (hardware), έχουν δώσει τεράστια ώθηση στην έρευνα και την ανάπτυξη της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Η δημοφιλία εφαρμογών όπως το ChatGPT, καθώς και η ενσωμάτωση βοηθών τεχνητής νοημοσύνης σε ήδη δημοφιλή λογισμικά όπως το Gemini στο Google Search και το Android, αλλά και το Meta AI στα κοινωνικά δίκτυα της Meta, αποδεικνύουν την άνθιση συστημάτων αρκετά ικανών και χρήσιμων ώστε να χρησιμοποιούνται από αρκετούς χρήστες αλλά και την τάση της βιομηχανίας να ερευνήσει και να αναπτύξει τέτοιου είδους συστήματα.

ARTIFICIAL INTELLIGENCE VS MACHINE LEARNING VS DEEP LEARNING



Σχήμα 5: Οι έννοιες Τεχνητή Νοημοσύνη, Μηχανική Μάθηση και Βαθιά Μηχανική Μάθηση

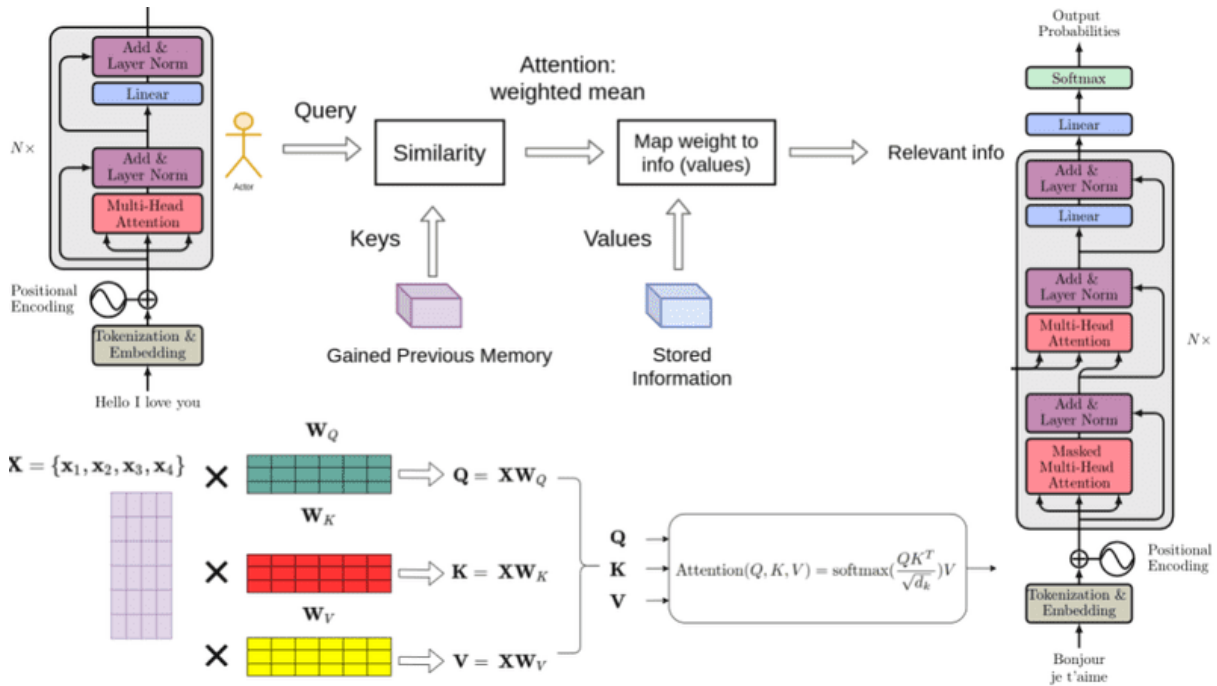
3.3 Αρχιτεκτονική Transformers

Η αρχιτεκτονική transformers αποτελεί ακρογωνιαίο λίθο στην μηχανική μάθηση για ζητήματα που αφορούν την επεξεργασία φυσικής γλώσσας (NLP). Τα μοντέλα που ακολουθούν την αρχιτεκτονική αυτή χρησιμοποιούν τον μηχανισμό της «προσοχής» (attention) για να εντοπίζουν την επιρροή και την εξάρτηση των δεδομένων εισόδου μεταξύ τους.

Τα μοντέλα αυτά αποτελούνται από δύο μέρη, τον κωδικοποιητή (encoder) και τον αποκωδικοποιητή (decoder), μία αρχιτεκτονική που μοιάζει με αυτή των αναδρομικών νευρωνικών δικτύων (Recurrent Neural Networks, RNNs). Τα μοντέλα Transformers σε αντίθεση με τα RNNs, δεν επεξεργάζονται τα δεδομένα σε ακολουθητική σειρά (sequential) αλλά όλα μαζί, δίνοντας την δυνατότητα ταχύτερης εκπαίδευσης και παραλληλοποίησης (parallelization). Στην διαδικασία της εκπαίδευσης, στον encoder δίνονται δεδομένα τα οποία παρομοιάζουν το input στην διαδικασία πρόβλεψης, ενώ στον decoder δίνονται δεδομένα που παρομοιάζουν το output στην διαδικασία της πρόβλεψης. Κατά την διάρκεια της πρόβλεψης, στον encoder εισέρχονται τα δεδομένα του input ενώ στον decoder ένα token «[BOS]» (begin of sequence) και όταν το μοντέλο κρίνει ότι πρέπει να σταματήσει παράγει ένα token «[EOS]» (end of sequence).

Η αρχιτεκτονική transformers έχει μονοπωλήσει την επεξεργασία φυσικής γλώσσας σε πάρα πολλές εφαρμογές της. Το μοντέλο BERT [12] και οι παραλλαγές του, αξιοποιώντας μόνο το κομμάτι του κωδικοποιητή της εφαρμογής, θεωρούνται state-of-the-art για εφαρμογές όπως, κατηγοριοποίηση

κειμένον (text classification), αναγνώριση ονοματικών οντοτήτων (Name Entity Recognition, NER), διανυσματοποίηση κειμένου (embedding generation) καθώς και σε αρκετές ακόμα εφαρμογές. Το κομμάτι του αποκωδικοποιητή μόνο, χρησιμοποιούν μοντέλα όπως τα GPT (Generation Pretrained Transformer) τα οποία χρησιμοποιούνται στην δημιουργία κειμένου (text generation), τους έξυπνους βοηθούς, τα chatbots και οποιαδήποτε εφαρμογή αφορά την παραγωγή κειμένου σε εξειδικευμένο πεδίο, όπως η συμπλήρωση κενού, η παραγωγή περίληψης κειμένου κ.ο.κ. Μοντέλα που χρησιμοποιούν και τα δύο μέρη της αρχιτεκτονικής, χρησιμοποιούνται στην μετάφραση (machine translation), όπως το μοντέλο BART [3]. Τέλος, η αρχιτεκτονική των transformers έχει εισέλθει και σε άλλους τομείς με τους ViT (Visual Transformers) για κατηγοριοποίηση εικόνων (Image Classification)[13] καθώς και στην πρόβλεψη χρονοσειρών (time series forecasting) [14].



Σχήμα 6: Η αρχιτεκτονική Transformers

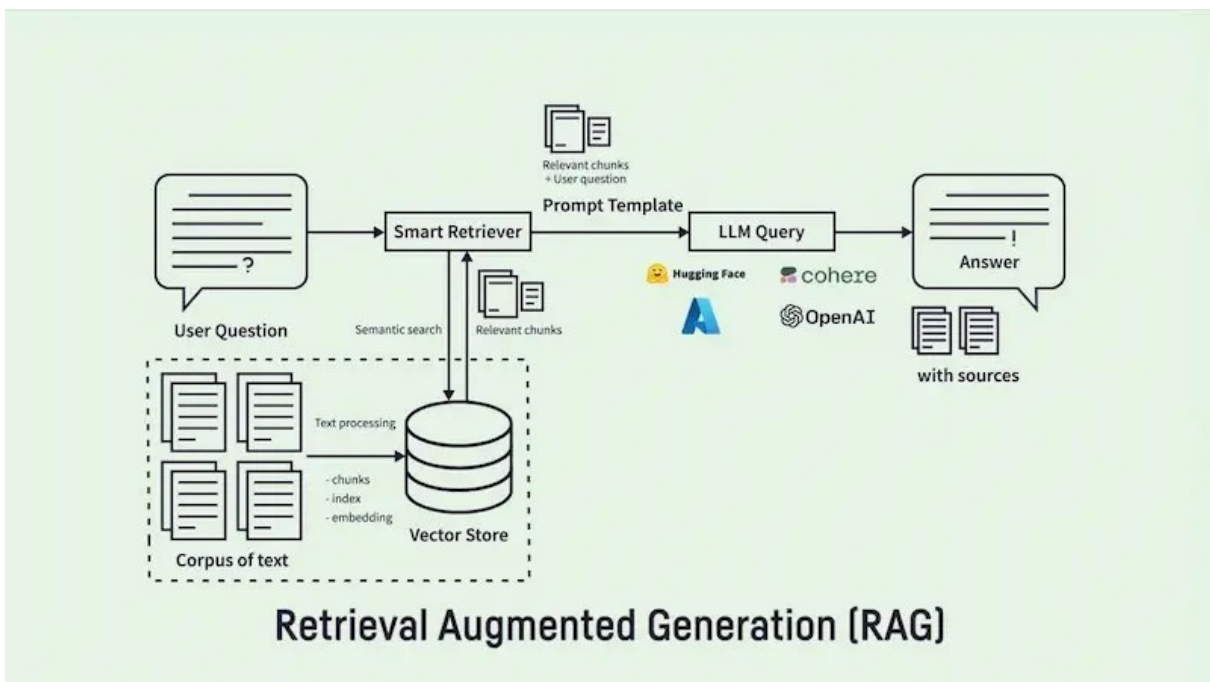
3.4 Η Αρχιτεκτονική RAG

Η αρχιτεκτονική RAG (Retrieval Augmented Generation) αποτελεί τα τελευταία χρόνια, τον κύριο τρόπο αξιοποίησης των μικρότερων, μεγάλων γλωσσικών μοντέλων στην βιομηχανία. Η ανάπτυξη των μεγάλων γλωσσικών μοντέλων εστιασμένων στην κατανόηση και παραγωγή φυσικής γλώσσας είναι ραγδαία τα τελευταία χρόνια. Το σημαντικότερο πρόβλημα που αντιμετωπίζουν για την είσοδο τους στην βιομηχανία είναι οι απαιτήσεις σε hardware και το κόστος εκπαίδευσης, ειδικότερα αν πρόκειται για τα πιο ισχυρά μοντέλα. Τα μικρότερα, συγκριτικά, μοντέλα αντιμετωπίζουν το πρόβλημα των hallucinations, την δημιουργία κειμένου δηλαδή το οποίο είναι λάθος δίνοντας έτσι ψευδείς απαντήσεις. Η αρχιτεκτονική RAG αποτελεί την λύση, σε πολύ μεγάλο βαθμό, σε αυτό το πρόβλημα όταν φυσικά οι ερωτήσεις αφορούν κάτι το οποίο βρίσκεται στην βάση δεδομένων. Χάρη στην αρχιτεκτονική RAG τα chatbot έχουν γίνει πολύ πιο προσιτά και δημοφιλή, δίνοντας τη δυνατότητα να υπάρξουν ακόμα και σε σελίδες και εφαρμογές μικρών εταιρειών.

Αναλυτικότερα, η αρχιτεκτονική RAG απαιτεί την ύπαρξη δύο μερών, ενός μεγάλου γλωσσικού μοντέλου και μίας διανυσματικής βάσης δεδομένων. Στην διανυσματική βάση αποθηκεύονται τα δεδομένα βάσει των οποίων θέλουμε να απαντάει το μεγάλο γλωσσικό μοντέλο. Τα δεδομένα αυτά,

αποθηκεύονται στην βάση με την μορφή διανυσμάτων, με το κείμενο να μετατρέπεται σε embeddings και να αποθηκεύεται. Όταν, ο χρήστης στέλνει ένα ερώτημα στο chatbot, το ερώτημα του πηγαίνει στην διανυσματική βάση, η οποία με την σειρά της κάνει ένα similarity search και μαζί με το αρχικό ερώτημα του χρήστη στέλνει και τα σχετικότερα δεδομένα που έχει αποθηκευμένα, στο μεγάλο γλωσσικό μοντέλο με σκοπό αυτό να απαντήσει όχι με βάση τα δεδομένα εκπαίδευσης του αλλά με βάση αυτά που έχει αποθηκευμένα η βάση. Η χρήση των διανυσματικών βάσεων δεδομένων γίνεται λόγω της δυνατότητας τους να προσφέρουν με μεγάλη ταχύτητα πολύ καλές απαντήσεις κάνοντας χρήση της αναζήτησης «προσεγγιστικά κοντινότερου γείτονα» (Approximate Nearest Neighbor, ANN), η chromadb που χρησιμοποιήθηκε για την ανάπτυξη της δικής μου εφαρμογής χρησιμοποιεί τον αλγόριθμο HNSW (Hierarchical Navigable Small Worlds), ο οποίος έχει εξαιρετικά αποτελέσματα σε αναζητήσεις τόσο σε κείμενο όσο και σε άλλες μορφές [15].

Με την χρήση της αρχιτεκτονικής RAG η βελτίωση στην απόδοση μικρών LLMs είναι μεγάλη, καθώς παρά την μικρή τους ικανότητα στην σωστή παραγωγή κειμένου λόγω του μεγέθους, διατηρούν την ικανότητα να κατανοούν κείμενο και να παράγουν κείμενο σε μορφή διαλόγου βάσει του δωσμένου context.



Σχήμα 7: Η αρχιτεκτονική RAG

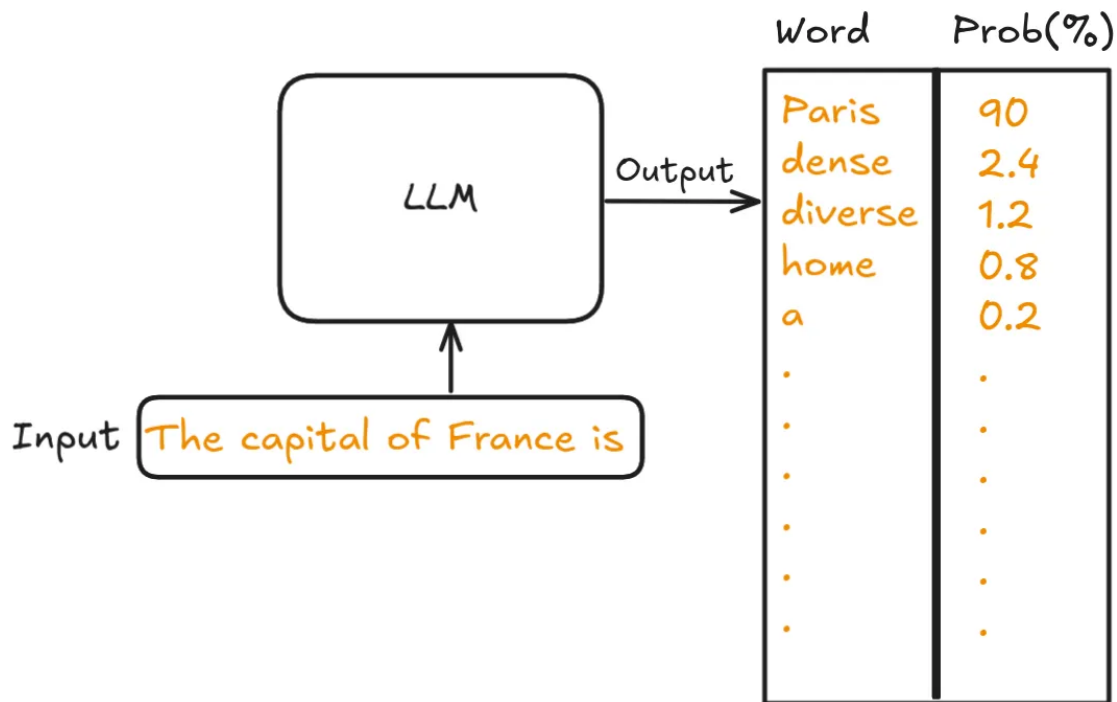
3.5 Μεγάλα Γλωσσικά Μοντέλα (Large Language Models, LLMs)

Τα μεγάλα γλωσσικά μοντέλα αποτελούν, κατά κύριο λόγο, μοντέλα που ακολουθούν την αρχιτεκτονική transformers, και παίρνουν το όνομα τους από το πολύ μεγάλο μέγεθος τους. Χρησιμοποιούνται σε εφαρμογές που αφορούν την κατανόηση και την παραγωγή φυσικού λόγου και έχουν πολύ μεγάλη άνθιση τα τελευταία χρόνια, τόσο στην βιομηχανία όσο και στην έρευνα. Οι μεγαλύτεροι τεχνολογικοί κολοσσοί καθώς και ερευνητικές ομάδες από όλο τον κόσμο έχουν επενδύσει πολλά στην έρευνα και δημιουργία μεγάλων γλωσσικών μοντέλων.

Ένα μεγάλο γλωσσικό μοντέλο, είναι ένα τεράστιο νευρωνικό δίκτυο εκπαιδευμένο να προβλέπει το επόμενο token βάσει των προηγούμενων δοσμένων tokens. Ένα token αποτελεί κάθε λέξη, κάθε σημείο στίξης, νέα γραμμή, ειδικός χαρακτήρας κ.ο.κ.. Κάθε στιγμή το μοντέλο επιλέγει κάποιες λέξεις οι

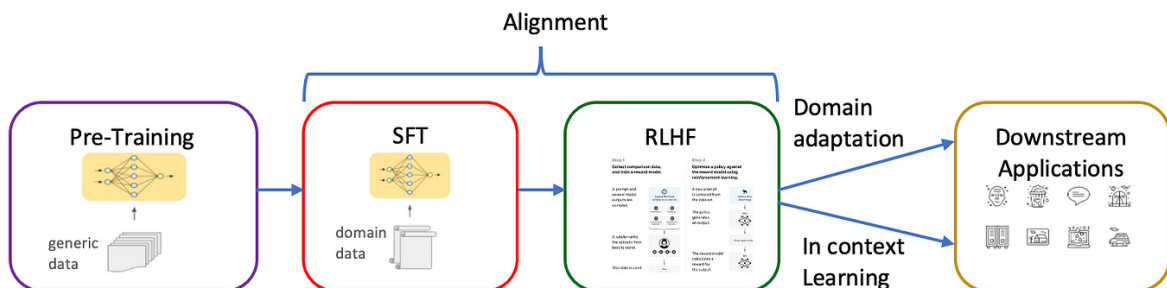
Θεωρητικό Υπόβαθρο

οποίες έχουν μία λογική συνέχεια συγκριτικά με το κείμενο εισόδου, στις οποίες δίνει μία πιθανότητα να επιλεγούν. Ένα παράδειγμα, με είσοδο “The cat sat on the “, ένα μεγάλο γλωσσικό μοντέλο μπορεί να έχει τις απαντήσεις: [mat, 0.9], [chair, 0.07], [roof, 0.03] και να επιλέξει βάσει αυτών το output του.



Σχήμα 8: Παράδειγμα λειτουργίας Μεγάλου Γλωσσικού μοντέλου

Τα μοντέλα αυτά εκπαιδεύονται σε μία τεράστια βάση δεδομένων αποτελούμενη από βιβλία, άρθρα, επιστημονικές δημοσιεύσεις και γενικότερα σε οποιαδήποτε ελεύθερη βάση γνώσης κειμένου διαθέτουν, ωστόσο λόγω της μορφής αυτών των κειμένων, τα μοντέλα σε αυτή τη φάση παράγουν λόγο που μοιάζει με αυτών που θα συναντήσουμε σε ένα βιβλίο περισσότερο από μία επικοινωνία ή διάλογο. Αυτή η φάση ονομάζεται Pretraining. Στην συνέχεια, γίνεται fine-tuning των μοντέλων αυτών σε διαλόγους τόσο με supervised Fine-tuning (εποπτευόμενη επανεκπαίδευση), όπου δίνονται πολλά μικρά αποσπάσματα ή και ολόκληροι διάλογοι, όσο και με Reinforcement Learning from Human Feedback (Ενισχυτική Μάθηση με Ανθρώπινη Ανατροφοδότηση) όπου γίνεται βαθμολόγηση και αξιολόγηση των απαντήσεων του μοντέλου από ανθρώπους, έτσι το μεγάλο γλωσσικό μοντέλο εκπαιδεύεται να απαντά με τρόπο που προτιμάται από τους ανθρώπους.



Σχήμα 9: Διαδικασία εκπαίδευσης ενός Μεγάλου Γλωσσικού Μοντέλου

3.6 Αναγνώριση Γλώσσας και Μετάφραση

Η μηχανική μετάφραση (machine translation) ήταν μία από τις δημοφιλέστερες εφαρμογές του διαδικτύου από την εμφάνιση της. Μηχανική μετάφραση ονομάζεται το πεδίο της επιστήμης των υπολογιστών που μελετά την αυτόματη μετάφραση από μία γλώσσα σε μία άλλη.

Παρά το γεγονός ότι τα μεγάλα γλωσσικά μοντέλα εκπαιδεύονται σε μεγάλους όγκους δεδομένων από πληθώρα γλωσσών, συνεχίζουν να παρουσιάζουν καλύτερα αποτελέσματα, κατα κύριο λόγο στα αγγλικά, λόγω της διαθεσιμότητας δεδομένων στην αγγλική γλώσσα κατά την διάρκεια της εκπαίδευσης. Η μηχανική μετάφραση βρίσκεται σε ένα πολύ ικανοποιητικό επίπεδο, δίνοντας έτσι την δυνατότητα μετάφρασης τόσο της εισόδου του χρήστη στα αγγλικά με σκοπό την καλύτερη απόδοση του μεγάλου γλωσσικού μοντέλου, όσο και της μετάφρασης της απάντησης του μοντέλου στην αρχική γλώσσα την οποία επέλεξε ο χρήστης.

Για την μετάφραση χρησιμοποιήθηκε άλλο ένα transformer based μοντέλο, το seamless4t από την meta [16], ένα μοντέλο που έχει δυνατότητα μετάφρασης από ήχο ή και γραπτό κείμενο σε ήχο ή και γραπτό κείμενο. Με την επίδοση του στο BLEU (Bilingual Evaluation Understudy), της δημοφιλέστερης μετρικής αξιολόγησης για την μηχανική μετάφραση, να είναι στα επίπεδα του state-of-the-art για text-to-text translation (μετάφραση από κείμενο σε κείμενο), ενώ ξεπερνάει όλες τις υπάρχουσες λύσεις στο voice-to-voice και voice-to-text translation.

Τέλος, για την αναγνώριση της γλώσσας χρησιμοποιείται η βιβλιοθήκη της python langdetect. Η langdetect χρησιμοποιεί την στατιστική μέθοδο Naive Bayes classifier, όπου πιθανοκρατικά κατηγοριοποιεί το κείμενο που εισάγεται από τον χρήστη, σε μία γλώσσα, αξιολογώντας την συχνότητα εμφάνισης των n-grams. Ένας κατηγοριοποιητής Naive Bayes, βασίζεται στο θεώρημα του Bayes:

$$P(\text{Κατηγορία} \mid \text{Χαρακτηριστικά}) = \frac{P(\text{Χαρακτηριστικά} \mid \text{Κατηγορία}) \times P(\text{Κατηγορία})}{P(\text{Χαρακτηριστικά})} \quad (1)$$

Κατά την εκπαίδευση, γίνεται υπολογισμός της πιθανότητας εμφάνισης κάθε n-gram σε μία κατηγορία ($P(\text{κατηγορία})$), καθώς και την συχνότητα εμφάνισης του κάθε n-gram σε κάθε κατηγορία ($P(\text{Χαρακτηριστικό} \mid \text{Κατηγορία})$). Κατά την διάρκεια της πρόβλεψης για κάθε κατηγορία-στην δική μας περίπτωση γλώσσα-υπολογίζουμε το σκορ σύμφωνα με την σχέση:

$$\text{Σκορ} = P(\text{Κατηγορία}) \times \prod P(\text{Χαρακτηριστικό}(i) \mid \text{Κατηγορία}) \quad (2)$$

Τέλος, επιλέγουμε την κατηγορία (γλώσσα) με το μεγαλύτερο σκορ.

3.7 Pytorch και CUDA

Η Pytorch είναι μία βιβλιοθήκη της python η οποία επιτρέπει την δημιουργία και την χρήση νευρωνικών δικτύων. Χρησιμοποιείται τόσο για την δημιουργία και την εκπαίδευση νέων νευρωνικών δικτύων όσο και για την χρήση έτοιμων νευρωνικών δικτύων και αρχιτεκτονικών ώστε να ρυθμιστεί ο τρόπος με τον οποίο τρέχουν όπως και οι συσκευές στην οποία θα τρέξουν. Η Pytorch αναπτύχθηκε από την Meta AI το 2016, και μαζί με την βιβλιοθήκη TensorFlow αποτελούν την ραχοκοκαλιά της ανάπτυξης νευρωνικών δικτύων.

Η CUDA είναι μία πλατφόρμα, η οποία παίρνει το όνομα της από τα αρχικά των λέξεων Compute Unified Device Architecture και αναπτύχθηκε από την Nvidia το 2006. Η πλατφόρμα αυτή επιτρέπει

την χρήση καρτών γραφικών της Nvidia κατά την εκτέλεση προγραμμάτων που χρησιμοποιούν μηχανική μάθηση και νευρωνικά δίκτυα, επιταχύνοντας έτσι την διαδικασία εκπαίδευσης και πρόβλεψης, λόγω της φύσης των καρτών γραφικών να υλοποιούν πολύ γρήγορα μεγάλο αριθμό αριθμητικών πράξεων. Επιπλέον, δίνουν την δυνατότητα παράλληλης επεξεργασίας και εκπαίδευσης σε περίπτωση ύπαρξης περισσότερων από μία κάρτες γραφικών.

3.8 Διεπαφή Προγραμματισμού Εφαρμογών

Η διεπαφή προγραμματισμού εφαρμογών (application programming interface, API) γνωστή και ως Διασύνδεση Προγραμματισμού Εφαρμογών είναι η διεπαφή που παρέχουν τα λειτουργικά ή τα λογισμικά προκειμένου να γίνεται επικοινωνία με άλλα λειτουργικά. Η χρήση των APIs είναι διαδεδομένη για την διασύνδεση διαφόρων εφαρμογών μεταξύ τους, καθώς και των τμημάτων μιας ολόκληρης εφαρμογής. Με την χρήση των APIs επιτυγχάνεται η διασύνδεση της κύριας λειτουργίας μίας εφαρμογής με την διεπαφή του χρήστη (διασύνδεση backend-frontend). Τέλος, υπάρχουν αρκετά APIs που προσφέρουν υπηρεσίες ελεύθερα για χρήση από άλλες εφαρμογές, παράδειγμα αποτελούν τα APIs τα οποία προσφέρουν πρόβλεψη του καιρού, πρόσβαση σε άρθρα κ.ο.κ.

Η επικοινωνία με ένα API γίνεται συνήθως με την χρήση των HTTP requests όπου συνήθως στα headers υπάρχει ένας τρόπος αυθεντικοποίησης του χρήστη, συνήθως με ένα μοναδικό κλειδί του χρήστη (api-key) ενώ στο body, ακολουθείται η προβλεπόμενη, από τον δημιουργό του API, δομή ώστε να υπάρχουν τα απαιτούμενα δεδομένα προκειμένου να λάβει σωστά την απάντηση.

Ένα API αποτελείται από ένα ή περισσότερα endpoints. Αναλογικά, θα μπορούσαμε να σκεφτούμε το API σαν ένα εστιατόριο ενώ το endpoint σαν ένα φαγητό. Τα endpoints είναι σημεία όπου το API δέχεται τα αιτήματα και δρα ώστε να φέρει εις πέρας το αίτημα του χρήστη και να απαντήσει αναλόγως.

3.9 Flask και Gunicorn

Το framework Flask της python αποτελεί ένα βασικό εργαλείο υλοποίησης backend για την γλώσσα της python με σκοπό την δημιουργία APIs και endpoints. Ανήκει στην κατηγορία των microframeworks καθώς δεν απαιτεί την εγκατάσταση και χρήση συγκεκριμένων εργαλείων ή βιβλιοθηκών. Αποτελεί μία ελαφριά λύση με την δυνατότητα γρήγορης ανάπτυξης εφαρμογών διατηρώντας την απλότητα και την δυνατότητα χρήσης μόνο των απαιτούμενων εργαλείων που προσφέρονται. Παρά την απλότητα του, δίνει την δυνατότητα ανάπτυξης συντηρήσιμων και επεκτάσιμων APIs.

Για την πρόσβαση του Flask backend απαιτείται η χρήση ενός WSGI (Web Server Gateway Interface) HTTP server. Το Gunicorn αποτελεί έναν pre-fork worker model server, δηλαδή δημιουργεί πολλούς workers για να μπορεί να διαχειριστεί πολλαπλά αιτήματα ταυτόχρονα. Αυτός ο σχεδιασμός διευκολύνει την ικανότητα της εφαρμογής να αντιδρά σε πολλαπλά αιτήματα και προσδίδει λειτουργικότητα την οποία δεν έχει το Flask backend στον development server.

Το Gunicorn δρα ως ενδιάμεσο επίπεδο (middle layer) μεταξύ της εφαρμογής και του πελάτη (client), δέχεται τα http requests και στην συνέχεια τα δρομολογεί στην εφαρμογή, μετά το πέρας της επεξεργασίας του αιτήματος από το backend, δέχεται την απάντηση της εφαρμογής και την στέλνει πίσω στον client. Η χρήση του Gunicorn ή άλλου παρόμοιου service είναι απαραίτητη με την του Flask καθώς προσδίδει τόσο στην απόδοση της εφαρμογής όσο και στην σταθερότητα και στην ασφάλεια στην όσον αφορά την διαχείριση πολλών ταυτόχρονων αιτημάτων. Επιπρόσθετα, είναι συμβατό με πολλούς διαχειριστές διεργασιών και load balancers (όπως το Nginx), δίνοντας την δυνατότητα η εφαρμογή να μεγαλώσει οριζόντια.

3.10 Hugging Face και Transformers

Η πλατφόρμα Hugging Face αποτελεί μία από τις βασικότερες πλατφόρμες για την μηχανική μάθηση και κυρίως για την επεξεργασία φυσικής γλώσσας. Αποτελεί ένα αποθετήριο για μοντέλα μηχανικής

μάθησης, δεδομένων εκπαίδευσης καθώς και άρθρων σχετικών με την μηχανική μάθηση, καθώς και με οδηγίες και διευκολύνσεις για την χρήση αυτών των μοντέλων.

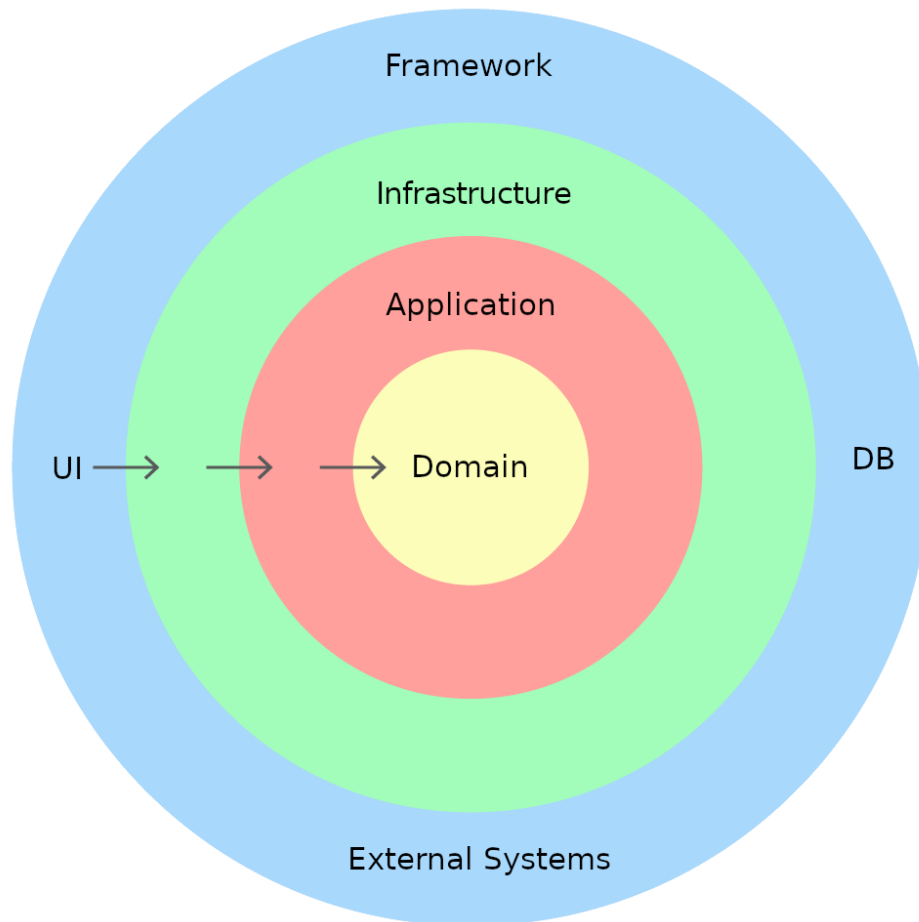
Το Hugging Face χρησιμοποιείται τόσο από τις εταιρείες και ερευνητικούς όπου δημοσιεύουν τα ελεύθερα προς χρήση μοντέλα τους όσο και από ιδιώτες όπου μοιράζουν fine-tuned εκδοχές μοντέλων σε συγκεκριμένα πεδία.

Η πλατφόρμα Hugging Face έχει δημιουργήσει και συντηρεί την βιβλιοθήκη Transformers της Python. Η βιβλιοθήκη Transformers είναι ανοιχτού κώδικα (open-source) και προσφέρει μια ολοκληρωμένη διεπαφή για την πρόσβαση και χρήση πολλών διαφορετικών μοντέλων που βασίζονται στην αρχιτεκτονική transformers και είναι διαθέσιμα στο Hugging face. Η βιβλιοθήκη δίνει την δυνατότητα άμεσης και αποτελεσματικής χρήσης πολλών διαφορετικών μοντέλων χωρίς να απαιτείται εξειδικευμένος κώδικας.

3.11 Clean Architecture

Κατά την δημιουργία εφαρμογών είναι σημαντικό ο κώδικας να είναι εύκολα κατανοησίμος και συντηρήσιμος. Οι μονολιθικές εφαρμογές αποτελούσαν ένα από τα σημαντικότερα προβλήματα της βιομηχανίας, καθιστώντας τις πολύ δύσκολα συντηρήσιμες. Στο βιβλίο του για την «Καθαρή Αρχιτεκτονική» (Clean Architecture) [17] ο Ρόμπερτ Μάρτιν θέσπισε τις αρχές του συντηρήσιμου κώδικα. Αρχικά, η λογική της εφαρμογής (business logic) πρέπει να είναι εντελώς διαχωρισμένη από οποιαδήποτε frameworks, βάσεις δεδομένων και διεπαφές. Συγκεκριμένα οι αρχές που θεσπίζονται στο βιβλίο είναι οι εξής:

- **Διαχωρισμός των Αρμοδιοτήτων:** Το σύστημα πρέπει να «σπάει» σε ξεκάθαρα επίπεδα εκ των οποίων το καθένα πρέπει να έχει μόνο μία αρμοδιότητα.
- **Ανεξαρτησία από τα Frameworks:** Η βασική λογική του προγράμματος θα πρέπει να μην εξαρτάται από συγκεκριμένες βιβλιοθήκες ή frameworks.
- **Η διεπαφή χρήστη είναι μία λεπτομέρεια:** Η διεπαφή του χρήστη, οποιασδήποτε μορφής, θα πρέπει να μπορεί να αλλάξει χωρίς να επηρεάζεται η βασική λογική της εφαρμογής.
- **Η βάση δεδομένων είναι μία λεπτομέρεια:** Οποιαδήποτε βάση δεδομένων θα πρέπει να μπορεί να αλλάξει χωρίς να επηρεάζεται καθόλου η βασική λογική της εφαρμογής.
- **Δοκιμαστικότητα (Testability):** Ο κώδικας θα πρέπει να μπορεί να ελεγχθεί εύκολα, γρήγορα και στοχευμένα, δεδομένου ότι η κύρια λογική της εφαρμογής είναι διαχωρισμένη από την υποδομή.
- **Κανόνας εξάρτησης (Dependency Rule):** Όλες οι εξαρτήσεις του πυρήνα της εφαρμογής πρέπει να αφορούν κομμάτια της βασικής λογικής της εφαρμογής και όχι κομμάτια της υποδομής. Σχηματικά τα επίπεδα συνήθως εμφανίζονται σε ομόκεντρους κύκλους με τις εξαρτήσεις να δείχνουν προς το εσωτερικό (πυρήνας).



Σχήμα 10: Επίπεδα του Clean Architecture

Στο domain ή core (πυρήνα) εμφανίζονται οι οντότητες (Entities) και οι γενικοί κανόνες του συστήματος. Αφορά τα κοινώς χρησιμοποιούμενα αντικείμενα και τις μεθόδους που αφορούν γενικές οντότητες, κοινώς χρησιμοποιούμενες από την εφαρμογή, οι οποίες δεν εξαρτώνται από οτιδήποτε.

Στο application layer ή use cases εμφανίζονται όλοι οι κανόνες και οι λειτουργίες που έχει μία εφαρμογή. Σε αυτό το επίπεδο υπάρχει η λογική σύμφωνα με την οποία θα διεκπαιρεύονται οι λειτουργίες της εφαρμογής χωρίς όμως να υπάρχει εξάρτηση από βιβλιοθήκες ή frameworks.

Στο επίπεδο της υποδομής (infrastructure layer) υπάρχουν οι βασικές υποδομές τις οποίες χρησιμοποιεί η εφαρμογή. Ο τρόπος διασύνδεσης με την βάση, οι υποδομές που υλοποιούν τις λειτουργίες που εμφανίζονται στο προηγούμενο επίπεδο κ.ο.κ.

Τέλος, στο εξωτερικό επίπεδο, εμφανίζονται τα frameworks που χρησιμοποιήθηκαν, οι βάσεις δεδομένων, εξωτερικά APIs, καθώς και η διεπαφή του χρήστη.

Η καθαρή αρχιτεκτονική είναι σημαντική για την ανάπτυξη προγραμμάτων με σκοπό την συνεχή ανάπτυξη και την εύκολη συντήρηση, διευκολύνοντας επίσης και την δυνατότητα ελέγχου της εφαρμογής. Με την χρήση της, διασφαλίζουμε ότι οποιαδήποτε αλλαγή σε υποδομές θα διατηρήσει την σωστή λειτουργία της εφαρμογής καθώς και την εύκολη μετάβαση από την χρήση μίας υποδομής σε μία άλλη. Τέλος, οι δυνατότητα του testing είναι πολύ πιο εύκολη όταν οι οντότητες και οι μέθοδοι της κύριας λειτουργίας του προγράμματος δεν εξαρτώνται από συγκεκριμένες υποδομές.

3.12 Discord

Το discord αποτελεί μία πολύ δημοφιλή πλατφόρμα για την επικοινωνία τόσο μέσω chatting όσο και με οπτικοακουστικά μέσα. Αρχικά είχε σχεδιαστεί για την επικοινωνία μεταξύ παικτών ηλεκτρονικών παιχνιδιών αλλά πλέον αποτελεί μία πολύ δημοφιλή πλατφόρμα για κοινότητες, εταιρείες και ερευνητικούς οργανισμούς.

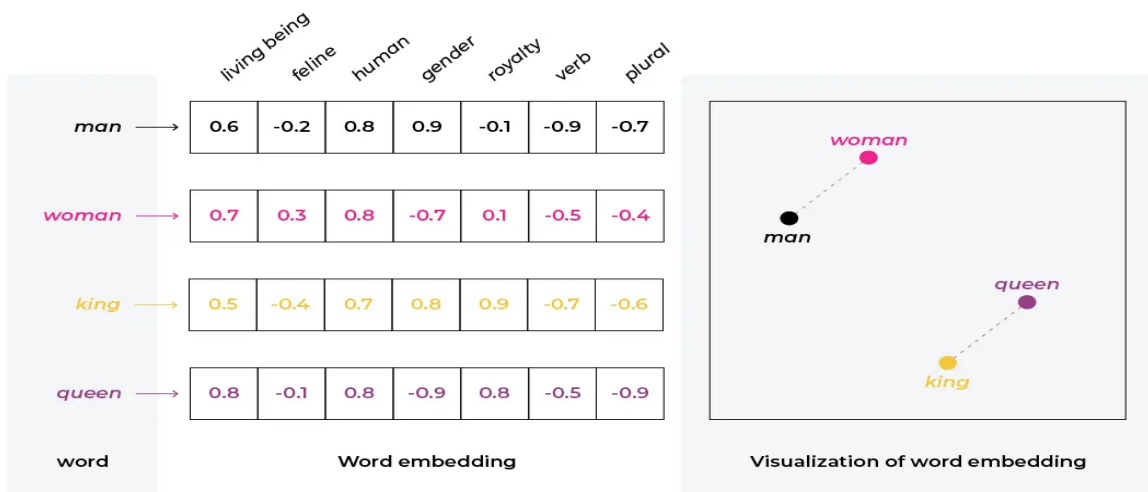
Η πλατφόρμα επιτρέπει την ανάπτυξη κάποιων bots τα οποία συνδέονται αυτόματα σε servers και μπορούν να επικοινωνήσουν μέσω εντολών με τους χρήστες. Ανάλογα με την χρησιμοποιούμενη από τον χρήστη εντολή και το κείμενο που ακολουθεί το Discord Bot δρα, και προσπαθεί να φέρει εις πέρας το ζητούμενο. Η επικοινωνία γίνεται μέσω του Discord API επιτρέποντας τους να στέλνουν τις απαντήσεις και να αντιδρούν στις εντολές.

Επιπρόσθετα, το Discord χρησιμοποιεί την τεχνολογία webhooks για την επικοινωνία και τις ενημερώσεις σχετικά με οτιδήποτε αφορά την επικοινωνία στο discord. Τα webhooks είναι απλά HTTP endpoints συνδεδεμένα με το κάθε “channel” στα οποία γίνεται ένα POST request με τα δεδομένα επικοινωνίας κάθε φορά (μηνύματα, επισυναπτόμενα αρχεία) καθιστώντας έτσι την μόνιμη σύνδεση με το Bot μη απαραίτητη. Είναι σχεδιασμένα να ενημερώνουν και να στέλνουν ειδοποιήσεις αυτόματα με κάθε αλλαγή, μειώνοντας έτσι το overhead της επικοινωνίας.

3.13 Embeddings

Τα embeddings αποτελούν μία από τις σημαντικότερες δομές στην επεξεργασία φυσικής γλώσσας. Embeddings ονομάζονται οι αριθμητικές αναπαραστάσεις δεδομένων -όπως λέξεις, προτάσεις, εικόνες και οποιασδήποτε μορφής δεδομένων- σε διανύσματα. Αυτά τα διανύσματα, έχουν την δυνατότητα κατανόησης τόσο της σημασίας, του συντακτικού αλλά και της δομής των δεδομένων με αριθμητικό τρόπο, δίνοντας έτσι την δυνατότητα σε αλγορίθμους και μοντέλα, να τα κατανοήσουν πλήρως και να τα χρησιμοποιήσουν για εκπαίδευση και πρόβλεψη με πολύ αποδοτικό τρόπο.

Τα embeddings εμφανίζονται σε όλες τις εφαρμογές που αφορούν την επεξεργασία φυσικής γλώσσας διότι λόγω της φύσης τους, κρατούν την πληροφορία και τις συσχετίσεις μεταξύ των οντοτήτων που εμφανίζονται στο κείμενο, μετατρέποντας το κείμενο σε μία μορφή εύκολα αξιοποιήσιμη από τα μοντέλα μηχανικής μάθησης διατηρώντας μάλιστα ολόκληρη την πληροφορία. Επιπρόσθετα, τα embeddings δίνουν την δυνατότητα να πραγματοποιηθούν και απλές αριθμητικές πράξεις σε δεδομένα κειμένου, π.χ. Διάνυσμα(Βασιλιάς) - Διάνυσμα(Αντρας) + Διάνυσμα(Γυναίκα) = Διάνυσμα(Βασίλισσα).



Σχήμα 11: Παράδειγμα Embeddings

3.14 Web Scraping και Beautiful Soup

Το Web Scraping (ιστοσυγκομιδή) αποτελεί μία διαδικασία αυτόματης εξαγωγής πληροφορίας από ιστοσελίδες. Επιτρέπει στο χρήστη την συλλογή μεγάλου όγκου δεδομένων από διαδικτυακές πηγές γρήγορα και αποδοτικά, οι οποίες με διαφορετική μέθοδο θα απαιτούσαν αντιγραφή και επικόλληση χειροκίνητα για μεγάλο χρονικό διάστημα. Η μέθοδος του web scraping είναι κοινώς χρησιμοποιούμενη σε κλάδους που αφορούν την άντληση ή ανάλυση δεδομένων.

Η διαδικασία του Web Scraping περιλαμβάνει την αποστολή ενός HTTP request σε μία ιστοσελίδα, την λήψη των περιεχομένων της σελίδας και την εξαγωγή των χρήσιμων μερών και της ζητούμενης πληροφορίας. Συχνά οι scrapers- τα προγράμματα που υλοποιούν αυτή τη διαδικασία- προσομοιώνουν την συμπεριφορά ενός πραγματικού χρήστη, στέλνοντας πολλαπλά HTTP requests, και πραγματοποιώντας μία κανονική διαδρομή μέσα στην δομή της ιστοσελίδας, παρόμοια με αυτήν που θα έκανε ένας πραγματικός χρήστης.

Ένα από τα πιο δημοφιλή εργαλεία υλοποίησης web scraping αποτελεί η βιβλιοθήκη Beautiful Soup της rython. Είναι ένα πολύ ισχυρό εργαλείο που επιτρέπει την εξαγωγή πληροφορίας και την πλοήγηση σε HTML (HyperText Markup Language) και XML (eXtensive Markup language) έγγραφα. Δημιουργεί μία δεντρική δομή δεδομένων, η οποία απλοποιεί την διαδικασία πλοήγησης και εξαγωγής δεδομένων χρησιμοποιώντας τα tags και τα attributes (χαρακτηριστικά) τους μέσα στο έγγραφο. Η χρήση της Beautiful Soup βιβλιοθήκης καθιστά το web scraping μία αρκετά απλή διαδικασία, η οποία απαιτεί της βασικές γνώσεις και αρχές που διέπουν τις ιστοσελίδες.

Η βιβλιοθήκη Beautiful Soup, επιπρόσθετα, μετατρέπει την αγνή HTML της ιστοσελίδας σε μία διεπαφή όπου μέσω της rython είναι εύκολη η διαχείριση της. Διαθέτει μία μεγάλη γκάμα στον τρόπο αναζήτησης στην ιστοσελίδα και την εξαγωγή της πληροφορίας, τόσο μέσω των tag names (ονόματα των στοιχείων), των κλάσεων και των ids, των CSS selectors, όσο και μέσω κανονικών εκφράσεων (regular expressions) όπου μπορεί ο χρήστης να ορίσει κάποιο μοτίβο το οποίο διέπει τα δεδομένα στόχο για τα οποία θέλει να εξάγει πληροφορία. Επιπλέον, η βιβλιοθήκη αποτελεί ένα πολύ ισχυρό εργαλείο καθώς μπορεί να διαχειριστή προβληματική ή ελλιπή HTML, ένα πολύ κοινό φαινόμενο στις ιστοσελίδες όπου υπάρχουν λάθη στην HTML λόγω της χρήσης των frameworks ή των πλατφορμών CMS (Content Management Systems, Συστήματα Διαχείρισης Περιεχομένου) τα οποία προσφέρουν λύσεις για ευκολότερη και γρηγορότερη ανάπτυξη και διανομή ιστοσελίδων, χωρίς απαραίτητα η ιστοσελίδα να είναι απολύτως σωστά δομημένη, καθιστώντας την Beautiful Soup ένα πολύ σημαντικό και στιβαρό εργαλείο για την υλοποίηση διαδικασιών Web Scraping.

Συνοπτικά, η ιστοσυγκομιδή (Web Scraping) αποτελεί μία πολύ σημαντική διαδικασία στην ανάπτυξη σύγχρονων εφαρμογών που απαιτούν την άντληση δεδομένων από ιστοσελίδες, δεδομένων οποιασδήποτε μορφής, και η βιβλιοθήκη Beautiful Soup αποτελεί ένα από τα χρησιμότερα εργαλεία υλοποίησης web scraping, διαδραματίζοντας έναν καίριο ρόλο στην γρήγορη, εύκολη και αποτελεσματική ανάκτηση δεδομένων από ιστοσελίδες.

3.15 GitHub

Το GitHub αποτελεί μία πλατφόρμα που χρησιμοποιεί το Git σαν βάση για τον κατανεμημένο έλεγχο εκδόσεων (Distributed Version Control) και επιπρόσθετα επιτρέπει τον έλεγχο πρόσβασης, τη συνεχή ανάπτυξη, τα αιτήματα για ανάπτυξη λειτουργίας, την παρακολούθηση σφαλμάτων, τον διαμοιρασμό των εργασιών και την εμφάνιση πληροφοριών για εφαρμογές και προγράμματα.

Το GitHub είναι ευρέως χρησιμοποιούμενο από προγραμματιστές και εταιρείες στην ανάπτυξη προγραμμάτων αλλά και στον διαμοιρασμό εφαρμογών ανοιχτού κώδικα. Διευκολύνει την συνεργατικότητα και επιτρέπει μέσω του ελέγχου εκδόσεων την αρμονική ανάπτυξη εφαρμογών με την δυνατότητα επιστροφής σε προηγούμενη έκδοση (rollback) σε περίπτωση σφαλμάτων, καθώς και την

ανάπτυξη σε διαφορετικό κλάδο (branch) ώστε να μην επηρεαστεί η εφαρμογή που τρέχει από τις ενημερώσεις κατά την διάρκεια της ανάπτυξης.

Κεφάλαιο 4ο: Αρχιτεκτονική της Εφαρμογής και Υλοποίηση

4.1 Αρχιτεκτονική της Εφαρμογής

Η εφαρμογή που υλοποιήσα αποτελείται από 5 μέρη:

1. Την διεπαφή χρήστη, ένα bot στην εφαρμογή discord
2. Το flask backend, μία υποδομή ώστε μέσω http request, η εφαρμογή να δέχεται είσοδο και να επικοινωνεί με την διεπαφή.
3. Το κομμάτι της μηχανικής μάθησης, όπου υπάρχουν οι υπηρεσίες για το μεγάλο γλωσσικό μοντέλο, την αναγνώριση γλώσσας, την μετάφραση και την διανυσματοποίηση.
4. Τη διανυσματική βάση δεδομένων.
5. Τις υπηρεσίες διαθέσιμες για function calling από το μεγάλο γλωσσικό μοντέλο.

4.2 Διεπαφή Χρήστη - Discord

Η διεπαφή χρήστη της εφαρμογής υλοποιείται μέσω ενός Discord bot, το οποίο έχει αναπτυχθεί με στόχο την αλληλεπίδραση χρηστών σε ένα περιβάλλον άμεσα αναγνωρίσιμο και ευχάριστο στη χρήση. Το bot αυτό είναι συνδεδεμένο σε έναν Discord server και εκτελεί εντολές που υποβάλλονται από τους χρήστες μέσω της πλατφόρμας του Discord, αξιοποιώντας τα Discord commands και τη σχετική λειτουργικότητα της πλατφόρμας.

Για την ανάπτυξη του χρησιμοποιήθηκε η βιβλιοθήκη discord.py της Python, η οποία παρέχει ένα πλήρες σύνολο εργαλείων για τη δημιουργία και διαχείριση Discord bots. Η εγγραφή του bot πραγματοποιήθηκε μέσω του Discord Developer Portal, όπου δημιουργήθηκε μια νέα εφαρμογή και αποδόθηκε ένα μοναδικό authentication token, το οποίο χρησιμοποιείται για την ασφαλή σύνδεση του bot με τον επιθυμητό server μέσω του προσωπικού μου λογαριασμού στο Discord. Ο μηχανισμός πιστοποίησης αυτός είναι ουσιώδης για τη διασφάλιση ότι μόνο το εξουσιοδοτημένο bot μπορεί να αλληλεπιδρά με τον server.

Η διαχείριση της επικοινωνίας μεταξύ χρηστών και συστήματος γίνεται μέσω της τεχνολογίας των webhooks. Το bot παρακολουθεί συγκεκριμένα κανάλια του server, αναμένοντας την εισαγωγή εντολών με προκαθορισμένη δομή. Μόλις εντοπιστεί μια έγκυρη εντολή, ενεργοποιείται ένας μηχανισμός ειδοποίησης που μεταφέρει το περιεχόμενο της εντολής στη μονάδα discord_service. Η μονάδα αυτή έχει σχεδιαστεί να διαχειρίζεται το routing των αιτημάτων, λειτουργώντας ως ενδιάμεσος ανάμεσα στην πλατφόρμα Discord και το backend της εφαρμογής.

Η αποστολή του αιτήματος στο backend πραγματοποιείται μέσω HTTP requests, ακολουθώντας το πρότυπο RESTful APIs. Συγκεκριμένα, το κείμενο που έχει υποβάλει ο χρήστης προωθείται προς το backend που έχει αναπτυχθεί με Flask, όπου υφίσταται την ανάλογη επεξεργασία. Το backend αναλαμβάνει την επεξεργασία της πληροφορίας, την εξαγωγή συμπερασμάτων ή την παραγωγή της επιθυμητής απάντησης, και επιστρέφει το αποτέλεσμα στο discord_service. Στη συνέχεια, η απάντηση μορφοποιείται κατάλληλα, ώστε να ανταποκρίνεται στα πρότυπα παρουσίασης της πλατφόρμας Discord και αποστέλλεται πίσω στον χρήστη μέσω του Discord API, ολοκληρώνοντας έτσι τον κύκλο αλληλεπίδρασης.

Η επιλογή της πλατφόρμας Discord ως κύριο μέσο αλληλεπίδρασης των χρηστών με την εφαρμογή δεν έγινε τυχαία. Το Discord παρουσιάζει υψηλή δημοτικότητα ιδιαίτερα στις νεανικές και ακαδημαϊκές κοινότητες, με πλήθος φοιτητών να χρησιμοποιούν ήδη την πλατφόρμα για καθημερινή επικοινωνία, ανταλλαγή γνώσεων και συνεργασία σε ομαδικές εργασίες. Η ύπαρξη ενός δραστήριου server της σχολής ενίσχυσε περαιτέρω την απόφαση, καθώς εξασφαλίζει άμεση πρόσβαση σε ένα κοινό με εξοικείωση στην πλατφόρμα και ελαχιστοποιεί τα εμπόδια υιοθέτησης της εφαρμογής.

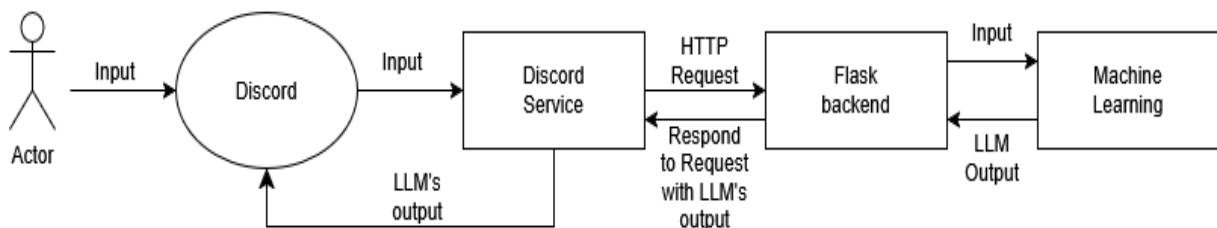
Σημαντικό πλεονέκτημα του Discord είναι ότι δεν απαιτείται επιπλέον εγκατάσταση λογισμικού από τους χρήστες, κάτι που αυξάνει σημαντικά το ποσοστό των φοιτητών οι οποίοι θα κάνουν χρήση της εφαρμογής. Η χρήση γνωστής τεχνολογίας καθιστά την εμπειρία του χρήστη ευκολότερη και ενισχύει την αίσθηση αξιοπιστίας της εφαρμογής.

Παράλληλα, από τεχνικής άποψης, το Discord επιτρέπει την απευθείας διασύνδεση με το backend χωρίς την ανάγκη ύπαρξης domain name ή στατικής δημόσιας IP διεύθυνσης για τον server. Το Discord διαχειρίζεται εσωτερικά τη σύνδεση μεταξύ του bot και της υποδομής του server, καθιστώντας το σύστημα πιο ασφαλές και λιγότερο εκτεθειμένο σε εξωτερικές επιθέσεις. Το γεγονός ότι το bot λειτουργεί σε έναν κλειστό, ελεγχόμενο server, στον οποίο πρόσβαση έχουν μόνο εξουσιοδοτημένοι χρήστες, προσφέρει ένα επιπλέον επίπεδο ασφάλειας και περιορίζει τους κινδύνους από μη εξουσιοδοτημένη χρήση ή κακόβουλες ενέργειες.

Επιπλέον, η πλατφόρμα Discord υποστηρίζει μηχανισμούς διαχείρισης δικαιωμάτων (permissions) σε επίπεδο bot και server, παρέχοντας τη δυνατότητα ελέγχου των ενεργειών που μπορεί να εκτελέσει το bot, καθώς και τον περιορισμό της πρόσβασής του μόνο στα απαιτούμενα κανάλια και λειτουργίες. Αυτό επιτρέπει την αυστηρή τήρηση των αρχών της ελαχιστοποίησης δικαιωμάτων (principle of least privilege) που αποτελεί βέλτιστη πρακτική στην ανάπτυξη ασφαλών συστημάτων.

Τέλος, αξίζει να σημειωθεί ότι η διεπαφή χρήστη (UI) αποτελεί ένα από τα πλέον κρίσιμα στοιχεία στην επιτυχία οποιασδήποτε εφαρμογής. Ανεξάρτητα από την ποιότητα του backend ή τη λειτουργικότητα του συστήματος, μια δύσχρηστη ή κακής ποιότητας διεπαφή μπορεί να οδηγήσει σε χαμηλά ποσοστά χρήσης και τελικά στην αποτυχία της εφαρμογής. Η χρήση του Discord ως διεπαφή προσφέρει ένα περιβάλλον ανεπτυγμένο και συνεχώς βελτιούμενο από μια εξειδικευμένη ομάδα ανάπτυξης, με έμφαση στην ευχρηστία, την απόδοση και την αισθητική. Επιπλέον, η γνώση του τρόπου λειτουργίας του Discord μεταξύ των χρηστών μειώνει δραστικά την ανάγκη για εκπαίδευση ή αναλυτικές οδηγίες χρήσης, συμβάλλοντας στην άμεση και ομαλή ενσωμάτωση της εφαρμογής στην καθημερινότητά τους.

Συνολικά, η επιλογή και υλοποίηση της διεπαφής χρήστη μέσω Discord συνδυάζει ευκολία για τον χρήστη, ασφάλεια, προσβασιμότητα και εξασφάλιση της ποιότητας της διεπαφής, συμβάλλοντας καθοριστικά στην επιτυχία της εφαρμογής.



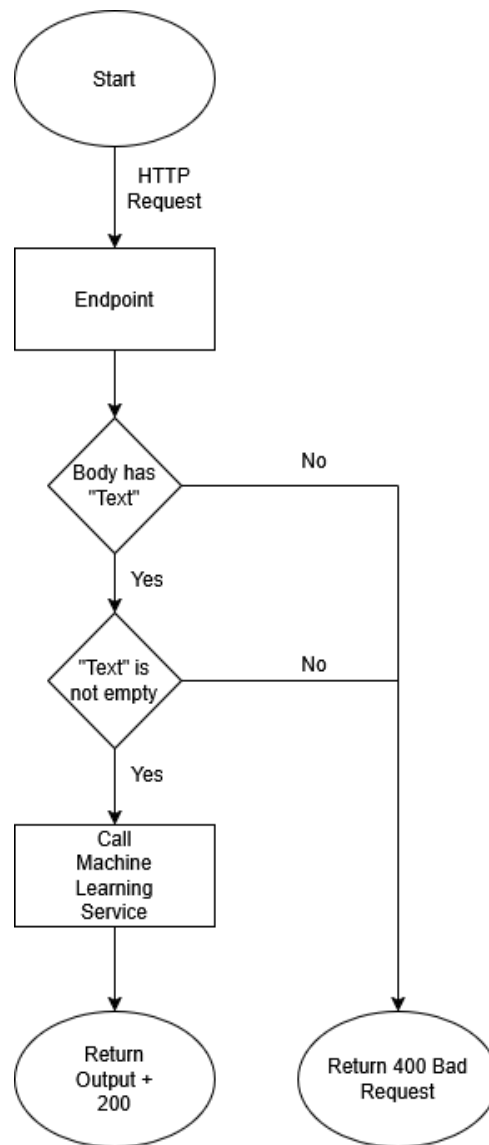
Σχήμα 12: Αρχιτεκτονική Υλοποίησης διασύνδεσης διεπαφής χρήστη με το backend

4.3 Flask Backend

Το framework Flask αποτελεί έναν από τους απλούστερους τρόπους υλοποίησης μίας υποδομής backend με την γλώσσα python. Χρησιμοποιήθηκε με σκοπό την δημιουργία ενός API (application programming interface) με δύο endpoints “process_query” και “rag” με σκοπό την διασύνδεση της διεπαφής χρήστη με το κομμάτι της μηχανικής μάθησης και την κύρια λογική της εφαρμογής. Τα δύο αυτά διαφορετικά endpoints καλούνται ανάλογα με την εντολή (command) που χρησιμοποίησε ο χρήστης. Το endpoint “process_query” αποτελεί αυτό της βασικής λειτουργικότητας της εφαρμογής όπου το μοντέλο βάσει του δοθέν input καλεί την συνάρτηση η οποία επιλύει το πρόβλημα και με τις απαραίτητες τιμές στις παραμέτρους. Το endpoint “rag” είναι υπεύθυνο για την κλήση και χρήση της μεθόδου RAG του μοντέλου, η οποία χρησιμοποιεί ως βάση δεδομένων, τα δεδομένα της ιστοσελίδας

της σχολής. Η δομή του API είναι πολύ απλή αποτελούμενη μόνο από δύο endpoints. Αναλυτικά η δομή των δύο endpoints είναι η εξής:

- **Process_Query:** Το endpoint περιμένει ένα POST HTTP request όπου στο body υπάρχει ένα json με ένα μόλις πεδίο "text" το οποίο έχει και το κείμενο εισόδου το οποίο αποστέλλει ο χρήστης. Γίνεται έλεγχος, στις περιπτώσεις όπου το πεδίο δεν υπάρχει ή είναι κενό το backend απαντάει ότι υπάρχει πρόβλημα με το request και επιστρέφει κωδικό 400: Bad Request. Στην περίπτωση όπου το text πεδίο υπάρχει και περιέχει περιεχόμενο, καλεί το service ask_model και επιστρέφει το αποτέλεσμα του με κωδικό 200: OK.
- **RAG:** Το endpoint περιμένει ένα POST HTTP request όπου στο body υπάρχει ένα json με ένα μόλις πεδίο "text" το οποίο είναι το κείμενο εισόδου που αποστέλλει ο χρήστης. Όπως και στο process_query endpoint γίνεται έλεγχος για την ύπαρξη του πεδίου και του περιεχομένου του, σε περίπτωση όπου κάτι από τα δύο είναι κενό επιστρέφεται αντίστοιχο μήνυμα και κωδικός 400: Bad Request. Διαφορετικά το κείμενο εισόδου στέλνεται στο rag_inference service και αποστέλλεται πίσω η απάντηση του μοντέλου με κωδικό 200: OK.



Σχήμα 13: Διάγραμμα Ροής των Endpoints

Επιλέχθηκε η μη χρησιμοποίηση κάποιας αυθεντικοποίησης, μέσω api-key ή jwt διότι στην υλοποίηση το flask backend είναι ορατό μόνο σε ένα εσωτερικό δίκτυο επομένως δεν υπάρχει εξωτερική πρόσβαση αυτού σε κανέναν πέρα από το discord service.

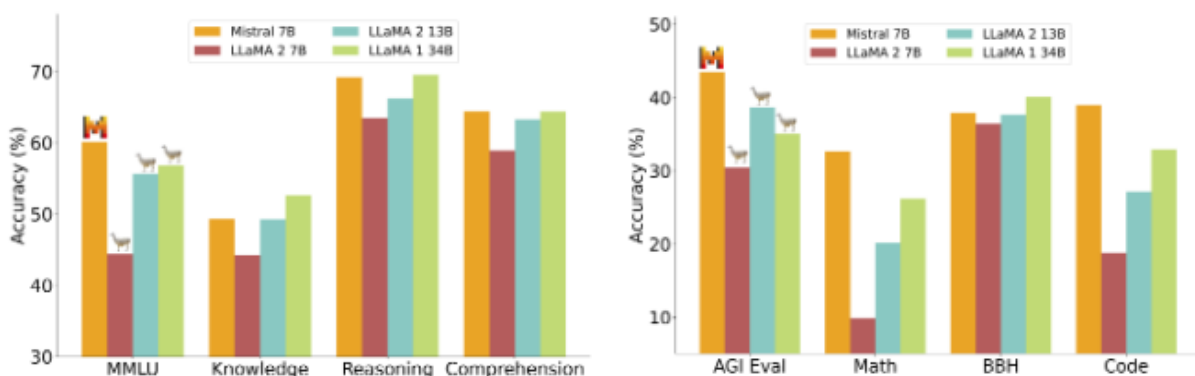
Το flask διατηρεί την δυνατότητα χρήσης μέσω ενός built-in server, ωστόσο δεν διαθέτει σημαντικές ιδιότητες απαραίτητες για την χρήση του API σε περιβάλλον παραγωγής. Έτσι επιλεχθηκε το Gunicorn μέσω του οποίου υπάρχει η δυνατότητα χρήσης πολλών workers για την διεκπεραίωση πολλαπλών ταυτόχρονων αιτημάτων, η εξατομίκευση στην πορτα (port) και την διεύθυνση στην οποία θα είναι ορατό το API καθώς και στην δυνατότητα σύνδεσής του με έναν load balancer (όπως ο Nginx) στο μέλλον ώστε η εφαρμογή να μπορεί να χρησιμοποιηθεί από διαφορετικό UI χωρίς πρόβλημα, είτε web ui, είτε android ή ios εφαρμογή.

Η δημιουργία ενός API για την χρήση της εφαρμογής, επιπρόσθετα από την υλοποίηση του UI με το discord επιλέχθηκε διότι η απευθείας κλήση των services της μηχανικής μάθησης από την διεπαφή χρήστη θα αποτελούσε παραβίαση των αρχών της καθαρής αρχιτεκτονικής και συγκεκριμένα του διαχωρισμού των αρμοδιοτήτων, καθώς και για την δυνατότητα επέκτασης της εφαρμογής στο μέλλον με κάποιο καινούργιο ή διαφορετικό UI (User Interface, διεπαφή χρήστη) χωρίς να υπάρχει η ανάγκη προσθήκης ή αλλαγής σε οποιοδήποτε από τα ήδη υπάρχοντα μέρη της εφαρμογής.

4.4 Υπηρεσίες Μηχανικής Μάθησης

Το κύριο μέρος της εφαρμογής όπου βρίσκεται όλη η λογική που επιτρέπει την χρήση της εφαρμογής είναι το μέρος της μηχανικής μάθησης. Σε αυτό το μέρος της εφαρμογής υλοποιείται η λειτουργικότητα της αρχιτεκτονικής RAG (Retrieval Augmented Generation), το function calling (κλήση συνάρτησης) καθώς και όλη η απαιτούμενη προ-επεξεργασία των δεδομένων εισόδου προκειμένου να έρθουν αυτά στην επιθυμητή μορφή για τροφοδότηση στα μοντέλα.

Τα μοντέλα που χρησιμοποιήθηκαν σε αυτό το τμήμα της εφαρμογής είναι τα Mistral-7B-Instruct-v0.3[18], ένα μεγάλο γλωσσικό μοντέλο το οποίο αποτελείται από 7 δισεκατομμύρια παραμέτρους και παρά το μικρό, συγκριτικά, μέγεθος του ξεπερνούσε σε επιδόσεις όλα τα μέχρι τότε χρησιμοποιούμενα «μικρά» μοντέλα τα οποία διέθεταν σχεδόν τις διπλάσιες παραμέτρους, όπως το Llama 2 13B και Vicuna 13B.

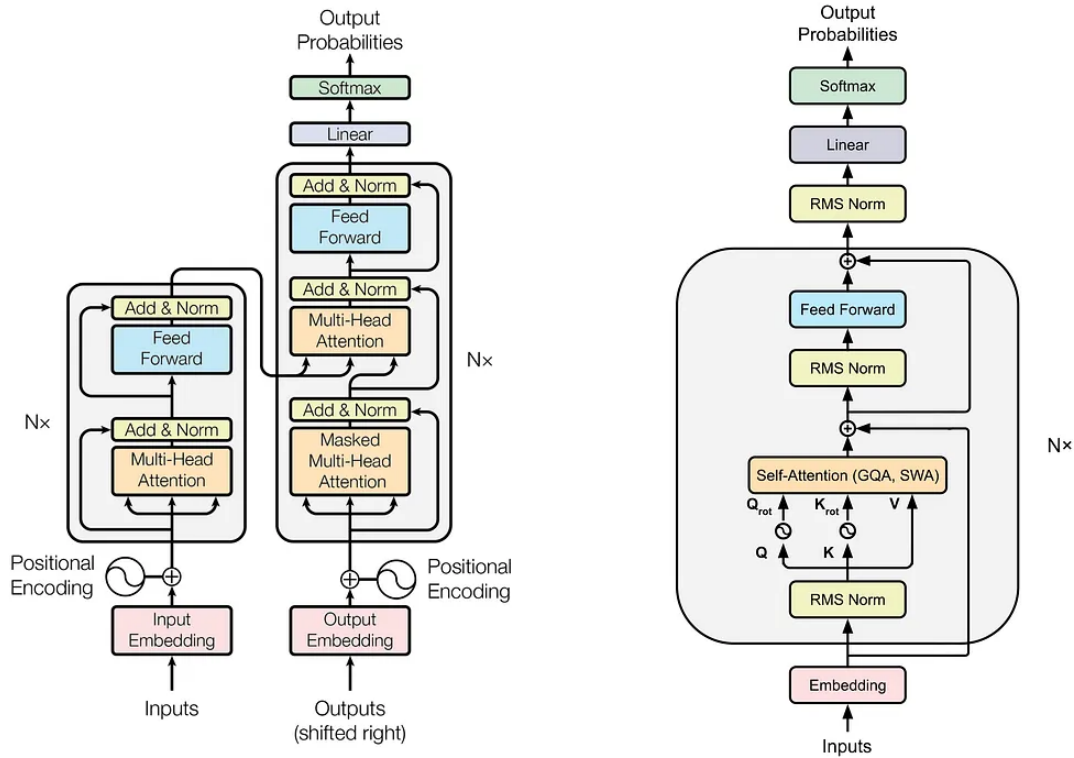


Σχήμα 14: Επιδόσεις του Mistral 7B μοντέλου

Transformer

Mistral 7B

medium.com/@bradneysmith



Σχήμα 15: Αρχιτεκτονική Mistral 7B σε αντιπαράθεση με transformer

Το Meltemi 7B Instruct v1.5 [19] μία επανεκπαιδευμένη (fine-tuned) εκδοχή του Mistral 7B, δημιουργημένο από το Ινστιτούτο Επεξεργασίας Λόγου. Το ινστιτούτο είχε κυκλοφορήσει και μία αρχική εκδοχή Meltemi 7B v1 της οποίας η v1.5 αποτελεί αναβάθμιση. Η επανεκπαίδευση του μοντέλου έγινε με την χρήση μίας τεράστιας βάσης δεδομένων η οποία αποτελείται κατα προσέγγιση από 55 δισεκατομμύρια tokens τα οποία μοιράζονται ως: 43.3 δισεκατομμύρια tokens μόνο για την ελληνική γλώσσα από δημόσιες πηγές. 10.5 δισεκατομμύρια tokens μόνο για την Αγγλική γλώσσα και 600 εκατομμύρια tokens ελληνικών και αγγλικών δεδομένων παράλληλα, για να διατηρήσουν την ικανότητα του μοντέλου να απαντάει στην Αγγλική γλώσσα.

Γλώσσα Δεδομένων	Αριθμός Tokens	Ποσοστό
Ελληνικά	43.383.244.502	79.5%
Αγγλικά	10.538.413.259	19.3%
Ελληνικά-Αγγλικά	633.816.023	1.2%
Σύνολο	54.555.473.784	100%

Πίνακας 1: Δεδομένα Επανεκπαίδευσης μοντέλου Meltemi

Το Meltemi αποτελεί μία πολύ καλή λύση για τις εφαρμογές που αφορούν την ελληνική γλώσσα, καθώς αποδεικνύει σημαντική βελτίωση σχετικά με το αρχικό μοντέλο με τα αποτελέσματα σε όλες τις διαθέσιμες μετρικές σύγκρισης για την ελληνική γλώσσα να παρουσιάζουν μία αύξηση 16.3% κατα μέσο όρο. Αναλυτικότερα:

Μετρική	Mistral 7B	Meltemi 7B v1	Meltemi 7B v1.5
Medical MCQA EL (15-shot)	29.8%	46.3%	48.1%
Belebele EL (5-shot)	45.0%	68.5%	68.6%
HellaSwag EL (10-shot)	36.5%	63.3%	65.7%
ARC-Challenge EL (25-shot)	27.1%	43.6%	47.1%
TruthfulQA MC2 EL (0-shot)	45.8%	44.6%	45.1%
MMLU EL (5-shot)	35%	42.4%	42.4%
Average	36.5%	51.4%	52.8%

Πίνακας 2: Επιδόσεις μοντέλου Meltemi

Για την καλύτερη κατανόηση των επιδόσεων των μοντέλων είναι σημαντικό να αναλύσουμε το πεδίο εξέτασης το οποίο εξετάζει κάθε μετρική.

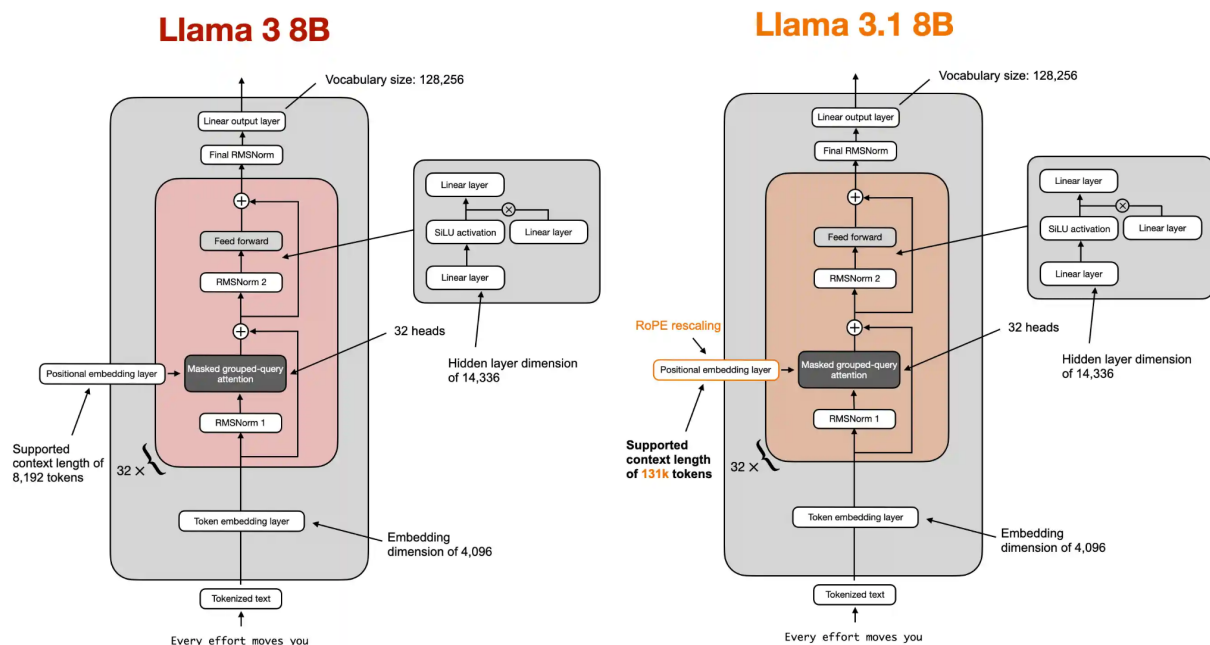
- Medical MCQA EL (15-shot):** Η μετρική αυτή εξετάζει την ικανότητα του μοντέλου να απαντά σωστά σε ερωτήσεις πολλαπλής επιλογής (MCQA, Multiple Choice Question Answering) πάνω στο γνωστικό αντικείμενο της ιατρικής επιστήμης [20], ενώ το EL σηματοδοτεί ότι η μετρική αφορά την ελληνική γλώσσα. Το μέρος (15-shot) σηματοδοτεί πως κατά την εισαγωγή των δεδομένων στο μοντέλο για τη μέτρηση, τροφοδοτούνται στο μοντέλο 15 προηγούμενα παραδείγματα, βοηθώντας το μοντέλο να κατανοήσει την μορφή της απάντησης την οποία πρέπει να δώσει.
- Belebele EL (5-shot):** Η μετρική belebele δημιουργήθηκε για να ελέγχει τις επιδόσεις των μεγάλων γλωσσικών μοντέλων στην κατανόηση του κειμένου κάνοντας χρήση ερωτήσεων πολλαπλών επιλογών [21]. Η μετρική διαθέτει έλεγχο για πολλές γλώσσες, το EL υποδηλώνει την ελληνική γλώσσα, και το 5-shot ότι δόθηκαν 5 παραδείγματα με την απάντηση τους μαζί με το ζητούμενο στο μοντέλο.
- HellaSwag EL (10-shot):** Το HellaSwag αποτελεί μία μετρική η οποία ελέγχει την ικανότητα των μεγάλων γλωσσικών μοντέλων στη διαδικασία συμπλήρωσης προτάσεων όπου δίνεται μία πρόταση και ζητείται η λογική συνέχεια, με το μοντέλο να πρέπει να επιλέξει την σωστή επιλογή από τις δοθείσες επιλογές, οι οποίες έχουν την μορφή ερώτησης πολλαπλών επιλογών[22]. Οι επιλογές EL και (10-shot), όπως και προηγουμένως, υποδηλώνουν ότι η μετρική αφορά ερωτήσεις στην ελληνική γλώσσα, και ότι δίνονται 10 παραδείγματα μαζί με το input στο μεγάλο γλωσσικό μοντέλο.
- ARC-Challenge EL (25-shot):** Η μετρική αυτή αποτελείται από ένα σύνολο επιστημονικών ερωτήσεων πολλαπλής επιλογής οι οποίες είχαν δημιουργηθεί με σκοπό τον έλεγχο των γνώσεων μαθητών για ηλικίες από 8 έως 13 ετών. Το ARC-Challenge ή αλλιώς AI2 Reasoning Challenge αποτελείται από 2 μέρη ένα με εύκολες ερωτήσεις και ένα με δυσκολότερες[23]. Τα

EL και (25-shot) υποδηλώνουν ότι χρησιμοποιείται μόνο το ελληνικό κομμάτι της μετρικής και δίνονται 25 παραδείγματα μαζί με το input σαν είσοδο στο μεγάλο γλωσσικό μοντέλο κατά την διάρκεια του ελέγχου.

- **TruthfulQA MC2 EL (0-shot):** Είναι μία μετρική σχεδιασμένη με σκοπό την αξιολόγηση της ικανότητας των μεγάλων γλωσσικών μοντέλων να παράγουν απαντήσεις με αληθή περιεχόμενο και όχι hallucinations (ψευδές ή λανθασμένο περιεχόμενο το οποίο παρουσιάζεται σαν πραγματικό) [24]. Το MC2 υποδηλώνει ότι οι ερωτήσεις έχουν την μορφή πολλών σωστών, δίνεται μία ερώτηση και πολλές σχετικές απαντήσεις με την μορφή Σωστό/Λάθος, ενώ τα EL και 0-shot υποδηλώνουν ότι οι ερωτήσεις είναι στα ελληνικά και δεν δίνεται παράδειγμα μαζί με το input στο μεγάλο γλωσσικό μοντέλο κατά την διάρκεια της αξιολόγησης.
- **MMLU EL (5-shot):** Η μετρική MMLU (Massive Multitask Language Understanding) αξιολογεί την ικανότητα των μεγάλων γλωσσικών μοντέλων σε 57 διαφορετικές απαιτήσεις, με αυτές να πηγάζουν από μαθηματικά δημοτικού έως εξετάσεις εισαγωγής ιατρικής σχολής [25]. Αυτή η μετρική ελέγχει την ικανότητα του μοντέλου να καταλαβαίνει το κείμενο, να κάνει λογικούς συλλογισμούς, και να αξιοποιεί υπάρχουσα γνώση μεταξύ διαφορετικών γνωστικών αντικειμένων. Τα EL και 5-shot υποδηλώνουν ότι η μετρική αφορά την ελληνική γλώσσα και ότι δόθηκαν 5 παραδείγματα μαζί με την είσοδο κατά την αξιολόγηση του μοντέλου.

Η ύπαρξη και η χρησιμοποίηση όλων αυτών των μετρικών είναι πολύ χρήσιμη για την κατανόηση των επιδόσεων αυτών των μοντέλων. Ωστόσο, για την επιλογή μεγάλου γλωσσικού μοντέλου σε μία εφαρμογή όπως ένα chatbot ή ένας έξυπνος ψηφιακός βοηθός, δεν μπορεί να αποδίδεται η αξία των μετρικών ισάριθμα. Ένα σύστημα πληροφόρησης χρηστών απαιτεί την αποστολή μόνο πραγματικών, αληθινών πληροφοριών με τα hallucinations να αποτελούν σοβαρό πρόβλημα, έτσι η μετρική TruthfulQA MC2 EL έχει μεγαλύτερη βαρύτητα σε εφαρμογές και συστήματα αυτής της μορφής, τα οποία έχουν σκοπό την σωστή πληροφόρηση του χρήστη. Κατά συνέπεια, εάν έπρεπε να επιλεγεί ένα μεγάλο γλωσσικό μοντέλο για μία εφαρμογή ψηφιακού βοηθού μόνο βάσει αυτών των μετρικών, το Mistral 7B μοντέλο θα αποτελούσε μία πιο στιβαρή επιλογή, παρά τις χειρότερες αποδόσεις του συνολικά σε όλες τις μετρικές, χωρίς ωστόσο η επιλογή του Meltemi 7B να αποτελεί μία αντικειμενικά λανθασμένη επιλογή, καθώς η διαφορά στην επίδοση δεν είναι ιδιαίτερα μεγάλη.

Τα δύο άλλα μεγάλα γλωσσικά μοντέλα τα οποία χρησιμοποιήθηκαν ήταν τα Llama 3.1 8B Instruct[26] και η finetuned στα ελληνικά έκδοση του από το Ινστιτούτο Επεξεργασίας Γλώσσας, Llama Krikri 8B Instruct.



Σχήμα 16: Αρχιτεκτονική Llama 3.1 μοντέλου

Το Llama 3.1 8B μοντέλο αποτελεί μέρος της οικογένειας των Meta Llama 3.1 μοντέλων τα οποία δημιούργησε η Meta το 2024 με τις επιδόσεις τους σε όλους του δείκτες να παρουσιάζουν σημαντική αύξηση, συγκριτικά με τα μοντέλα συγκρίνουμε μεγέθους τα οποία αποτελούσαν τις state-of-the-art επιλογές στον κλάδο. Πιο συγκεκριμένα:

Μετρική	Llama 3.1 8B	Gemma 2 9B IT	Mistral 7B Instruct
MMLU (0-shot)	73.0	72.3	60.5
MMLU PRO (5-shot)	48.3	-	36.9
IFEval	80.4	73.6	57.6
HumanEval (0-shot)	72.6	54.3	40.2
MBPP EvalPlus base (0-shot)	72.8	71.7	49.5
GSM8K (8-shot)	84.5	76.7	53.2
MATH (0-shot)	51.9	44.3	13.0
ARC Challenge (0-shot)	83.4	87.6	74.2
GPQA (0-shot)	32.8	-	28.8
BFCL	76.1	-	60.4
Nexus	38.5	30.0	24.7
ZeroSCROLLS/QuALITY	81.0	-	-
InfiniteBench / En. MC	65.1	-	-
NIH / Multi-needle	98.8	-	-
Multilingual MGSM (0-shot)	68.9	53.2	29.9

Πίνακας 3: Επιδόσεις Llama 3.1 8B Instruct

Είναι σημαντικό για την καλύτερη κατανόηση των επιδόσεων του Llama 3.1 8B μοντέλου να εξετάσουμε το πεδίο το οποίο αξιολογούν οι μετρικές οι οποίες δεν αναφέρθηκαννωρίτερα.

- **IFEval:** Η μετρική IFEval [27] εξετάζει την ικανότητα των μεγάλων γλωσσικών μοντέλων να ακολουθούν οδηγίες δοσμένες σε φυσική γλώσσα. Στόχος της μετρικής είναι η αξιολόγηση των μοντέλων όσον αφορά την κατανόηση φυσικής γλώσσας καθώς και της αξιοπιστίας τους στην τήρηση κανόνων και την εναρμόνιση με τις οδηγίες με έναν αξιόπιστο και εύκολα αναπαραγώμενο τρόπο.
- **HumanEval:** Τα μεγάλα γλωσσικά μοντέλα χρησιμοποιούνται συχνά για την συγγραφή και διόρθωση κώδικα. Η μετρική HumanEval [28] αποτελεί έναν τρόπο για την αξιολόγηση των

μεγάλων γλωσσικών μοντέλων στην ικανότητα τους να κατανοούν, να διορθώνουν και να παράγουν σωστό κώδικα από docstrings (κυριολεκτικές συμβολοσειρές, χρησιμοποιούμενες στον προγραμματισμό για να διαλευκάνουν την δομή και την λειτουργία τμημάτων κώδικα).

- **MBPP EvalPlus base:** Η μετρική MBPP EvalPlus [29] αποτελεί ένα υπερσύνολο της HumanEval μετρικής με περίπου 80 φορές περισσότερα test cases. Είναι και αυτό ένα κριτήριο αξιολόγησης τις ικανότητας των μεγάλων γλωσσικών μοντέλων να παράγουν σωστό κώδικα. Ενδιαφέρον παρουσιάζει ότι για την δημιουργία των δεδομένων αξιολόγησης των μεγάλων γλωσσικών μοντέλων χρησιμοποιήθηκαν μεγάλα γλωσσικά μοντέλα καθώς και στρατηγικές βασισμένες σε μεταλλάξεις (mutation-based strategies) όπου γίνονταν στοχευμένα ή/και τυχαίες αλλαγές σε ήδη υπάρχουσες εγγραφές του dataset με σκοπό τον εμπλουτισμό του.
- **GSM8K:** Τα μεγάλα γλωσσικά μοντέλα έχουν αποδείξει ότι μπορούν να ανταπεξέλθουν σε μία πληθώρα από διαφορετικές εργασίες, ωστόσο δυσκολεύονται στην υλοποίηση μαθηματικών πράξεων οι οποίες απαιτούν λογικά βήματα για να επιλυθούν. Η GSM8K[30] αποτελεί μία μετρική αξιολόγησης των μεγάλων γλωσσικών μοντέλων στην ικανότητα τους να επιλύουν μαθηματικά προβλήματα τα οποία απαιτούν από 2 έως 8 βήματα για να επιλυθούν. Τα προβλήματα είναι επιπέδου δημοτικού σχολείου με τα βήματα να είναι μία από τις 4 θεμελιώδεις αριθμητικές πράξεις (πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση). Τα δεδομένα αξιολόγησης είναι 8.5 χιλιάδες διαφορετικά προβλήματα.
- **MATH:** Η μετρική MATH [31], όπως υποδηλώνει και το όνομα της, εξετάζει και αξιολογεί την ικανότητα των μεγάλων γλωσσικών μοντέλων στην επίλυση μαθηματικών προβλημάτων. Αποτελείται από 12.500 απαιτητικά μαθηματικά προβλήματα τα οποία ακολουθούνται και από μία ενδελεχή επεξήγηση των λύσεων τους, βήμα προς βήμα.
- **GPQA:** Η μετρική GPQA (Google-Proof Question and Answer) [32] αξιολογεί την ικανότητα των μεγάλων γλωσσικών μοντέλων να απαντούν σε δύσκολες ερωτήσεις γνώσεων σε διαφορετικά πεδία όπως η βιολογία, η φυσική και η χημεία. Οι ερωτήσεις είναι υψηλής ποιότητας και εξαιρετικά δύσκολες, για να γίνει κατανοητή η δυσκολία των ερωτήσεων είναι σημαντικό να αναφερθεί ότι ειδική οι οποίοι έχουν ή φοιτούν για την απόκτηση διδακτορικού τίτλου σπουδών απάντησαν με ακρίβεια 65%, ενώ πολύ ικανοί μη ειδικοί αξιολογητές (highly skilled non-expert validators) απάντησαν με ακρίβεια μόλις 34%, έχοντας στην διάθεση τους 30 λεπτά με πρόσβαση στο διαδίκτυο χωρίς περιορισμούς.
- **BFCL:** Η μετρική BFCL (Berkeley Function-Calling Leaderboard) [33] αξιοποιεί την ικανότητα των μεγάλων γλωσσικών μοντέλων σχετικά με την χρησιμοποίηση των διαθέσιμων εργαλείων και την σωστή κλήση συναρτήσεων (function calling). Αποτελείται από τρεις διαφορετικές κατηγορίες κλήσεων συναρτήσεων τις πολλαπλές κλήσεις συναρτήσεων ενός γύρου (single-turn multiple function calling) όπου απαιτείται η κλήση μόνο μίας συνάρτησης για την επίλυση του προβλήματος, με διαφορετικές συναρτήσεις να είναι διαθέσιμες. Την κλήση συναρτήσεων πολλαπλών βημάτων (Multiple-Step Function Calling) όπου το μεγάλο γλωσσικό μοντέλο απαιτείται να καλέσει παραπάνω από μία συναρτήσεις συνεχόμενα με το αποτέλεσμα της πρώτης να χρησιμοποιείται σαν είσοδος για την επόμενη κ.ο.κ.. Τέλος, η κλήση συναρτήσεων πολλών γύρων (Multiple-Turn Function Calling) όπου το μεγάλο γλωσσικό μοντέλο πρέπει να καλέσει συναρτήσεις και μετά με βάση την απάντηση που πήρε, πρέπει να ζητήσει περισσότερες πληροφορίες ώστε να καλέσει διαφορετική συνάρτηση για να λύσει οριστικά το πρόβλημα.
- **Nexus:** Η μετρική Nexus [34] αποτελεί άλλο ένα κριτήριο αξιολόγησης της ικανότητας των μεγάλων γλωσσικών μοντέλων να αξιοποιούν τα διαθέσιμα σε αυτά εργαλεία και να καλούν συναρτήσεις. Χρησιμοποιούν 3 τύπους κλήσεων, τις απλές κλήσεις (simple calls) όπου το μοντέλο πρέπει να καλέσει μία συνάρτηση για να φέρει το πρόβλημα εις πέρας, τις εμφωλευμένες κλήσεις (nested calls) όπου το μεγάλο γλωσσικό μοντέλο οφείλει να καλέσει ταυτόχρονα παραπάνω από μία συναρτήσεις για να λύσει το πρόβλημα όπου τα αποτελέσματα της κλήσης μίας συνάρτησης μπορεί να χρησιμοποιηθούν ως είσοδος σε μία άλλη συνάρτηση,

και τέλος, οι παράλληλες κλήσεις όπου το μοντέλο με μία εντολή εισόδου οφείλει να καλέσει παραπάνω από μία συναρτήσεις οι οποίες όμως δεν συνδέονται με κάποιον τρόπο.

- **ZeroSCROLLS:** Η κατανόηση του κειμένου από τα μεγάλα γλωσσικά μοντέλα αποτελεί μία από τις σημαντικότερες απαιτήσεις για οποιαδήποτε εφαρμογή, διότι χωρίς την κατανόηση των ζητούμενων η παραγωγή σωστών απαντήσεων είναι αδύνατη. Η μετρική ZeroSCROLLS [35] αποτελεί έναν τρόπο αξιολόγησης της ικανότητας των μεγάλων γλωσσικών μοντέλων να κατανοούν φυσικό λόγο σε πολύ μεγάλα κείμενα. Το μοντέλο εξετάζει την ικανότητα κατανόησης των κειμένων ζητώντας από αυτά να κάνουν 5 διαφορετικές εργασίες σε 10 διαφορετικά datasets (σύνολα δεδομένων) όπως περίληψη, απάντηση ερωτήσεων πολλαπλής επιλογής κ.α. με τη χρήση σχεδόν 4.500 παραδειγμάτων.
- **InfiniteBench:** Η επεξεργασία και η δυνατότητα δημιουργίας λογικών συλλογισμών πάνω σε μεγάλα κείμενα αποτελεί μία σημαντική λειτουργία για τα μεγάλα γλωσσικά μοντέλα. Η μετρική InfiniteBench [36] αποτελεί ένα κριτήριο αξιολόγησης της ικανότητας των μεγάλων γλωσσικών μοντέλων να διαχειρίζονται σωστά πολύ μεγάλα κείμενα, τα οποία αποτελούνται από περισσότερα από εκατό χιλιάδες tokens, και δίνονται σαν context για την επίλυση προβλημάτων, καθώς και ως κείμενο πάνω στο οποίο απαιτείται να γίνει κατανόηση και επεξεργασία.
- **NIH/Multi-Needle:** Η μετρική NIH/Multi-Needle (Needle in Haystack) [37] αξιολογεί την ικανότητα των μεγάλων γλωσσικών μοντέλων να βρίσκουν τον στόχο ή το ζητούμενο το οποίο θα δώσει την απάντηση στο ερώτημα του χρήστη μέσα σε ένα πολύ μεγάλο κείμενο.
- **Multilingual MGSM:** Τα μεγάλα γλωσσικά μοντέλα λόγω των δεδομένων εκπαίδευσής τους, τα οποία αποτελούνται κατά κύριο λόγο από έγγραφα στην αγγλική γλώσσα τείνουν να συμπεριφέρονται καλύτερα σε ερωτήσεις στην αγγλική γλώσσα. Η μετρική Multilingual MGSM (Multilingual Grade School Math) [38] αξιολογεί την ικανότητα των μεγάλων γλωσσικών μοντέλων να λύνουν προβλήματα μαθηματικών τα οποία απαιτούν λογικούς συλλογισμούς σε 10 διαφορετικές γλώσσες. Πρόκειται για 250 εγγραφές δεδομένων από το σύνολο δεδομένων GSM8K τα οποία μεταφράστηκαν χειροκίνητα και εξετάζουν αν τα μεγάλα γλωσσικά μοντέλα διατηρούν την ικανότητα των λογικών συλλογισμών και σε γλώσσες πέρα των αγγλικών.

Σε μία εφαρμογή της οποίας η λειτουργικότητα βασίζεται στην ικανότητα των μεγάλων γλωσσικών μοντέλων να καλούν αποτελεσματικά συναρτήσεις οι οποίες επιλύουν τα προβλήματα του χρήστη οι μετρικές BFCL και Nexus αποτελούν το κύριο σημείο αξιολόγησης. Ενώ σε έναν έξυπνο ψηφιακό βοηθό οι απαιτήσεις παραγωγής κώδικα δεν είναι αξιοσημείωτες. Η αρχιτεκτονική RAG δίνει στα μεγάλα γλωσσικά μοντέλα γνώση και δεδομένα για να απαντήσουν ορθά στο ζητούμενο από τον χρήστη σε μορφή κειμένου. Αυτό το κείμενο σπανίως ξεπερνά τα 10.000 tokens, μπορεί να υπάρξει διαφορά ανάλογα και με την διαχείριση των Υπερπαραμέτρων και της βάσης γνώσης, επομένως μετρικές όπως το ZeroSCROLLS, InfiniteBench και Needle in Haystack, μπορούν να φανούν χρήσιμες σε συγκεκριμένα συστήματα τα οποία υλοποιούν την αρχιτεκτονική RAG αλλά κατά κανόνα αυτό δεν ισχύει.

Το Llama Krikri 8B Instruct [39], η fine-tuned εκδοχή του Llama 3.1 8B Instruct μοντέλου στα ελληνικά από το Ινστιτούτο Επεξεργασίας Γλώσσας, αποτελεί μία πολύ καλή επιλογή για χρήση τόσο στην ελληνική όσο και στην αγγλική γλώσσα. Το Krikri μοντέλο, έγινε fine-tuned σε ένα σύνολο δεδομένων 91 δισεκατομμυρίου tokens, τα οποία αποτελούνταν από δεδομένα στην ελληνική και αγγλική γλώσσα, ένα κομμάτι δεδομένων που περιέχουν και τις δύο γλώσσες καθώς και δεδομένα τα οποία αφορούν μαθηματικά και κώδικα. Αναλυτικά τα δεδομένα εκπαίδευσης του μοντέλου:

Υποσύνολο Δεδομένων	Αριθμός Tokens	Ποσοστό
Ελληνικά	56.7 δισεκατομμύρια	62.3%
Αγγλικά	21.0 δισεκατομμύρια	23.1%
Παράλληλα	5.5 δισεκατομμύρια	6.0%
Μαθηματικά/ Κώδικας	7.8 δισεκατομμύρια	8.6%
Σύνολο	91 δισεκατομμύρια	100%

Πίνακας 4: Δεδομένα επανεκπαίδευσης του μοντέλου Krikri

Το μοντέλο Krikri παρουσιάζει πολύ καλή απόδοση στις επιλεγμένες από τους δημιουργούς μετρικές, όταν συγκρίνεται με μοντέλα παρόμοιου μεγέθους, και δείχνει την σαφή ανωτερότητα του από το μοντέλο Meltemi 7B. Οι μετρικές που επέλεξαν οι δημιουργοί του μοντέλου είναι η IFEval σε ελληνική και αγγλική γλώσσα μόνο, την μετρική που αφορά την κατανόηση κειμένου και την πιστότητα στην τήρηση κανόνων και οδηγιών, καθώς και τη μετρική MT-Bench (Multi Turn Benchmark) επίσης μόνο σε ελληνικά και αγγλικά. Η μετρική MT-Bench [40] χρησιμοποιεί πολύ ισχυρά μεγάλα γλωσσικά μοντέλα τα οποία κρίνουν τις απαντήσεις διαφορετικών μεγάλων γλωσσικών μοντέλων σε ερωτήσεις όπου γίνεται μία συζήτηση 8 γύρων με 4 ερωτήσεις και 4 απαντήσεις του μεγάλου γλωσσικού μοντέλου το οποίο αξιολογείται. Οι ερωτήσεις αφορούν πολλά διαφορετικά θέματα όπως λογικοί συλλογισμοί, μαθηματικά, παιχνίδια ρόλων (όπου το μεγάλο γλωσσικό μοντέλο καλείται να απαντήσει ως κάποιος συγκεκριμένος ρόλος) κ.α.. Στόχος αυτής της μετρικής είναι να αξιολογήσει τα μεγάλα γλωσσικά μοντέλα σε σενάρια που προσομοιώνουν αρκετά την εμπειρία του χρήστη. Η μετρική αξιολογεί από το 1 (εντελώς μη ικανοποιητική) έως 10 (άκρως ικανοποιητική).

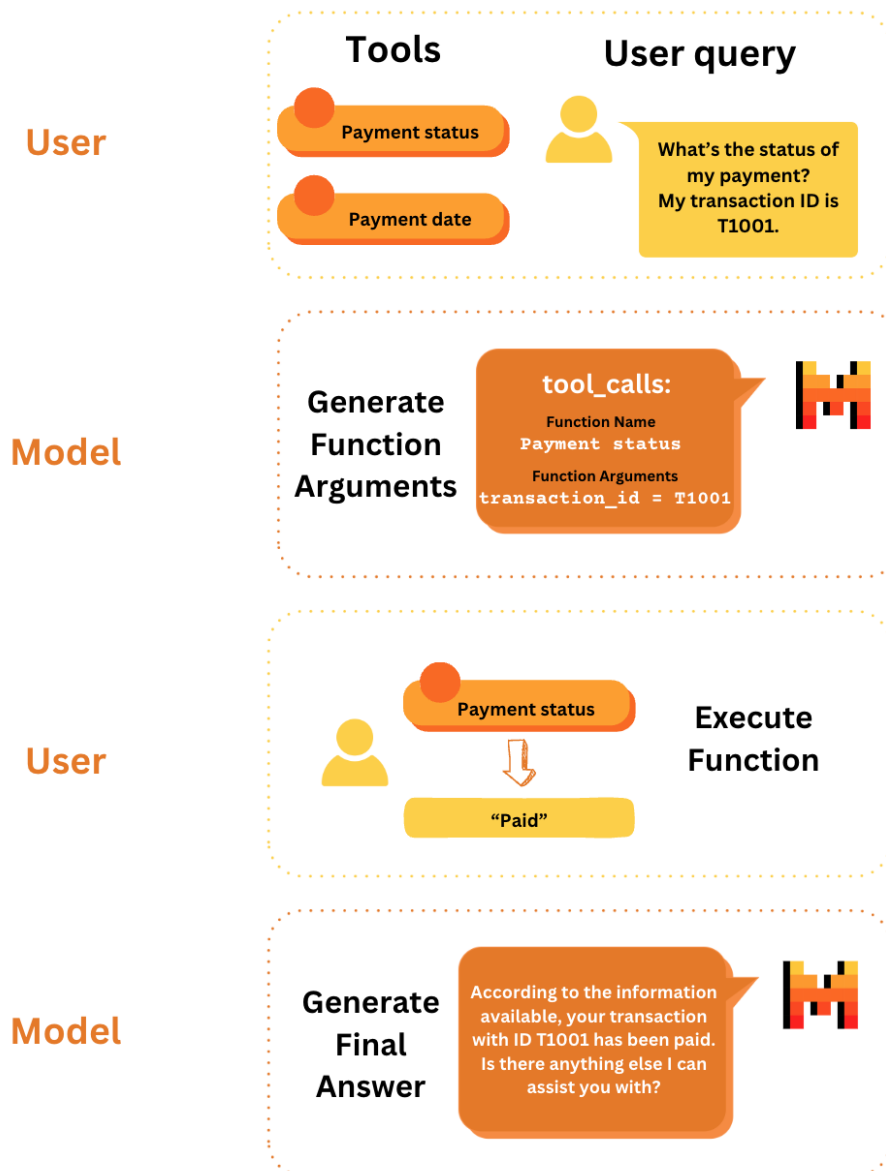
Μοντέλο	IFEval EL	IFEval EN	MT-Bench EL	MT-Bench EN
Qwen 2.5 7B Instruct	46.2%	74.8%	5.83	7.87
EuroLLM 9B Instruct	51.3%	64.5%	5.98	6.27
Aya Expanse 8B	50.4%	62.2%	7.68	6.92
Meltemi 7B v1.5 Instruct	32.7%	41.2%	6.25	5.46
Llama 3.1 8B Instruct	45.8%	75.1%	6.46	7.25
Llama Krikri 8B Instruct	67.5%	82.4%	7.96	7.21

Πίνακας 5: Επιδόσεις μοντέλου Krikri

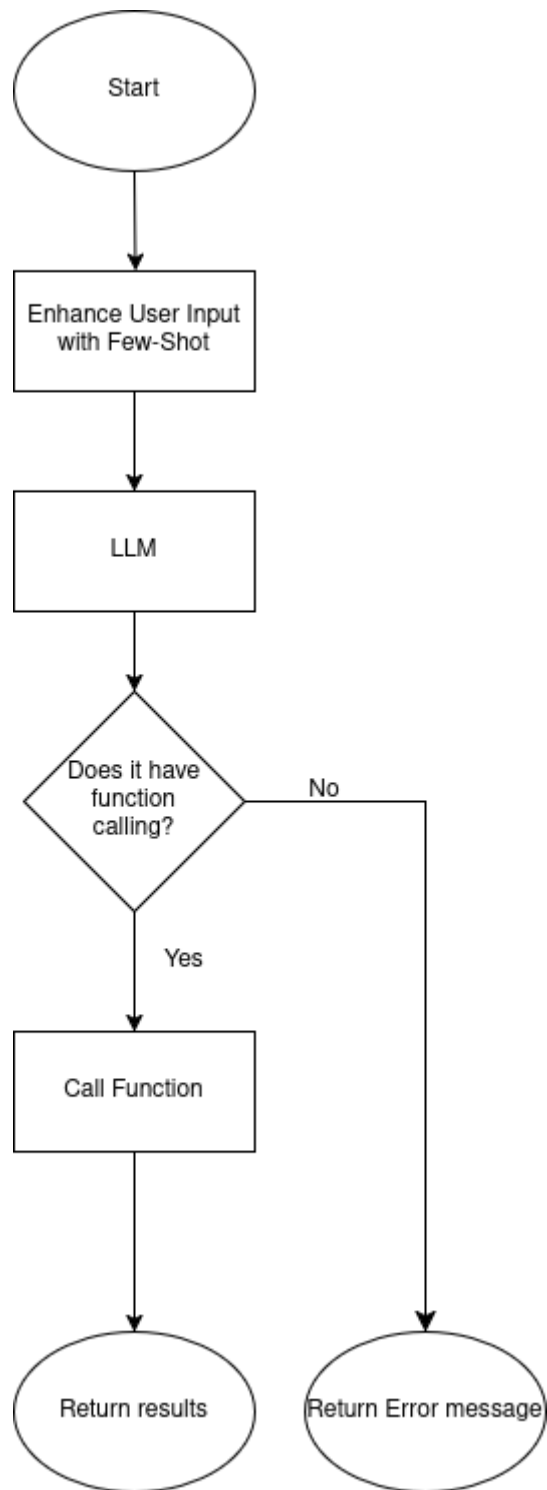
Η χρησιμοποίησης δεδομένων εκπαίδευσης και στην αγγλική γλώσσα αποδεικνύεται χρήσιμη καθώς το μοντέλο δείχνει βελτιωμένες επιδόσεις σε σύγκριση με το αρχικό πολυγλωσσικό (multilingual) μοντέλο στον δείκτη IFEval για την ελληνική γλώσσα και υστερεί ελάχιστα στον δείκτη MT-Bench. Το Krikri μοντέλο μοιάζει να είναι το καλύτερο αν λάβουμε υπόψη τους διαδεδομένους αυτούς δείκτες ωστόσο

για την διασφάλιση ότι η καλύτερη επιλογή έχει επιλεγεί στα πλαίσια της εφαρμογής της διπλωματικής εργασίας καθώς και για την αξιολόγηση της συμπεριφοράς των μεγάλων, αυτών, γλωσσικών μοντέλων σε πραγματικά δεδομένα ήταν σημαντική η περαιτέρω διερεύνηση των επιδόσεων των μοντέλων, περισσότερο στο Κεφάλαιο 5. Για τη δυνατότητα αξιολόγησης αλλά και μετέπειτα χρήσης στην εφαρμογή, υλοποιήθηκαν τα 8 services (υπηρεσίες) 2 για κάθε μοντέλο, χωρισμένα αναφορικά με την απαίτηση για function calling (κλήση συνάρτησης) και την αρχιτεκτονική RAG (Retrieval Augmented Generation). Η αρχιτεκτονική για κάθε διαφορετικό μοντέλο είναι ίδια για κάθε διεργασία, πιο συγκεκριμένα:

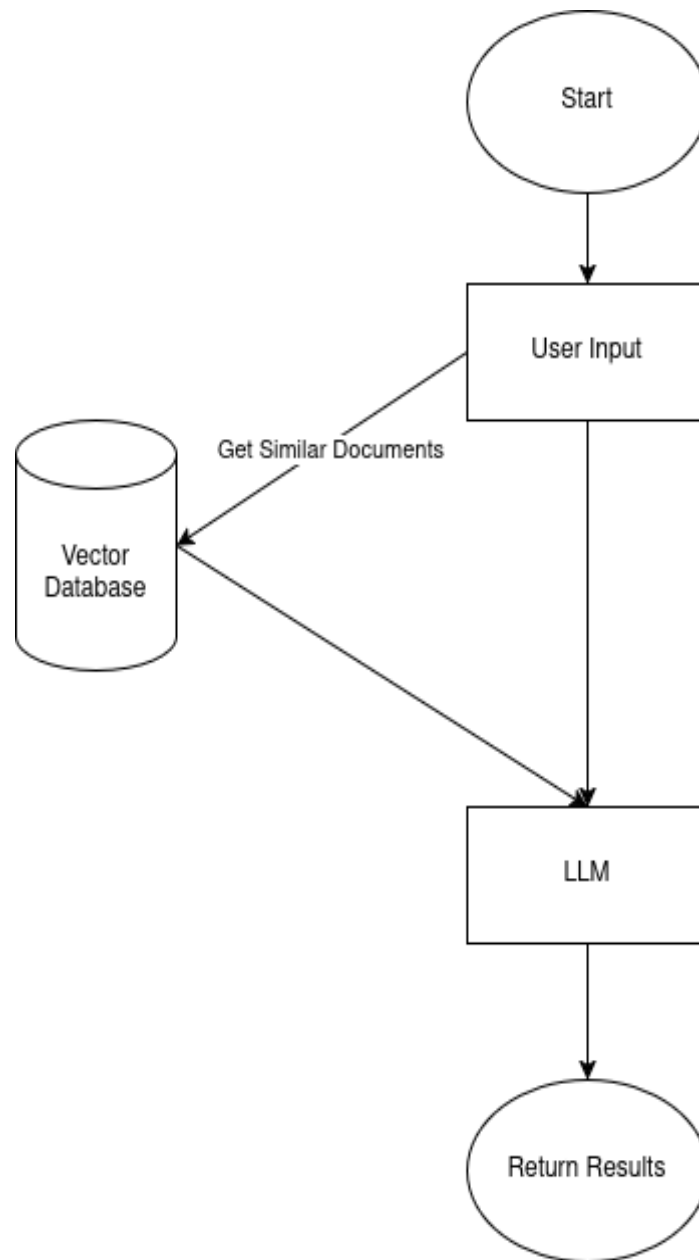
- **Function Calling Service:** Το κείμενο το οποίο έστειλε ο χρήστης, τροφοδοτείται στο μοντέλο μαζί με ένα κατάλληλο instruction prompt και 5 παραδείγματα τύπου few-shot examples, ένα για κάθε μία από τις πιθανές συναρτήσεις τις οποίες έχει στην διάθεση του το μεγάλο γλωσσικό μοντέλο. Μόλις το μεγάλο γλωσσικό μοντέλο παράξει την απάντηση, ελέγχεται η ύπαρξη κλήσης συνάρτησης, αν υπάρχει και είναι σωστά δομημένη, καλείται η συνάρτηση και το αποτέλεσμα της επιστρέφεται στο τέλος του service. Διαφορετικά, το service επιστρέφει ότι υπήρξε σφάλμα κατα την επεξεργασία και προτρέπει τον χρήστη να αλλάξει ελαφρώς την διατύπωση της ερώτησης.
- **Retrieval Augmented Generation Service:** Το service το οποίο υλοποιεί την αρχιτεκτονική RAG παίρνει το κείμενο το οποίο έστειλε ο χρήστης, και καλώντας το gero της διανυσματικής βάσης δεδομένων με είσοδο το κείμενο το χρήστη, παίρνει τα πιο κοντινά σημασιολογικά κομμάτια της βάσης και μαζί με το κείμενο το χρήστη δίνονται στο μεγάλο γλωσσικό μοντέλο το οποίο παράγει μία απάντηση στην ερώτηση του χρήστη χρησιμοποιώντας τα κοντινά σημασιολογικά δεδομένα ως βάση γνώσης.



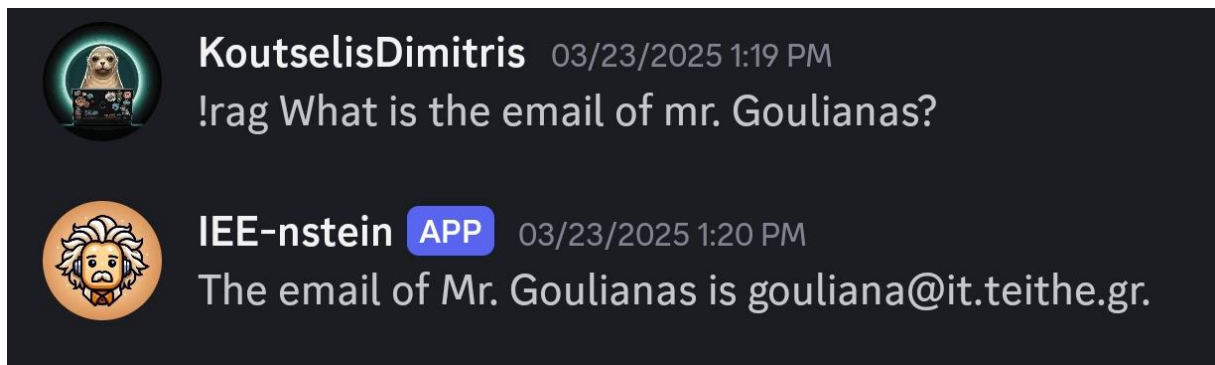
Σχήμα 17: Παράδειγμα function calling



Σχήμα 18: Διάγραμμα ροής του function calling service



Σχήμα 19: Διάγραμμα ροής του RAG service





KoutselisDimitris 12:34 PM

!rag Ποια είναι τα ΥΠ-ΕΠ μαθήματα στο προπτυχιακό πρόγραμμα σπουδών;



IEE-nstein APP 12:34 PM

Τα ΥΠ-ΕΠ (Υποχρεωτικά-Επιλογής) μαθήματα στο προπτυχιακό πρόγραμμα σπουδών είναι τα εξής:

- Σύνθεση Ηλεκτρονικών Κυκλωμάτων
- Αριθμητικές Μέθοδοι
- Συστήματα Αυτομάτου Ελέγχου
- Αισθητήρια και Επεξεργασία Μετρήσεων
- Εισαγωγή στην Αναλυτική των Δεδομένων
- Μηχανική Λογισμικού
- Εφαρμογές Ενσωματωμένων Συστημάτων
- Ρομποτική
- Ανάπτυξη Διαδικτυακών Συστημάτων και Εφαρμογών
- Επιχειρησιακή Έρευνα



KoutselisDimitris 2:00 PM

!rag Ποιό μάθημα γίνεται κάθε Τετάρτη 14:00-16:00 στην αίθουσα 101 και με ποιόν καθηγητή;



IEE-nstein APP 2:01 PM

Το μάθημα που γίνεται κάθε Τετάρτη 14:00-16:00 στην αίθουσα 101 είναι "Όργ. Δεδ. και Εξόρυξη Πληροφορίας" και με τον καθηγητή ο Ουγιάρουλου.



KoutselisDimitris 1:35 PM

!rag Ποιός και που διδάσκει το ρεαστήριο του μαθήματος Εισαγωγή στα Λειτουργικά Συστήματα;



IEE-nstein APP 1:35 PM

Οι Σιδηρόπουλος και Ιωαννίδης διδάσκουν το ρεαστήριο του μαθήματος Εισαγωγή στα Λειτουργικά Συστήματα στην αίθουσα 201.



KoutselisDimitris 1:33 PM

!rag κάθε πότε, σε ποια αίθουσα και με ποιον καθηγητή έχει μάθημα το εργαστήριο τ1/τ2 (Ομάδα Α) στον Αντικειμενοστρεφή Προγραμματισμό;



IEE-nstein APP 1:34 PM

Κάθε Δευτέρα από 09:00 έως 11:00 στην αίθουσα 211 με την κυρία Ασδρέ.



Σχήμα 20: Screenshots από την εφαρμογή κατά την χρήση RAG

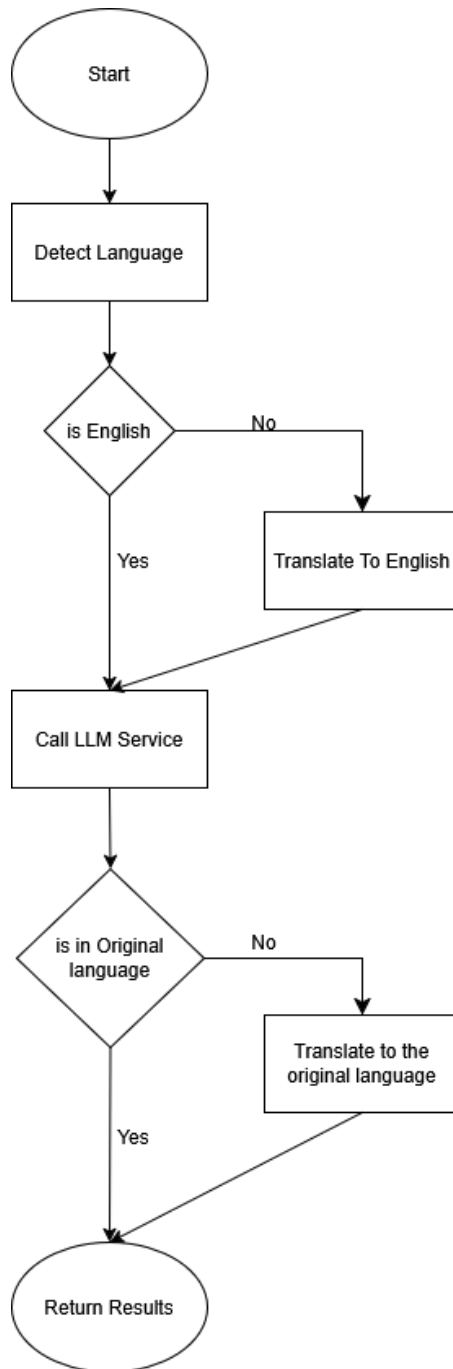
Οι δύο υπηρεσίες αυτές του μέρους της μηχανικής μάθησης της εφαρμογής καλούνται από το backend σαν τμήμα ενός μεγαλύτερου pipeline (αλυσίδα διεργασιών). Αυτό το pipeline περιλαμβάνει την αναγνώριση της χρησιμοποιούμενης γλώσσας με την βιβλιοθήκη langdetect της python καθώς και την

αυτόματη μετάφραση του κειμένου που δίνεται ως είσοδος στις παραπάνω υπηρεσίες στα αγγλικά, και μετέπειτα την μετάφραση της απάντησης στην αρχική γλώσσα του χρήστη.

Η διαδικασία της αναγνώρισης γλώσσας και της μετάφρασης επιλέχθηκε καθώς έδειξε καλύτερη συμπεριφορά στα πειράματα που έκανα, περισσότερο στο 5ο κεφάλαιο.

Μετά την λήψη ενός αιτήματος από τον χρήστη το backend καλεί δυο services τα οποία διαχειρίζονται ολόκληρο το pipeline της μηχανικής μάθησης. Η διαδικασία είναι η ίδια ανεξάρτητα από το service του μεγάλου γλωσσικού μοντέλου που εκτελείται. Αναλυτικά:

1. Τρέχει η βιβλιοθήκη langdetect και αναγνωρίζει την αρχική γλώσσα, αποθηκεύει την τιμή σε μία μεταβλητή
2. Αν η αρχική γλώσσα δεν είναι η αγγλική, καλείται το text-to-text translation service (υπηρεσία μετάφρασης από κείμενο σε κείμενο), το οποίο με την χρήση του SeamlessM4T μοντέλου μεταφράζει το κείμενο στα αγγλικά.
3. Το αγγλικό κείμενο τροφοδοτείται στην είσοδο της υπηρεσίας του μεγάλου γλωσσικού μοντέλου που πρέπει να χρησιμοποιηθεί.
4. Αν η αρχική γλώσσα δεν ήταν τα αγγλικά, η έξοδος μεταφράζεται στην αρχική γλώσσα του χρήστη.
5. Η απάντηση του μεγάλου γλωσσικού μοντέλου στην αρχική γλώσσα, επιστρέφεται.



Σχήμα 21: Διάγραμμα ροής του Machine Learning Service

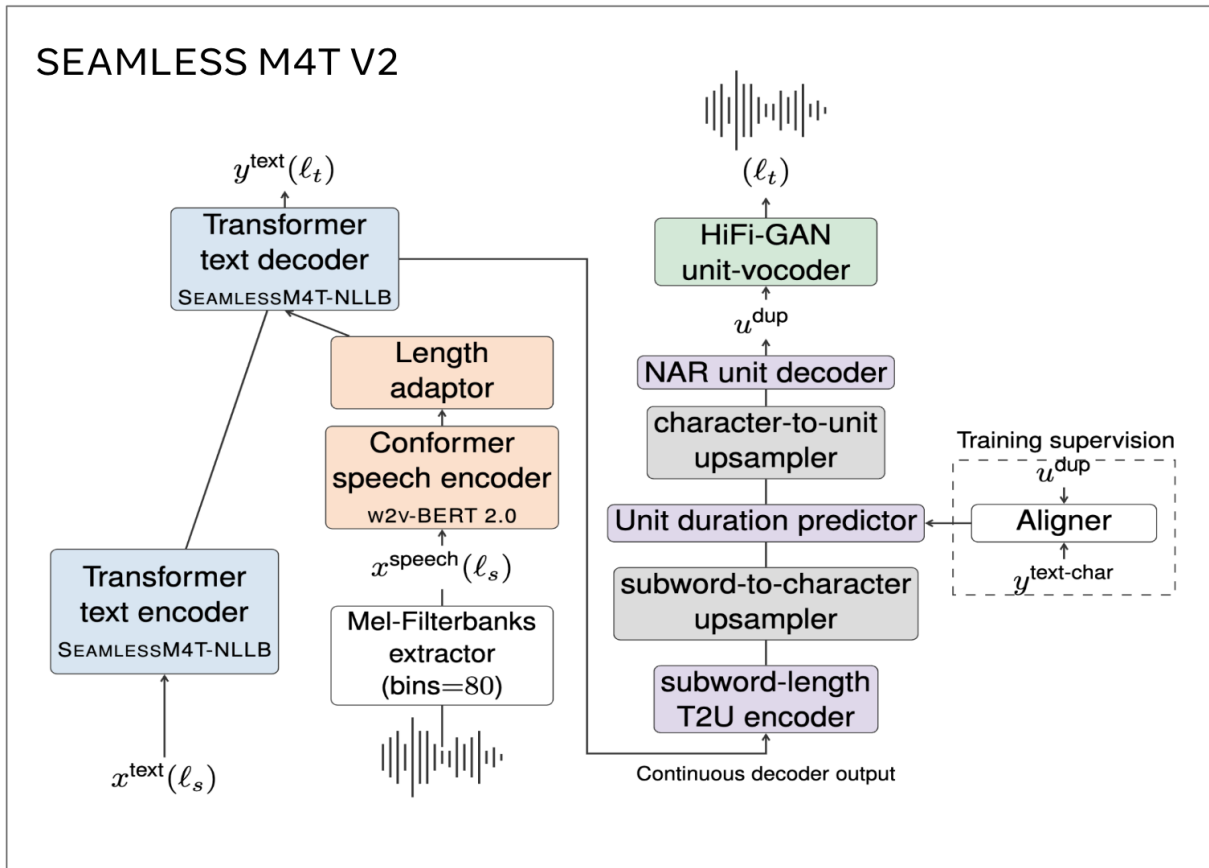
Η μετάφραση του κειμένου δεν είναι απαραίτητη και θα μπορούσε να αποφευχθεί με την χρήση ισχυρότερων μοντέλων ικανών να διεκπεραιώνουν αιτήματα σε αρκετές γλώσσες. Ωστόσο, στα πλαίσια της διπλωματικής μου εργασίας είναι σκόπιμο και απαραίτητο να ελεγχθεί η απόδοση των μεγάλων γλωσσικών μοντέλων fine-tuned στα ελληνικά συγκριτικά με τις αρχικές εκδοχές τους, έτσι χρησιμοποιήθηκαν αυτά τα μοντέλα και μετά από έλεγχο αποδείχτηκε καλύτερη η συγκεκριμένη μεθοδολογία. Η προσθήκη δύο κλήσεων σε μοντέλο μετάφρασης αυξάνει τον χρόνο απόκρισης του συστήματος σε ένα ερώτημα, ωστόσο η αύξηση στην ακρίβεια της εφαρμογής κρίνεται σημαντικότερη, δεδομένου ότι η αύξηση στον χρόνο απόκρισης είναι μικρή, συγκριτικά με το χρόνο τον οποίο χρειάζεται το μεγάλο γλωσσικό μοντέλο.

Στο κομμάτι της μηχανικής μάθησης ανήκει και μία από τις συναρτήσεις τις οποίες μπορεί να χρησιμοποιήσει το μεγάλο γλωσσικό μοντέλο ως εργαλείο, μέσω της κλήσης συναρτήσεων (function calling). Η συνάρτηση αυτή είναι η “summarization”, η οποία δέχεται ως είσοδο κείμενο, καλεί ένα εξειδικευμένο μοντέλο το οποίο με την σειρά του παράγει την περίληψη του κειμένου η οποία επιστρέφεται. Το μοντέλο το οποίο χρησιμοποιήθηκε είναι το BART Large από την facebook [3] προεκπαιδευμένο στην αγγλική γλώσσα, το οποίο έγινε fine-tuned στο dataset CNN Daily Mail [39]. Αυτό το σύνολο δεδομένων αποτελείται από περιλήψεις τις οποίες έγραψαν άνθρωποι για νέα και ιστορίες που εμφανίστηκαν στις ιστοσελίδες των CNN και Daily Mail. Το σύνολο δεδομένων αποτελείται από περισσότερες από 300.000 εγγραφές, με την κάθε μία εγγραφή να είναι ένα ζευγάρι ιστορίας/νέου με μία περίληψη πολλών προτάσεων. Το μεγάλο γλωσσικό μοντέλο μπορεί να κάνει την συγκεκριμένη δουλειά σε έναν αρκετά ικανοποιητικό βαθμό, ωστόσο η ύπαρξη εξειδικευμένων εργαλείων βελτιώνει την επίδοση.

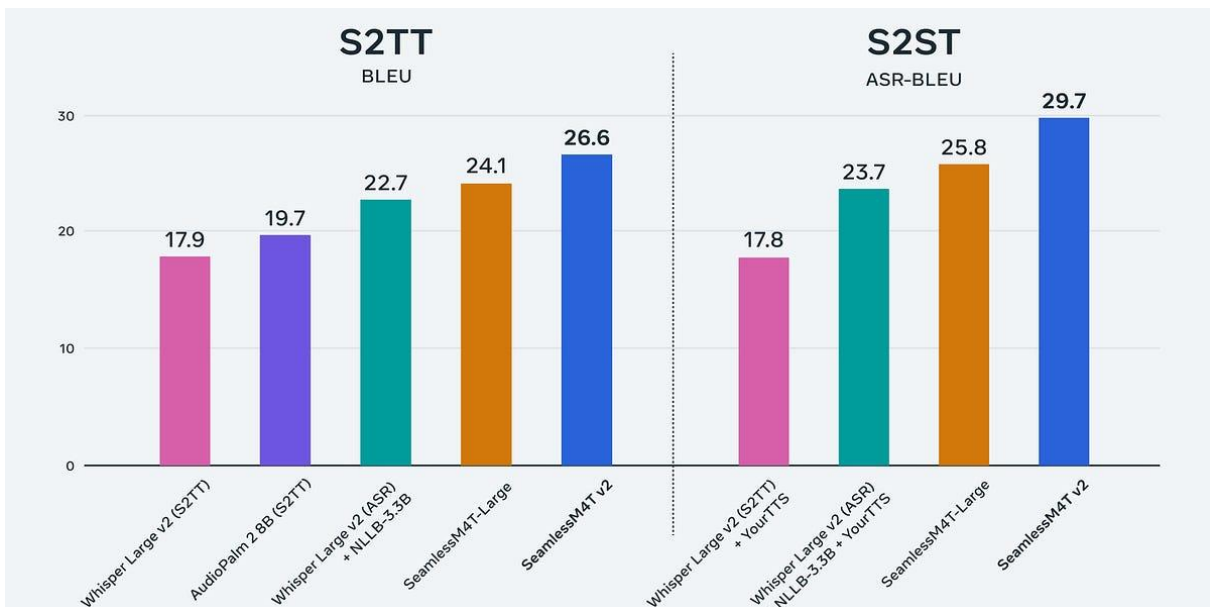
Το κομμάτι της μηχανικής μάθησης περιλαμβάνει και τα services της αναγνώρισης γλώσσας και της μετάφρασης. Τα δύο services είναι ξεχωριστά παρά το μικρό τους μέγεθος και την κοινή χρήση τους ώστε να μην παραβιάζεται η αρχή του διαχωρισμού αρμοδιοτήτων της καθαρής αρχιτεκτονικής. Το πρώτο service έχει πολύ απλή δομή, δέχεται ως είσοδο το κείμενο του χρήστη, το επεξεργάζεται (pre-processing) αφαιρώντας του ειδικούς χαρακτήρες και τυχόν κενά οι αλλαγές γραμμών οι οποίες δεν θα έπρεπε να υπάρχουν και μετέπειτα, τροφοδοτεί το κείμενο στην βιβλιοθήκη langdetect της python οι οποία αναγνωρίζει την γλώσσα και επιστρέφει το επεξεργασμένο κείμενο σε συνδυασμό με τον κωδικό της γλώσσας την οποία ανίχνευσε. Η υπηρεσία της μετάφρασης, διαθέτει 4 διαφορετικές συναρτήσεις οι οποίες μπορούν να αξιοποιηθούν στο έπακρο τις δυνατότητες του μοντέλου SeamlessM4T της Meta. Συγκεκριμένα, τα 4 σενάρια τα οποία υλοποιούνται είναι:

1. Είσοδος: Κείμενο → Έξοδος: Κείμενο
2. Είσοδος: Κείμενο → Έξοδος: Αρχείο Ήχου
3. Είσοδος: Αρχείο Ήχου → Έξοδος: Κείμενο
4. Είσοδος: Αρχείο Ήχου → Έξοδος: Αρχείο Ήχου

Το seamlessM4T είναι ένα πανίσχυρο μοντέλο, με τις επιδόσεις του να είναι αυτές που θέτουν τα standards της βιομηχανίας. Υλοποιήθηκαν και οι υπόλοιπες λειτουργίες του μοντέλου σε ξεχωριστές συναρτήσεις ώστε να υπάρχει η δυνατότητα χρήσης τους σε μελλοντική υλοποίηση μίας διαφορετικής διεπαφής, όπως εφαρμογή σε κινητό, δεδομένου ότι στην παρούσα υλοποίηση χρήσιμη είναι μόνη η μετάφραση από κείμενο σε κείμενο. Το μοντέλο αυτό διαθέτει υποστήριξη στην κατανόηση της ελληνικής γλώσσας σε κείμενο και ηχητική μορφή, για τροφοδότηση ως είσοδο, ωστόσο δεν διαθέτει την υποστήριξη παραγωγής ηχητικού στα ελληνικά, παρά μόνο μετάφραση σε ελληνικά στην μορφή του κειμένου, έτσι στην περίπτωση όπου η γλώσσα παραγωγής ηχητικού δεν υποστηρίζεται από το seamlessM4T μοντέλο, γίνεται χρήση της gTTS (google Text-To-Speech) βιβλιοθήκης της python [41], μέσω της οποίας παράγεται ηχητικό από το κείμενο το οποίο παρήγαγε το seamlessM4T μοντέλο και αποτελεί την μετάφραση του αρχικού κειμένου. Είναι σημαντικό να αναφερθεί ότι οι κωδικοί των γλωσσών για την βιβλιοθήκη langdetect και το seamlessM4T μοντέλο διαφέρουν κατα περίπτωση, για αυτό τον λόγο απαιτείται mapping των κωδικών για τις δύο υπηρεσίες (services).



Σχήμα 22: Αρχιτεκτονική seamlessm4t μοντέλου



Σχήμα 23: Επιδόσεις seamlessm4t μοντέλου

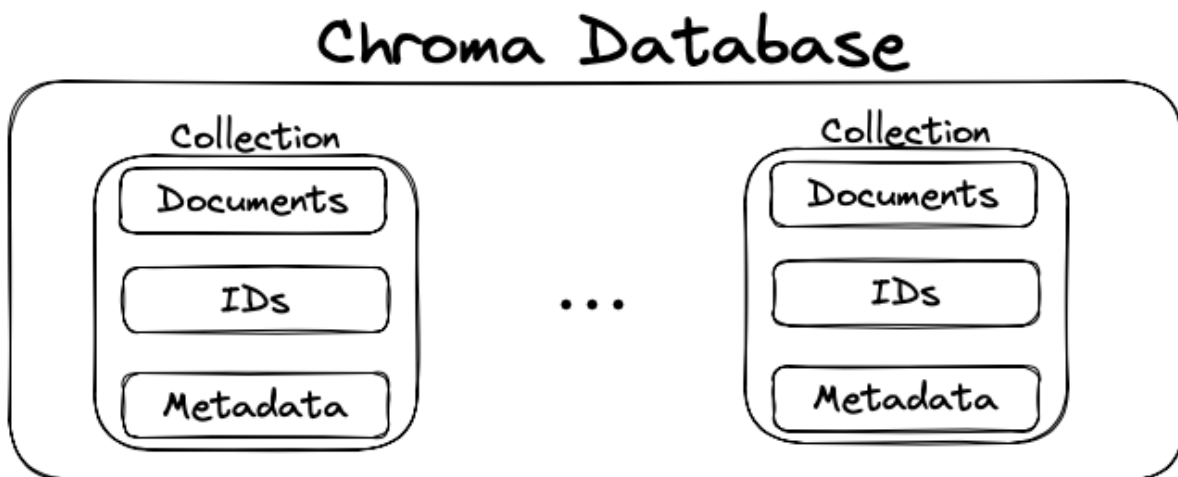
Τέλος, το τελευταίο τμήμα της μηχανικής μάθησης είναι το embeddings service, η υπηρεσία, δηλαδή, η οποία δέχεται ως είσοδο κείμενο και παράγει το διάνυσμα στο οποίο το κείμενο αντιστοιχεί. Η υπηρεσία διανυσματοποίησης είναι σημαντική τόσο για την αποθήκευση των δεδομένων στην διανυσματική βάση, όσο και για την χρήση της διανυσματικής βάσεις κατά τον υπολογισμό για την αναζήτηση των περισσότερο όμοιων εγγράφων. Για τον υπολογισμό των embeddings χρησιμοποιήθηκε το μοντέλο multilingual e5 large instruct [42] καθώς αποτελούσε το μοντέλο με την υψηλότερη

βαθμολογία στο leaderboard MTEB (Massive Text Embedding benchmark)[43] με native υποστήριξη της ελληνικής γλώσσας το οποίο έχει ως μοναδική χρήση την παραγωγή embeddings. Η ποιότητα των embeddings είναι καθοριστική για την επιτυχία της RAG αρχιτεκτονικής η οποία βασίζεται πάνω σε αυτά τόσο για την αποθήκευση όσο και για την εύρεση των εγγράφων.

4.5 Διανυσματική Βάση Δεδομένων

Οι διανυσματικές βάσεις δεδομένων αποτελούν απαραίτητο και αναπόσπαστο κομμάτι οποιασδήποτε εφαρμογής υλοποιεί την αρχιτεκτονική Retrieval Augmented Generation. Για την υλοποίηση της παρούσας εφαρμογής χρησιμοποιήσα την ChromaDB. Η ChromaDB είναι μια διανυσματική βάση δεδομένων ανοιχτού κώδικα ειδικά φτιαγμένη για χρήση με μεγάλα γλωσσικά μοντέλα για σημασιολογική αναζήτηση και ειδικότερα σε εφαρμογές οι οποίες υλοποιούν αρχιτεκτονική RAG. Με την χρήση της chromaDB επιτυγχάνεται η γρήγορη αποθήκευση των embeddings στην βάση, καθώς και η αναζήτηση με queries των οποίων την μετατροπή σε embeddings για αναζήτηση χειρίζεται μόνη της με το δοθέν στο configuration μοντέλο. Επιπρόσθετα, επιτρέπει την δυνατότητα αποθήκευσης μεταδεδομένων (metadata) ώστε να αποθηκεύεται περισσότερη πληροφορία και να δίνει δυνατότητα αναζήτησης εγγράφων μέσω αυτών. Η chromaDB επιλέχθηκε τόσο για την ικανότητα της να διαχειρίζεται την αναζήτηση με πολύ αποδοτικό τρόπο, χρησιμοποιώντας τον καλύτερο σε απόδοση αλγόριθμο όσο και για την απλότητα στην εγκατάσταση και την χρήση, όπου μέσω της chromadb βιβλιοθήκης υπάρχει άμεση σύνδεση του κώδικα και της διανυσματικής βάσεις χωρίς να χρειάζονται εξωτερικά εργαλεία ή κλήσεις σε APIs τα οποία υπάρχουν τοπικά όπως θα γινόταν με άλλες βάσεις.

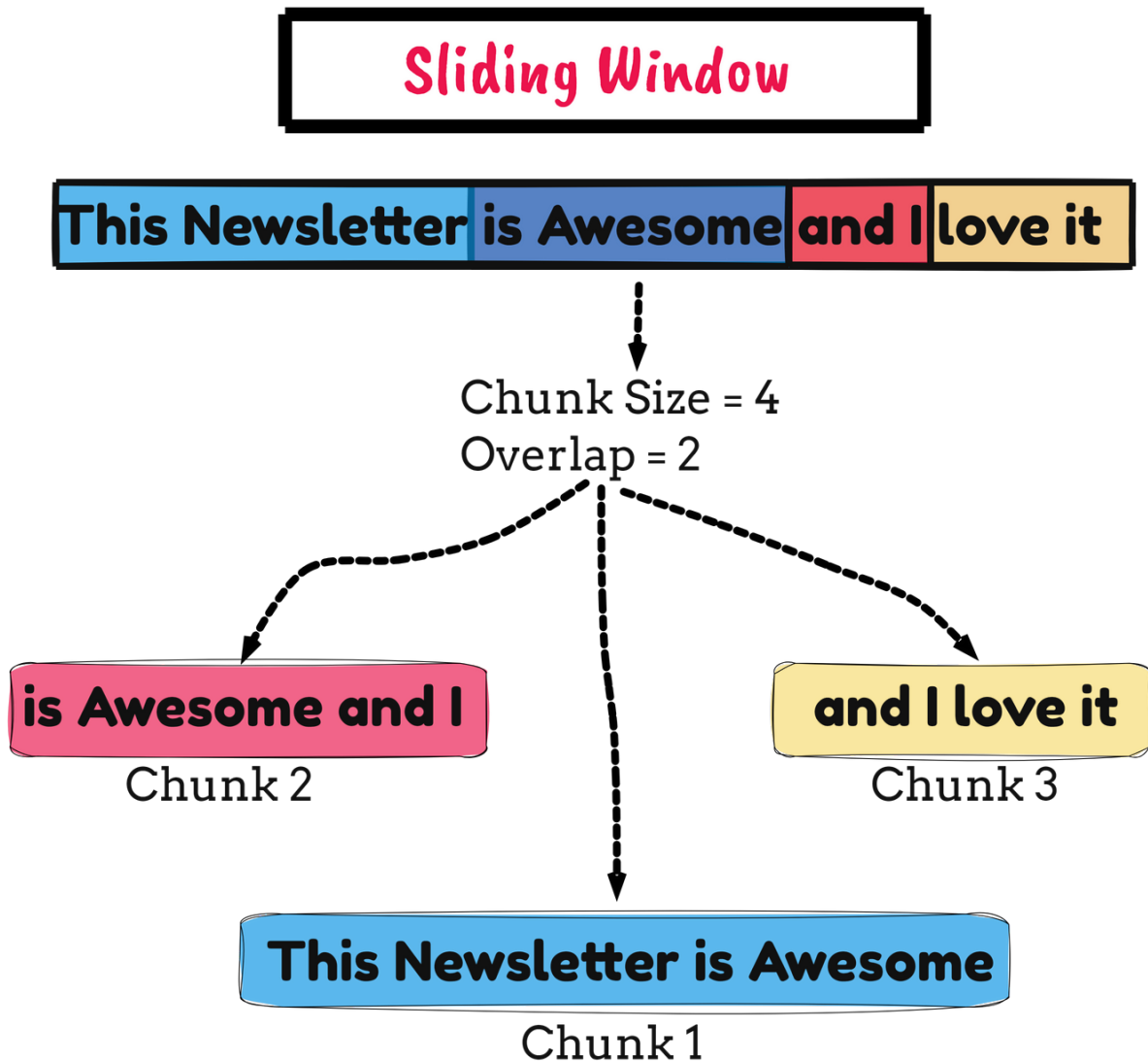
Για την επικοινωνία του κώδικα με τη διανυσματική δημιουργήθηκε ένα repo το οποίο διαχειρίζεται τα queries τόσο για εγγραφή στη βάση όσο και για την αναζήτηση. Για κάθε μία βάση δεδομένων μπορούν να υπάρξουν πολλές “συλλογές” (collections) τα οποία αποθηκεύουν, ως υποσύνολο της βάσης δεδομένων, με την σειρά του τα έγγραφα, τα ids και τα μεταδεδομένα αυτών.



Σχήμα 24: Αρχιτεκτονική της ChromaDB

Για την σωστή λειτουργία της αρχιτεκτονικής Retrieval Augmented Generation απαιτείται η σωστή διαχείριση των υπερπαραμέτρων κατά την αποθήκευση των εγγράφων στην διανυσματική βάση. Κατά την εισαγωγή εγγράφων στην διανυσματική βάση δεδομένων, είναι σύνηθες να χρησιμοποιείται η μέθοδος του κυλιόμενου παραθύρου (sliding window). Η μέθοδος αυτή παίρνει ως είσοδο το αρχικό έγγραφο και το σπάει σε ισομερή κομμάτια, τα οποία όμως δεν κόβονται απλά σειριακά διαιρώντας το αρχικό μέγεθος του εγγράφου με τον αριθμό των τμημάτων (chunks) αλλά σε κάθε καινούργιο κομμάτι υπάρχει ένα ποσοστό του προηγούμενου (overlap). Η μέθοδος αυτή χρησιμοποιείται ώστε η αναζήτηση να γίνεται στοχευμένα σε μικρότερα μέρη ολόκληρων εγγράφων τα οποία μπορεί να αποτελούν πολυσέλιδα αρχεία ή και βιβλία ώστε το μεγάλο γλωσσικό μοντέλο να δέχεται ως χρήσιμη πληροφορία

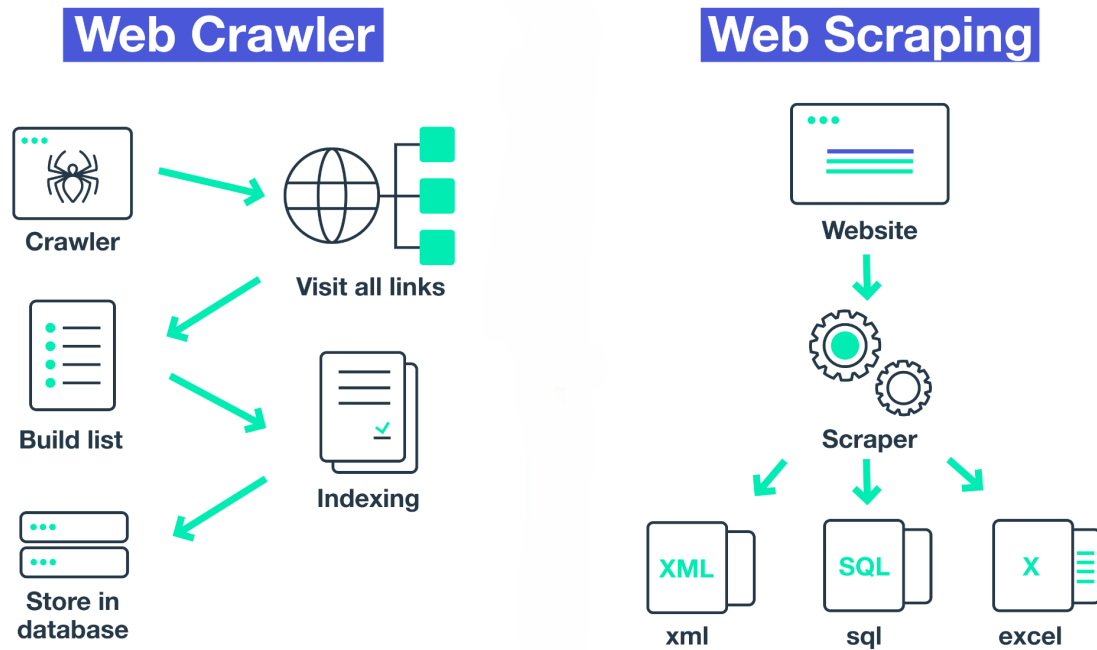
μόνο το απόσπασμα του αρχικού εγγράφου το οποίο διαθέτει όντως χρήσιμη πληροφορία, αλλά παράλληλα να μην χάνεται πληροφορία λόγω της αυθαίρετης διάσπασης του αρχικού εγγράφου.



Σχήμα 25: Παράδειγμα του sliding window

Τα δεδομένα τα οποία αποθηκεύτηκαν στην διανυσματική βάση δεδομένων είναι ολόκληρη η πληροφορία η οποία υπάρχει στην επίσημη ιστοσελίδα του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος. Ο τρόπος άντλησης των δεδομένων είναι το web scraping με την χρήση της βιβλιοθήκης beautiful soup της rython, αφού πρώτα υπήρξε μία διαδικασία web crawling. Η διαδικασία που ακολουθήθηκε ήταν η εξής:

1. Ξεκινώντας από την αρχική σελίδα, έγινε μία δημιουργία με όλες τις υποδιευθύνσεις (web crawling)
2. Από την λίστα αφαιρέθηκαν οι υποδιευθύνσεις οι οποίες αφορούσαν τις ανακοινώσεις και τις εκδηλώσεις, λόγω του τεράστιου όγκου δεδομένων
3. Στην λίστα η οποία απέμεινε έγινε web scraping με αποθήκευση ολόκληρου του κειμένου της κάθε σελίδας σε .txt αρχεία. Όλα τα έγγραφα με την μορφή .pdf τα οποία υπήρχαν μετατράπηκαν σε κείμενο και αποθηκεύτηκαν σε .txt μέσω της βιβλιοθήκης PyPDF
4. Για κάθε ένα από τα .txt αρχεία έγινε η απαιτούμενη προεπεξεργασία των δεδομένων, εφαρμόστηκε η μέθοδος sliding window
5. Τέλος, μετατράπηκαν σε embeddings και αποθηκεύτηκαν στην διανυσματική βάση δεδομένων.

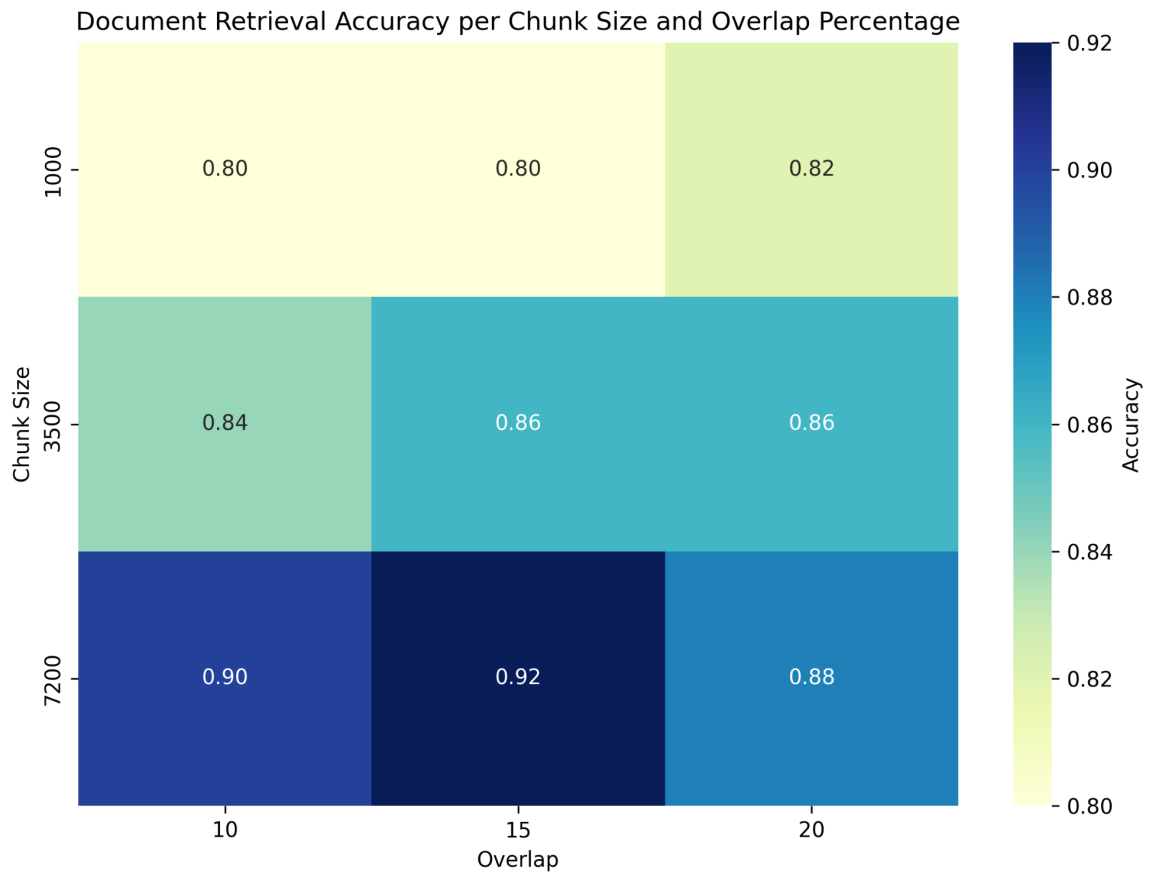


Σχήμα 26: Web crawler vs Web Scraping

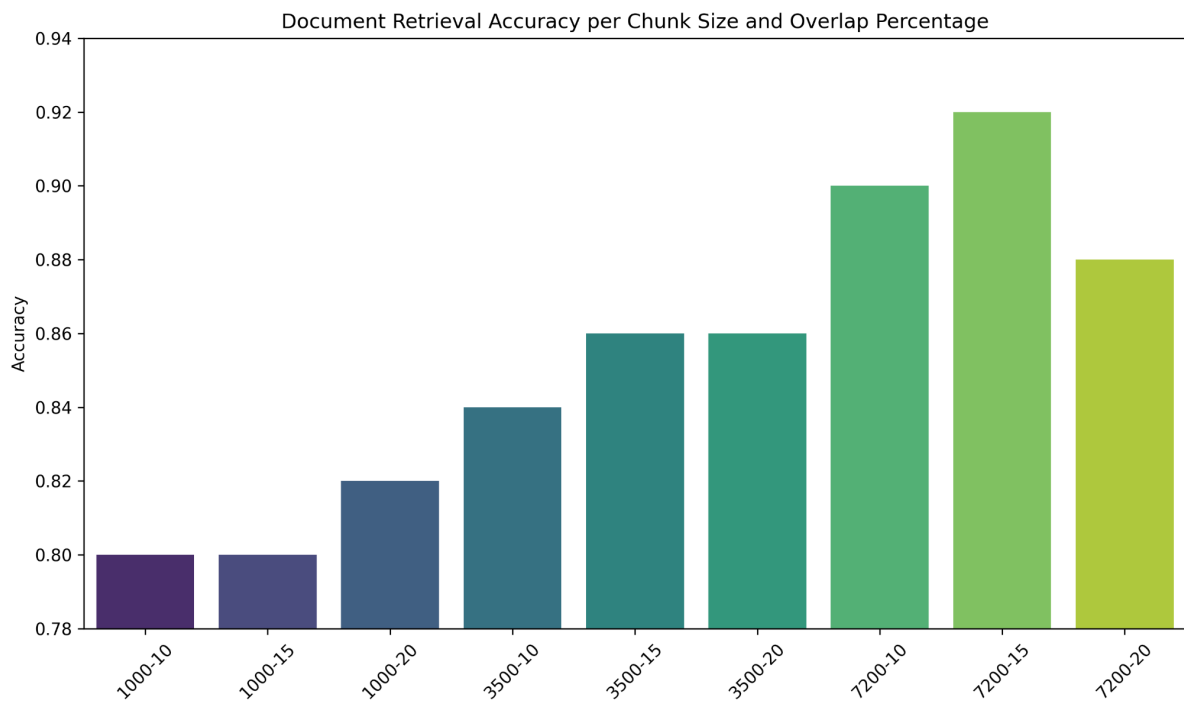
Είναι σημαντικό να αναφερθεί ότι μερικά δεδομένα ήταν ανέφικτο να παρθούν με αυτήν την τεχνική, λόγω της μορφής τους, ένα παράδειγμα αποτελεί η διεύθυνση email κάθε καθηγητή, η οποία έχει την μορφή εικόνας και όχι κειμένου, επομένως δεν μπορούσε να αναγνωριστεί σαν κείμενο μέσα σε html tags της ιστοσελίδας και έπρεπε να περαστεί με χειροκίνητα.

Το preprocessing το οποίο έγινε στα έγγραφα ήταν να ορισθεί ένα ελάχιστο μέγεθος (minimum length) για τα έγγραφα το οποίο ορίστηκε με βάση το μεγαλύτερο έγγραφο στην κατηγορία “Προσωπικό” όπου βρίσκονται οι πληροφορίες για το εκπαιδευτικό και όχι μόνο προσωπικό. Αυτό επιλέχθηκε διότι κατά την διάρκεια του ελέγχου λειτουργίας της εφαρμογής ήταν σύνηθες φαινόμενο να ζητείται πληροφορία σχετική με κάποιο μέρος του προσωπικού η οποία όμως δεν υπήρχε στα κοντινότερα σημασιολογικά έγγραφα διότι το όνομα του καθηγητή βρίσκεται στην αρχή του εγγράφου, ενώ το email στο τέλος. Με αυτόν τον τρόπο διασφαλίζεται ότι ολόκληρη η πληροφορία για όλα τα μέλη του προσωπικού είναι αποθηκευμένη σαν ένα έγγραφο και μπορεί να ανακτηθεί χωρίς πρόβλημα. Έτσι το document size (μέγεθος του εγγράφου) ορίστηκε στο 7200. Λόγω του μεγάλου μεγέθους του document size το overlap κινήθηκε σε χαμηλά επίπεδα, στο 15%, διότι το chunk κατά πάσα πιθανότητα ήδη έχει ολόκληρη την χρήσιμη πληροφορία.

Δημιούργησα ένα μικρό dataset από queries σε ελληνικά και αγγλικά για να ελέγξω την απόδοση με διάφορες τιμές σε chunk size και overlap. Ελέγχθηκαν οι τιμές 1000, 3500, 7200 για το chunk size ενώ οι τιμές 10%, 15%, 20% για το overlap. Το dataset περιλάμβανε 50 ερωτήσεις 25 στα ελληνικά και 25 στα αγγλικά οι οποίες είχαν την μορφή ερώτησης, προσομοιώνοντας την είσοδο που θα είχε η εφαρμογή από έναν πραγματικό χρήστη, καθώς και την λέξη ή πρόταση την οποία θεώρησα ως χρήσιμη πληροφορία η οποία θα πρέπει να υπάρχει στην έξοδο μετά το similarity search. Οι ερωτήσεις στο dataset έχουν την μορφή: “Ποιό είναι το email του κ. Γουλιάνα;”, “Πόσο διαρκεί ο κύκλος σπουδών του τμήματος;”, “Σε πόσους άξονες στηρίζεται ο κανονισμός των εξετάσεων;”, καθώς και την μεταφρασμένη στα αγγλικά εκδοχή τους. Αναλυτικά τα αποτελέσματα των πειραμάτων:



Σχήμα 27: Heatmap πειραμάτων για υπερπαραμέτρους διανυσματικής βάσης



Σχήμα 28: Διάγραμμα ράβδων για υπερπαραμέτρους διανυσματικής βάσης

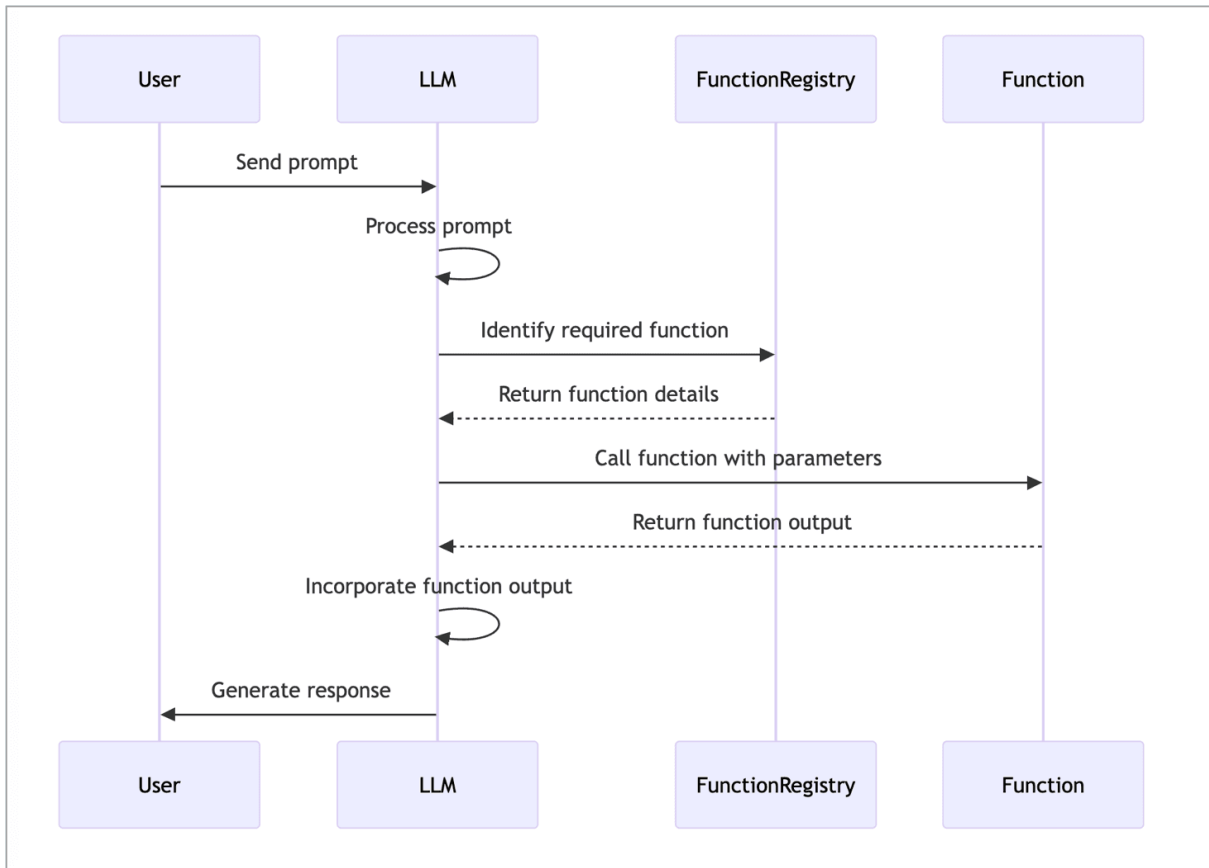
Στα πλαίσια του πειράματος, χρησιμοποιήθηκε $\text{top}_k = 1$, δηλαδή για κάθε ερώτηση η βάση επέστρεφε μόνο το κοντινότερο σημασιολογικά έγγραφο, με την αύξηση του top_k σε τιμές όπως 3 ή 5 το accuracy άγγιζε το 100% για όλες τις τιμές. Οι διαφορές για όλες τις τιμές είναι αρκετά μικρές με το accuracy να ξεκινάει από το 80% (40/50) ένα αρκετά ικανοποιητικό ποσοστό όπως ήταν αναμενόμενο. Κατά την διερεύνηση των αποτελεσμάτων θεωρήθηκε ενδιαφέρουσα η ύπαρξη τιμών όπως το 0.82 (41/50) δεδομένου ότι οι ερωτήσεις ήταν 1-1 αγγλικά και ελληνικά άρα αναμενόταν ζυγός αριθμός σωστών ανακτήσεων. Πιθανές αιτίες αυτής της συμπεριφοράς είναι η έλλειψη μερικών εγγράφων στην αγγλική γλώσσα (όπως τα “έντυπα χρήσιμα για τους φοιτητές”) ή η συμπερίληψη και της αγγλικής εκδοχής της ιστοσελίδας του τμήματος με μερικά έγγραφα να ανακτώνται σωστά ως συναφή στην αγγλική γλώσσα, με μικρότερη επιτυχία να παρουσιάζεται στην ελληνική.

Η ακρίβεια (accuracy) μετρήθηκε με την χρήση κανονικών εκφράσεων (regular expression) ελέγχοντας αν η λέξη ή πρόταση στόχος εμφανίζεται στο έγγραφο που επιστρέφεται από την διανυσματική βάση. Η ακρίβεια μπορεί να αγγίξει το 100% με την χρήση $\text{top}_k = 5$ ωστόσο, εάν επιστραφούν στο μεγάλο γλωσσικό μοντέλο 5 έγγραφα με 5 διαφορετικά emails αυξάνονται σημαντικά οι πιθανότητες λάθος πληροφόρησης του χρήστη από το μεγάλο γλωσσικό μοντέλο, με την παραγωγή λανθασμένων στοιχείων. Η χρήση τιμής μεγαλύτερης του 1 είναι πολύ χρήσιμη σε εφαρμογές που αφορούν την χρήση της αρχιτεκτονικής Retrieval Augmented Generation όταν η χρήσιμη πληροφορία είναι στο μέγεθος παραγράφων αλλά όταν πρόκειται για την αναζήτηση πληροφοριών όπως email, τηλέφωνο κ.ο.κ. η ύπαρξη περισσότερων από ένα έγγραφο είναι πιθανότερο να μειώσει την απόδοση παρά να την βελτιώσει.

4.6 Υπηρεσίες διαθέσιμες για Function Calling

Τα μεγάλα γλωσσικά μοντέλα δεν διαθέτουν βάσεις γνώσης, και όπως είδαμε παραπάνω παράγουν κείμενο με βάση το τι θεωρούν λογική συνέχεια του κειμένου το οποίο δέχτηκαν σαν είσοδο. Για να παράξουν αυτό το κείμενο βασίζονται στα δεδομένα κειμένου στα οποία εκπαιδεύτηκαν κατά την διάρκεια του pre-training τα οποία συλλέχθηκαν μέχρι κάποια συγκεκριμένη ημερομηνία, έτσι τα μεγάλα γλωσσικά μοντέλα δεν έχουν την δυνατότητα απόκρισης με επικυρωμένα δεδομένα. Για να προστεθεί αυτή η ικανότητα χρησιμοποιούνται οι ικανότητες των μοντέλων αυτών να καλούν συναρτήσεις, οι οποίες με την σειρά τους στέλνουν αιτήματα σε εξωτερικά APIs για να λάβουν επικυρωμένα δεδομένα.

Το function calling (κλήση συνάρτησης) αποτελεί μία από τις σημαντικότερες λειτουργίες ενός μεγάλου γλωσσικού μοντέλου, όταν εκείνο υπάρχει η επιθυμία να χρησιμοποιηθεί ως ψηφιακός βοηθός. Πολλά διαφορετικά εργαλεία μπορούν να αποτελέσουν εργαλεία στην φαρέτρα ενός μεγάλου γλωσσικού μοντέλου, πέρα από την κλήση εξωτερικών APIs όπως η χρήση εξειδικευμένων συναρτήσεων για μαθηματικούς υπολογισμούς, η χρήση διαφορετικών εξειδικευμένων μοντέλων για την υλοποίηση μίας διεργασίας, ακόμα και συχνά χρησιμοποιούμενη λειτουργία των δημοφιλών εφαρμογών ψηφιακών βοηθών για παραγωγή εικόνων ή βίντεο, καθώς και η αναζήτηση στο διαδίκτυο αποτελούν παραδείγματα κλήσης συναρτήσεων από τα μεγάλα γλωσσικά μοντέλα.



Σχήμα 29: Αρχιτεκτονική function calling

Για την σωστή χρήση της ικανότητας αυτής των μεγάλων γλωσσικών μοντέλων είναι απαραίτητο να δοθούν few-shot examples στο μοντέλο, ώστε η κλήση συνάρτησης από το μοντέλο να ακολουθεί ένα επαναλαμβανόμενο μοτίβο ώστε να μπορεί να κατανοήσει η εφαρμογή ότι πρόκειται για κλήση συνάρτησης και να την καλέσει σωστά. Few-shot examples είναι η μέθοδος όπου στο μεγάλο γλωσσικό μοντέλο μαζί με το κείμενο εισόδου του χρήστη στέλνονται παραδείγματα που εξομοιώνουν την αλληλεπίδραση του συστήματος με το χρήστη, έτσι το μεγάλο γλωσσικό μοντέλο θεωρεί λογική συνέχεια στο κείμενο του χρήστη ένα κείμενο που μοιάζει με αυτό που δόθηκε ως παράδειγμα, αυξάνοντας τις πιθανότητες όταν το μεγάλο γλωσσικό μοντέλο κρίνει ότι πρέπει να κληθεί συνάρτηση μία συνάρτηση να ακολουθήσει την μορφή την οποία είχαν τα παραδείγματα. Η μέθοδος αυτή είναι κοινώς χρησιμοποιούμενη στην κλήση συναρτήσεων και έχει επιδείξει τεράστιες αυξήσεις στην επίδοση των μικρών συγκριτικά, μεγάλων γλωσσικών μοντέλων [44].

Η μορφοποίηση που επιλέχθηκε για την κλήση συναρτήσεων και δόθηκε στο μεγάλο γλωσσικό μοντέλο ως few-shot examples είναι τα μπλοκ κώδικα με την χρήση των τριπλών backticks (```) στην αρχή και το τέλος τους, με αυτόν τον τρόπο είναι εύκολη η εξαγωγή του κώδικα κλήσης της συνάρτησης με εύκολο τρόπο.

```

4 messages = [
5     {"role": "system", "content": "You are an AI assistant capable of answering questions and calling functions when necessary."
6       "Do not suggest how to call a function, call it yourself."
7       "Available functions: "
8       "get_current_weather(location: str, unit: str = 'celsius'), "
9       "get_forecast_weather(location: str, days: int, unit: str = 'celsius'), "
10      "get_top_articles(country: str = 'us', category: str = 'general'), "
11      "get_articles_by_keywords(keywords: str, language: str = 'en'), "
12     },
13     {"role": "user", "content": "What is the weather now in Thessaloniki?"},
14     {"role": "assistant", "content": "```get_current_weather('Thessaloniki', 'celsius')```"},
15     {"role": "user", "content": "What will the weather be in Athens the next 3 days?"},
16     {"role": "assistant", "content": "```get_forecast_weather('Athens', 3, 'celsius')```"},
17     {"role": "user", "content": "What are the top news now in the US about technology?"},
18     {"role": "assistant", "content": "```get_top_articles('us', 'tech')```"},
19     {"role": "user", "content": user_input}
20 ]

```

Σχήμα 30: Απόσπασμα κώδικα υλοποίησης few shot examples

Συχνά χρησιμοποιούμενη είναι επίσης η χρήση των “λεξικών” της python όπου γίνεται δημιουργία ενός dictionary από το μοντέλο το οποίο γεμίζει το dictionary με ζευγάρια κλειδί-τιμή (key-value pairs) με την μορφή “function name”, “parameter 1” κ.ο.κ. Αυτή η μορφή είναι καλύτερη διότι την επιλογή μορφοποίησης της απάντησης, κάνοντας casting τιμές από συμβολοσειρές σε αριθμητικού τύπου δεδομένα, ενώ η το κείμενο το οποίο βγαίνει σαν κώδικας από το μοντέλο μέσα στο block δεν διαθέτει την δυνατότητα επεξεργασίας σε αυτό το επίπεδο. Ωστόσο, τα μικρά μοντέλα είναι σύνηθες να αλλάζουν ελάχιστα τα ονόματα στα key-value pairs ή αλλάζουν την μορφή που ακολουθούν οι τιμές, ήταν σύνηθες φαινόμενο στο function name τα μοντέλα να παράγουν ολόκληρη την κλήση της συνάρτησης με τις κανονικές τιμές των παραμέτρων μέσα στις παρενθέσεις, με τα ονόματα των παραμέτρων μέσα στις παρενθέσεις, το όνομα της συνάρτησης με κενές παρενθέσεις ή και χωρίς παρενθέσεις παρά την χρήση της few-shot μεθόδου. Με την χρήση ισχυρότερων μεγάλων γλωσσικών μοντέλων τα οποία θα μπορούσαν να ακολουθούν την δομή σε όλες τις περιπτώσεις συνιστάται η χρήση της μεθόδου των python dictionaries.

Είναι σημαντικό επίσης να αναφερθεί ότι προκειμένου το μεγάλο γλωσσικό μοντέλο να μπορεί να χρησιμοποιήσει συναρτήσεις και εργαλεία μέσω της μεθόδου function calling απαιτείται η σωστή δομή της συνάρτησης. Η δομή που πρέπει να ακολουθεί η συνάρτηση είναι η ξεκάθαρη ονομασία των παραμέτρων της σχετικά με το τι ρυθμίζουν, η δήλωση του τύπου της παραμέτρου, η οποία δεν είναι απαραίτητα στην python και τέλος η ύπαρξη ενός docstring, στην πρώτη γραμμή στο εσωτερικό της συνάρτησης. Μέσα στο docstring πρέπει να δηλώνει ξεκάθαρα η λειτουργία της συνάρτησης, το τι υλοποιεί καθώς και το τι πρεσβεύουν οι παράμετροι της συνάρτησης με την σειρά που εμφανίζονται, όπως και οι default τιμές τις οποίες μπορεί να πάρουν αν δεν οριστούν από το μεγάλο γλωσσικό μοντέλο ή τις πιθανές τιμές αν οφείλει η τιμή της παραμέτρου να είναι κατηγορικού τύπου, όπως π.χ. η μονάδα μέτρησης της θερμοκρασίας.

```

def get_current_weather(location: str, format: str):
    """
    Get the current weather.

    Args:
        location: The city and state, e.g. San Francisco, CA.
        format: The temperature unit to use. Infer this from the user's location. (choices: ["celsius", "fahrenheit"])
    """
    return current_weather(location, format)

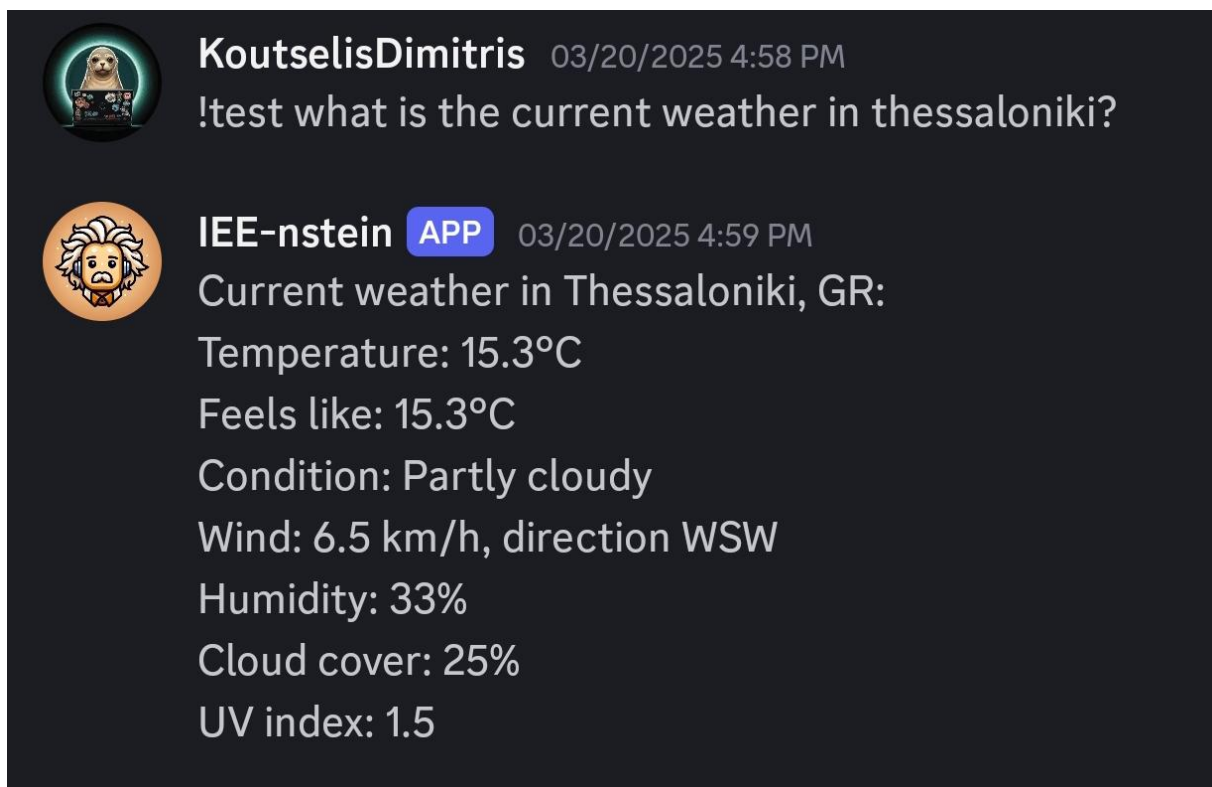
```

Σχήμα 31: Παράδειγμα συνάρτησης διαθέσιμη στο Μεγάλο Γλωσσικό Μοντέλο

Το μεγάλο γλωσσικό μοντέλο στην εφαρμογή που υλοποιήθηκε έχει στην διάθεση του 5 διαφορετικές συναρτήσεις. Οι 4 κάνουν χρήση εξωτερικών APIs για να φέρουν επικυρωμένα δεδομένα τα οποία μπορεί να είναι χρήσιμα στους φοιτητές. Τα δύο αφορούν τον καιρό, με την χρήση του weatherapi, ενός API ανοιχτού κώδικα που επιτρέπει έναν μεγάλο αριθμό ερωτημάτων στο δωρεάν επίπεδο,

υλοποιούνται δύο λειτουργίες, αυτή της επιστροφής του καιρού αυτή την στιγμή σε οποιοδήποτε μέρος στον κόσμο, καθώς και αυτή της πρόβλεψης του καιρού για ένα διάστημα 2 εβδομάδων. Αναλυτικά, για την ενημέρωση για τον καιρό αυτή τη στιγμή, χρειάζεται απλώς να σταλεί το όνομα της πόλης στο API. Έπειτα, το API επιστρέφει αρκετές τιμές σε πολλούς δείκτες, επιλέχθηκε να κρατηθεί ένα υποσύνολο αυτών και πιο συγκεκριμένα:

- **Θερμοκρασία:** Η θερμοκρασία περιβάλλοντος με την δυνατότητα να είναι σε βαθμούς Κελσίου ή Φαρενάιτ
- **Πραγματική Αίσθηση:** Αφορά την αίσθηση της θερμοκρασίας στους ανθρώπους σε συνδυασμό με τους υπόλοιπους παράγοντες πέρα από την θερμοκρασία.
- **Κατάσταση:** Η κατάσταση του καιρού αφορά το εάν η μέρα είναι ηλιόλουστη, συννεφιά, βροχή, κ.ο.κ.
- **Αέρας:** Επιστρέφεται τόσο η ταχύτητα του αέρα σε χιλιόμετρα ανα ώρα, όσο και η κατεύθυνση του αέρα.
- **Υγρασία:** Το ποσοστό της υγρασίας στην ατμόσφαιρα εκφρασμένο σαν ποσοστό τοις εκατό (%).
- **Κάλυψη από σύννεφα:** Το ποσοστό του ουρανού το οποίο καλύπτεται από σύννεφα, εκφρασμένο σαν ποσοστό τοις εκατό (%).
- **UV Index:** Ο UV index είναι ένας δείκτης μέτρησης της δύναμης της υπεριώδους ακτινοβολίας, με τιμές από 0 (χαμηλή) έως 11 (ακραία).



Σχήμα 32: Screenshot από την εφαρμογή με κλήση συνάρτησης για τωρινό καιρό

Για την πρόβλεψη του καιρού των επόμενων ημερών είναι απαραίτητη η αποστολή του ονόματος της πόλης για την οποία θέλει ο χρήστης την πρόβλεψη του καιρού, καθώς και τον αριθμό των ημερών για τις οποίες θέλει πρόβλεψη. Αυτό το endpoint του Weather API διαθέτει λιγότερες κατηγορίες δεδομένων όπως είναι φυσιολογική καθώς πρόκειται για πρόβλεψη και όχι για μέτρηση. Συγκεκριμένα τα δεδομένα που χρησιμοποιήθηκαν ήταν:

- **Κατάσταση:** Η κατάσταση του καιρού, π.χ. Ηλιόλουστος, Βροχή, Συννεφιά κ.ο.κ.
- **Μέγιστη τιμή θερμοκρασίας:** η προβλεπόμενη μέγιστη τιμή θερμοκρασίας η οποία θα εμφανιστεί την συγκεκριμένη μέρα
- **Ελάχιστη τιμή θερμοκρασίας:** η προβλεπόμενη ελάχιστη τιμή θερμοκρασίας για την ημέρα η οποία προβλέπεται

Αυτές οι τιμές έρχονται μαζί με την ημερομηνία της ημέρας πρόβλεψης.



Σχήμα 33: Screenshot από την εφαρμογή με κλήση συνάρτησης για πρόβλεψη καιρού

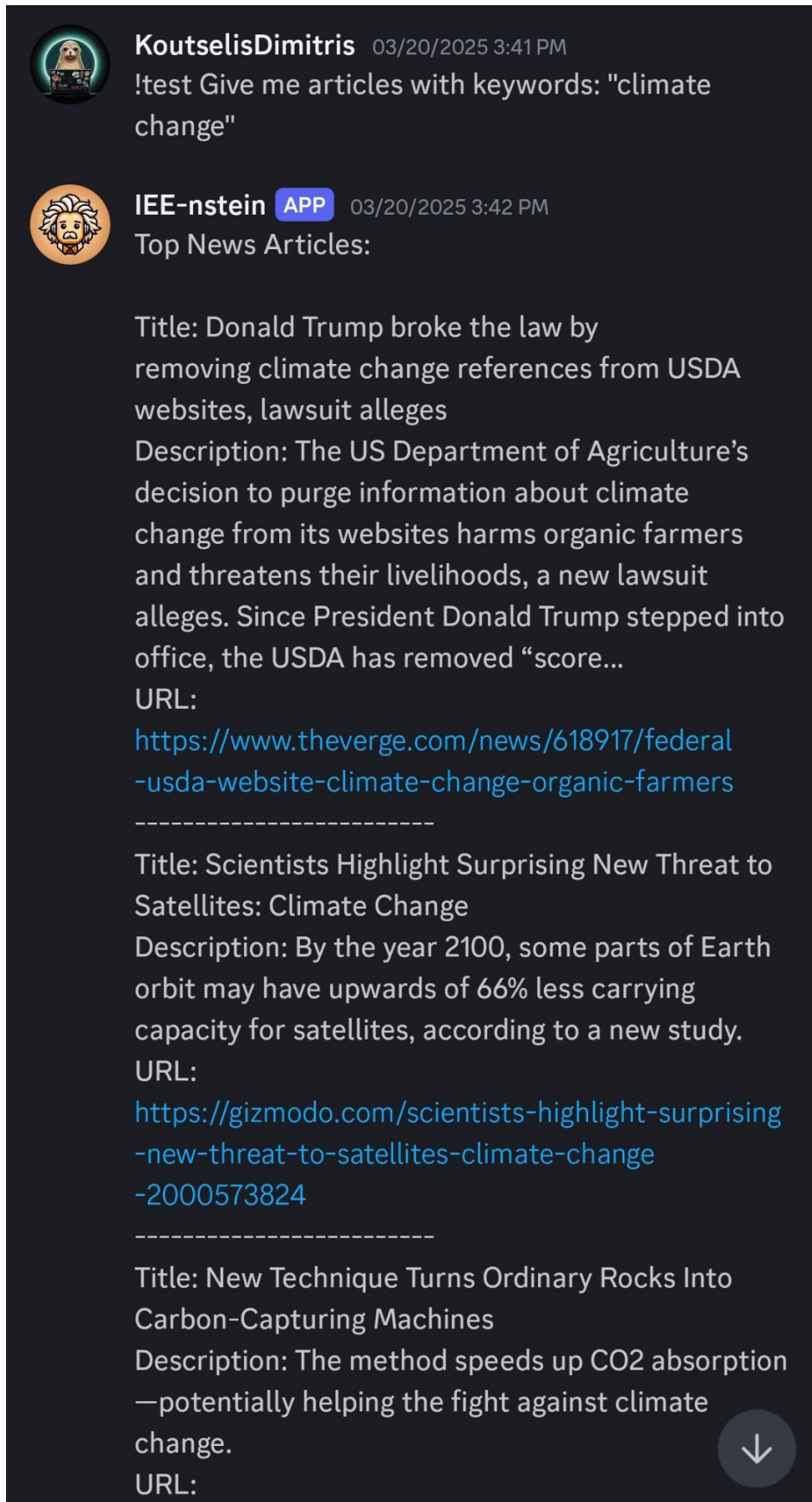
Στο μεγάλο γλωσσικό μοντέλο διατίθενται ακόμα δύο συναρτήσεις οι οποίες χρησιμοποιούν εξωτερικό API. Συγκεκριμένα χρησιμοποιείται το newsapi.org το οποίο αναζητά άρθρα σε περισσότερες από 150.000 πηγές ειδήσεων και ιστολογία, μέσα στα τελευταία 5 χρόνια. Χρησιμοποιούνται δύο endpoints του news API και κατά συνέπεια, λόγω της καθαρής αρχιτεκτονικής δημιουργήθηκαν δύο συναρτήσεις, μία η οποία επιστρέφει τις δημοφιλέστερες ειδήσεις μίας χώρας και μία η οποία επιστρέφει τις

δημοφιλέστερες ειδήσεις οι οποίες αφορούν ένα συγκεκριμένο πεδίο. Και τα δύο endpoints επιστρέφουν έναν μεγάλο αριθμό από ειδήσεις και άρθρα αλλά εξαιτίας της μορφής της διεπαφής χρήστη μέσω discord δεν μπορεί να γίνει κάποια βελτιστοποίηση ή αλλαγή του τρόπου εμφανίσεις των άρθρων, έτσι επιλέχθηκε να διατηρούνται μόνο τα 5 δημοφιλέστερα άρθρα. Τα endpoints έχουν την δυνατότητα να επιστρέφουν τις εξής πληροφορίες για κάθε άρθρο:

- **Source:** Η πηγή από την οποία προέρχεται το άρθρο, μέσα από το σύνολο των περισσότερων από 150.000 διαθέσιμων στο API πηγών
- **Author:** Το ονοματεπώνυμο του συγγραφέα της είδησης
- **Title:** Ο τίτλος του άρθρου
- **Description:** Η περιγραφή του άρθρου, μία μικρή περίληψη η οποία επιτρέπει στον χρήστη να καταλάβει τι αφορά το άρθρο.
- **URL:** Το URL είναι η ηλεκτρονική διεύθυνση την οποία μπορεί να επισκεφθεί ο χρήστης για να βρει ολόκληρο το άρθρο.
- **URLToImage:** Σε αυτό το πεδίο βρίσκεται ο σύνδεσμο στον οποίο είναι διαθέσιμη η εικόνα η οποία έχει επιλεγεί να φαίνεται στο εξώφυλλο του άρθρου.

Είναι σημαντικό να αναφερθεί πως το συγκεκριμένο API διαθέτει ελάχιστες ελληνικές πηγές ειδήσεων ωστόσο επιλέχθηκε διότι έχει έναν ασύλληπτα μεγάλο αριθμό πηγών για ένα ελεύθερο API σε πολλούς διαφορετικούς κλάδους. Επιπρόσθετα, η επιλογή της χρήσης άντλησης ειδήσεων σε έναν ψηφιακό βοηθό για φοιτητές πληροφορική και ηλεκτρονικής έχει προσανατολισμό στην άντληση ειδήσεων σχετικών με το συγκεκριμένο αντικείμενο με σκοπό οι φοιτητές να μπορούν να έχουν πρόσβαση στις εξελίξεις στον κλάδο εύκολα και απλά. Η κυριότερη βιβλιογραφία και ανάλυση της βιβλιογραφίας, καθώς και οι ειδήσεις σχετικές με τις εξελίξεις και τα νέα γράφονται στην αγγλική γλώσσα και δημοσιεύονται σε σημεία όπου το συγκεκριμένο API έχει πρόσβαση, οπότε εκπληρώνει στο έπακρο τον σκοπό χρήσης του παρά την έλλειψη διασύνδεσης με πηγές ελληνικών ιστοσελίδων.

Λόγω της φύσης της διεπαφής χρήστη –το discord διαχειρίζεται τους συνδέσμους, εμφανίζοντας αυτόματα τον τίτλο της ιστοσελίδας, και την φωτογραφία που τη συνοδεύει εαν υπάρχει– επιλέχθηκε η απάντηση της συνάρτησης να έχει μόνο τον τίτλο του άρθρου, μία περίληψη, καθώς η το discord δεν τις εμφανίζει, αλλά εμφανίζει τις πρώτες γραμμές του άρθρου και τον σύνδεσμο του άρθρου. Παρακάτω παρατίθεται screenshot από την εφαρμογή κατά την χρήση αυτών συναρτήσεων για την καλύτερη κατανόηση των αποτελεσμάτων και της διεπαφής.



The screenshot shows a chat interface with a dark background. At the top, a user named 'KoutselisDimitris' (profile picture of a dog) sends a message at 03/20/2025 3:41 PM: '!test Give me articles with keywords: "climate change"'. Below this, a bot named 'IEE-nstein' (profile picture of a cartoon character) responds at 03/20/2025 3:42 PM with 'Top News Articles:'. The bot lists three articles, each with a title, description, and URL. The first article is about a lawsuit against the USDA for removing climate change references. The second is about a study predicting a 66% decrease in satellite carrying capacity by 2100. The third is about a new technique for carbon capture from rocks. A downward arrow icon is visible in the bottom right corner of the chat area.

KoutselisDimitris 03/20/2025 3:41 PM
!test Give me articles with keywords: "climate change"

IEE-nstein APP 03/20/2025 3:42 PM
Top News Articles:

Title: Donald Trump broke the law by removing climate change references from USDA websites, lawsuit alleges
Description: The US Department of Agriculture’s decision to purge information about climate change from its websites harms organic farmers and threatens their livelihoods, a new lawsuit alleges. Since President Donald Trump stepped into office, the USDA has removed “score...
URL:
<https://www.theverge.com/news/618917/federal-usda-website-climate-change-organic-farmers>

Title: Scientists Highlight Surprising New Threat to Satellites: Climate Change
Description: By the year 2100, some parts of Earth orbit may have upwards of 66% less carrying capacity for satellites, according to a new study.
URL:
<https://gizmodo.com/scientists-highlight-surprising-new-threat-to-satellites-climate-change-2000573824>

Title: New Technique Turns Ordinary Rocks Into Carbon-Capturing Machines
Description: The method speeds up CO2 absorption —potentially helping the fight against climate change.
URL:

Σχήμα 34: Screenshot από την απάντηση του μοντέλου κατα την αναζήτηση ειδήσεων



Σχήμα 35: Screenshot από την αυτόματη διαχείριση των συνδέσμων από το discord

Τέλος, υπάρχει η συνάρτηση του summarization, είναι μία συνάρτηση που καλεί ένα εξειδικευμένο μοντέλο, εκπαιδευμένο αποκλειστικά στην παραγωγή περιλήψεων. Αναλύθηκε αρκετά ο τρόπος λειτουργίας του στο κομμάτι της μηχανικής μάθησης, ωστόσο είναι σημαντικό να αναφερθεί πως ο χρήστης έχει τη δυνατότητα να ορίσει το μέγεθος της περίληψης όταν ρωτήσει το μοντέλο, το οποίο με την δική του σειρά αναθέτει την αντίστοιχη τιμή κατά την κλήση της συνάρτησης.

4.7 Σύνοψη

1. Η εφαρμογή ακολουθεί την καθαρή αρχιτεκτονική μοιράζοντας όλες τις διαφορετικές διεργασίες σε διαφορετικά αυτόνομα services. Η διαδικασία υλοποίησης ενός ερωτήματος από τον χρήστη ακολουθεί την εξής διαδικασία:
2. Ο χρήστης επιλέγει αν θέλει να χρησιμοποιήσει την RAG ή την function calling λειτουργία του ψηφιακού βοηθού
3. Το discord service στέλνει HTTP request στο αντίστοιχο endpoint το οποίο πρέπει να χρησιμοποιηθεί
4. Γίνεται διαπίστωση της γλώσσας στην οποία είναι γραμμένη η είσοδος του χρήστη και μετάφραση στα αγγλικά
5. Καλείται η αντίστοιχη υπηρεσία μηχανικής μάθησης του μεγάλου γλωσσικού μοντέλου, το οποίο δέχεται δεδομένα από την διανυσματική βάση δεδομένων για το RAG ή από τις συναρτήσεις που έχει στην διάθεση της με το function calling.
6. Στην περίπτωση της χρήσης RAG και του Summarization η απάντηση μεταφράζεται στην αρχική γλώσσα στην οποία έκανε το ερώτημα ο χρήστης
7. Η απάντηση επιστρέφεται και εμφανίζεται στο discord ως μήνυμα το οποίο έστειλε το bot.

Κεφάλαιο 5ο: Διερεύνηση του καλύτερου Μοντέλου και των Υπερπαραμέτρων

5.1 Εισαγωγή

Για την επιλογή του ιδανικού μεγάλου γλωσσικού μοντέλου για την χρήση της εφαρμογής καθώς και των σωστών τιμών των υπερπαραμέτρων του δημιουργήσα ένα μικρό κριτήριο αξιολόγησης το οποίο αφορά αποκλειστικά τον συγκεκριμένο ψηφιακό βοηθό.

Το κριτήριο αξιολόγησης, αποτελείται από 20 ερωτήσεις 10 στην ελληνική γλώσσα και 10 στην αγγλική οι οποίες δεν έχουν 1-1 αντιστοιχία και αφορούν διαφορετικά θέματα. Για την κάθε γλώσσα οι ερωτήσεις μοιράστηκαν σε 5 και 5, για την αξιολόγηση του function calling και της αρχιτεκτονικής Retrieval Augmented Generation αντίστοιχα. Το dataset δεν είναι μεγάλο, ωστόσο για την επαρκές για την εξακρίβωση του καλύτερου συνδυασμού μοντέλου και υπερπαραμέτρων. Μετά το πέρας των πειραμάτων, επαναλήφθηκε μόνο για το επιλεγθέν μοντέλο ένα πείραμα με μεγαλύτερο dataset και οι αποκλίσεις στις επιδόσεις ήταν ελάχιστες.

Ο λόγος της χρήσης ενός τόσο μικρού συνόλου δεδομένων είναι ο περιορισμός εξαιτίας του hardware, για κάθε ερώτηση το μεγάλο γλωσσικό μοντέλο χρειάζεται περίπου 50 δευτερόλεπτα, ενώ εξετάστηκαν 9 διαφορετικοί συνδυασμοί υπερπαραμέτρων για κάθε μεγάλο γλωσσικό μοντέλο, σε συνδυασμό με την εξέταση της απόδοσης με την χρήση μετάφρασης των ερωτήσεων στα αγγλικά εξετάζοντας το κάθε μεγάλο γλωσσικό μοντέλο ξανά, δημιουργήθηκε ένα πείραμα διάρκειας 5 ωρών για κάθε μεγάλο γλωσσικό μοντέλο και 20 ωρών συνολικά.

5.2 Dataset

Το dataset που δημιουργήθηκε συνολικά αποτελείται από 40 ερωτήσεις εκ των οποίων χρησιμοποιήθηκαν οι μισές για την αξιολόγηση των μοντέλων. Το dataset είναι χωρισμένο σε 20 ερωτήσεις οι οποίες αφορούν την αξιολόγηση της αρχιτεκτονικής Retrieval Augmented Generation και 20 ερωτήσεις οι οποίες αφορούν την κλήση συναρτήσεων, ενώ και οι δύο κατηγορίες αποτελούνται από δέκα ερωτήσεις στην ελληνική γλώσσα και δέκα στην αγγλική. Επίσης δημιουργήθηκαν άλλες 20 ερωτήσεις οι οποίες αποτελούν την μεταφρασμένη από το seamless4t μοντέλο εκδοχή των 20 ερωτήσεων της ελληνικής γλώσσας ώστε να ελεγχθεί αν τα αποτελέσματα είναι καλύτερα με αυτή την ακολουθεί διεργασιών.

Τα δεδομένα που τοποθετούνται ως είσοδο στο σύστημα έχουν την μορφή ερωτήσεων τις οποίες θα μπορούσε να κάνει ένας χρήστης της εφαρμογής και για τις δύο κατηγορίες του dataset ενώ αυτό το οποίο διαφέρει είναι ο “στόχος” αναφορικά με τις ερωτήσεις. Αναλυτικά:

Κατηγορία	Ερώτηση	Στόχος
Function Calling	Τι καιρό θα κάνει αύριο στην Θεσσαλονίκη;	<code>``get_forecast_weather('thessaloniki', 1)``</code>
Function Calling	What is the weather now in Thessaloniki?	<code>``get_current_weather('thessaloniki')``</code>
RAG	Ποιό είναι το email του κυρίου Γουλιάνα;	Το email του κυρίου Γουλιάνα είναι gouliana@it.teithe.gr.
RAG	What is the duration of studying in semesters?	There are ten (10) academic semesters for the first cycle of studies.

Πίνακας 6: Παραδείγματα από το dataset

5.3 Μέθοδος Αξιολόγησης

Για την αξιολόγηση των μοντέλων και των υπερπαραμέτρων χρησιμοποιήθηκαν 2 μετρικές, η πολύ δημοφιλής μετρική accuracy για τα δεδομένα της function calling και το BLEU (bilingual evaluation understudy) score για την αξιολόγηση της RAG αρχιτεκτονικής.

Τα μοντέλα αξιολογήθηκαν στο function calling ελέγχοντας την έξοδο του μεγάλου γλωσσικού μοντέλου, αν αυτή ήταν ίδια με τον στόχο στο dataset ανέβαζε το score κατά μία σωστή απάντηση, ενώ στις περιπτώσεις όπου υπάρχει η παραμικρή αλλαγή έπαιρνε μηδέν.

Η RAG αρχιτεκτονική αξιολογήθηκε με το BLEU score, το BLEU είναι ένας αλγόριθμος αξιολόγησης αρχικά χρησιμοποιούμενος για την αξιολόγηση της μηχανικής μετάφρασης, ωστόσο χρησιμοποιείται και για την αξιολόγηση της ποιότητας κειμένου παραγόμενο από ένα μεγάλο γλωσσικό μοντέλο, συγκρίνοντας το κείμενο με αναφορές. Το BLEU score χρησιμοποιείται διότι είναι γρήγορο και αντικειμενικό, δεν απαιτεί την ανθρώπινη αξιολόγηση για την οποία θα απαιτούνταν πολύ περισσότερος χρόνος καθώς και θα υπήρχε υποκειμενικότητα λόγω του αξιολογητή. Το BLEU κάνει μία σύγκριση n-grams μεταξύ της πρότασης που παράχθηκε από το μοντέλο και της αναφοράς, υπολογίζεται το precision για 1-grams και 2-grams και στην συνέχεια υπολογίζεται ο γεωμετρικός μέσος αυτών των precisions. Είναι σημαντικό να αναφερθεί ότι συχνά χρησιμοποιούνται και τα BLEU-3 και BLEU-4 όπου ψάχνουν 3-grams και 4-grams αντίστοιχα ωστόσο στο συγκεκριμένο σύνολο δεδομένων οι απαντήσεις είναι πολύ μικρές και η συμπερίληψή τους θα έφερνε το BLEU score συνέχεια κοντά στο 0. Τέλος, στον γεωμετρικό μέσο που βρέθηκε πριν προστίθεται ένας συντελεστής ποινής μήκους (brevity penalty - BP) ώστε να αποφεύγονται οι υπερβολικά μικρές προτάσεις οι οποίες αντιγράφουν μόνο μερικά n-grams. Το BLEU score υπολογίζεται από τον τύπο:

$$BLEU = BP \times \exp\left(\sum_{n=1}^n (w_n \times \log(p_n))\right) \quad (3)$$

Όπου:

- p_n : precision για n-gram
- w_n : βάρος, στην περίπτωση μας $1/N$
- BP: Brevity Penalty

Με το BP να παίρνει τιμή 1 εάν το μήκος της παραγόμενης πρότασης είναι μεγαλύτερο από αυτό της πρότασης που δίνεται ως αναφορά, ενώ ακολουθεί την εξίσωση:

$$BP = e^{(1-\frac{r}{c})} \quad (4)$$

Όταν η παραγόμενη πρόταση έχει μέγεθος μικρότερο ή ίσο αυτής που δίνεται ως αναφορά με:

- C: το μήκος της παραγόμενης πρότασης
- R: το μήκος της reference πρότασης

Για την καλύτερη κατανόηση της μετρικής ακολουθεί το εξής παράδειγμα:

Το μοντέλο παράγαγε την πρόταση “Ο κύριος Γουλιάνας απαντά στο email gouliana@it.teithe.gr” ενώ η πρόταση η οποία δόθηκε ως αναφορά ήταν η “Το email του κυρίου Γουλιάννα είναι gouliana@it.teithe.gr”. Αρχικά θα γίνει tokenization των προτάσεων στα n-grams τους:

- Για $n = 1$ έχουμε:

[‘Ο’, ‘κύριος’, ‘Γουλιάνας’, ‘απαντά’, ‘στο’, ‘email’, ‘gouliana@it.teithe.gr’]

[‘Το’, ‘email’, ‘του’, ‘κυρίου’, ‘Γουλιάννα’, ‘είναι’, ‘gouliana@it.teithe.gr’]

Έχουμε 2 κοινά 1-grams τα “email” και “gouliana@it.teithe.gr” επομένως το precision είναι $2/7 \approx 0.29$

- Για $n = 2$ έχουμε:

['Ο κύριος', 'κύριος Γουλιάνας', 'Γουλιάνας απαντά', 'απαντάει στο', 'στο email', 'email gouliana@it.teithe.gr']

['Το email', 'email του', 'του κυρίου', 'κυρίου Γουλιάνα', 'Γουλιάνα είναι', 'είναι gouliana@it.teithe.gr']

Έχουμε 1 κοινό 2-gram το “email gouliana@it.teithe.gr” επομένως το precision για $n = 2$ είναι $1/6 \approx 0.17$.

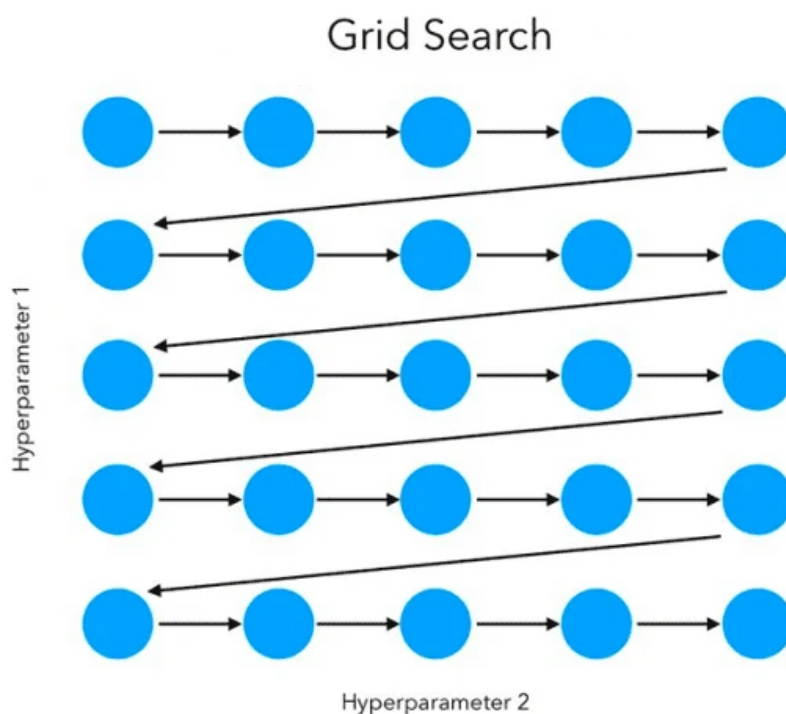
Το μέγεθος και των δύο προτάσεων είναι ίσο, επομένως το BP = 1.

Τελικό BLEU score = $1 \times \exp(1/2 \log(0.29) + 1/2 \log(0.17)) = 0.22$

Για κάθε απάντηση υπολογίζεται το BLEU score και στο τέλος εμφανίζεται ο μέσος όρος για κάθε μοντέλο.

5.4 Αναζήτηση πλέγματος και υπερ παράμετροι

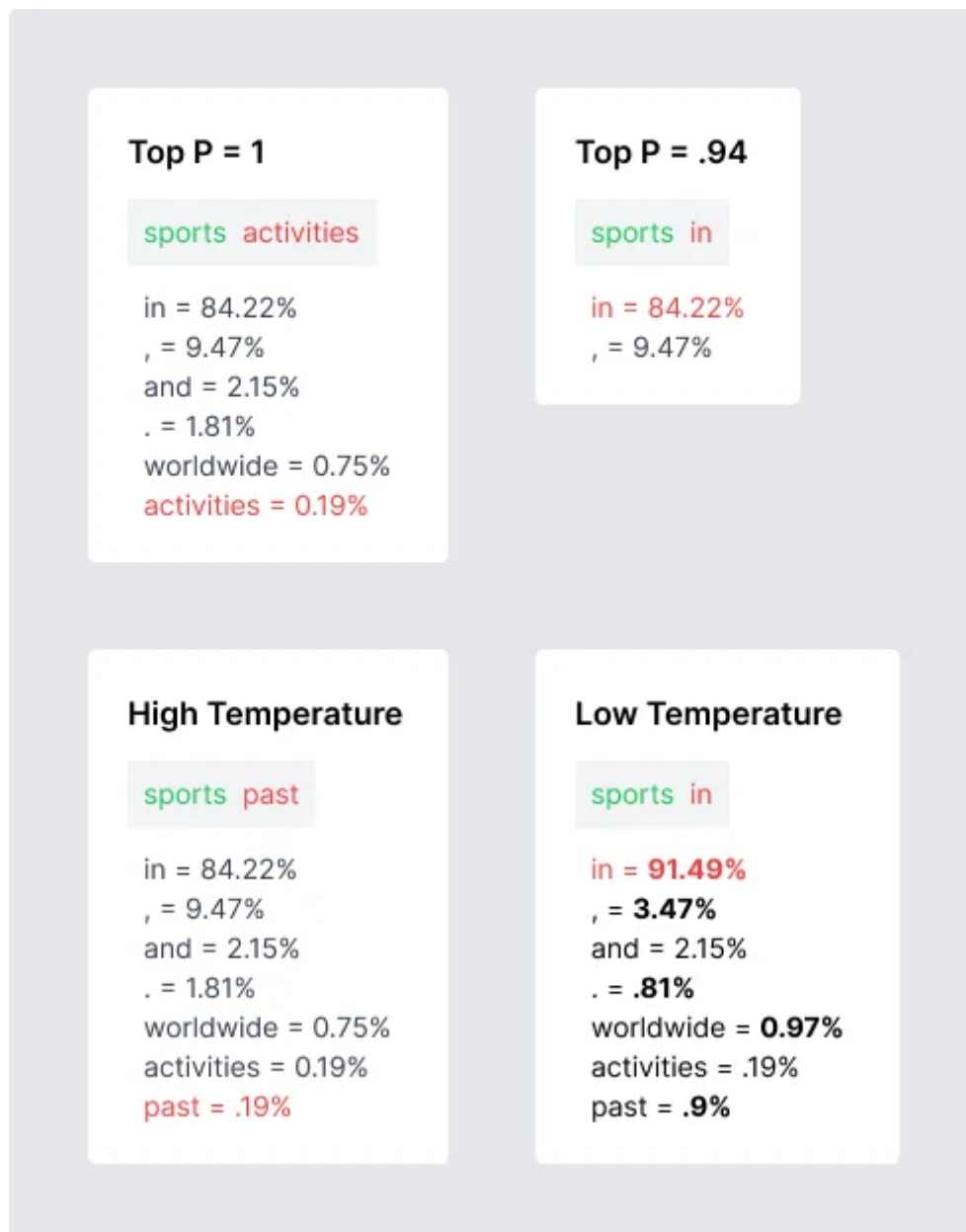
Η αναζήτηση πλέγματος (grid search) αποτελεί μία εξαντλητική μέθοδο εύρεσης των καλύτερων τιμών των υπερπαραμέτρων των μεγάλων γλωσσικών μοντέλων. Η διαδικασία αυτή περιλαμβάνει τον ορισμό συγκεκριμένων τιμών για κάθε μία από τις υπερπαραμέτρους οι οποίες εξετάζονται και έπειτα την αξιολόγηση των μοντέλων σε κάθε έναν από τους πιθανούς συνδυασμούς τιμών των υπερπαραμέτρων. Η αναζήτηση πλέγματος είναι μία εύκολα κατανοητή μέθοδος η οποία λόγω της εξαντλητικής φύσης της καθιστά την εύρεση των βέλτιστων τιμών σίγουρη αλλά απαιτεί μεγάλη υπολογιστική δύναμη.



Σχήμα 36: Οπτικοποίηση της αναζήτησης πλέγματος

Οι υπερ παράμετροι temperature και top_p παίζουν κρίσιμο ρόλο στη ρύθμιση της δημιουργικότητας και της ποικιλομορφίας των απαντήσεων σε ένα μεγάλο γλωσσικό μοντέλο (LLM), ιδιαίτερα κατά τη φάση της δειγματοληψίας. Η temperature ελέγχει το επίπεδο τυχαιότητας στις προβλέψεις του μοντέλου: χαμηλές τιμές οδηγούν σε πιο προβλέψιμες απαντήσεις, ενώ υψηλότερες ενισχύουν τη δημιουργικότητα με μεγαλύτερο ρίσκο μη σχετικών ή ασυνεπών αποτελεσμάτων. Η top_p (ή nucleus sampling) περιορίζει τις πιθανές προβλέψεις μόνο στο μικρότερο σύνολο λέξεων του οποίου το αθροιστικό άθροισμα πιθανοτήτων υπερβαίνει ένα κατώφλι p, αποφεύγοντας έτσι προβλέψεις από το τέλος της κατανομής. Η αναζήτηση των βέλτιστων τιμών για αυτές τις υπερπαραμέτρους είναι σημαντική για την επίτευξη της επιθυμητής ισορροπίας μεταξύ ακρίβειας και δημιουργικότητας.

Για τα πειράματα αξιολόγησης των μοντέλων της εφαρμογής, για κάθε μεγάλο γλωσσικό μοντέλο εξετάστηκαν οι τιμές 0.1, 0.5 και 1 για το temperature και 0.7, 0.8 και 0.9 για το top_p.



Σχήμα 37: Παράδειγμα των Υπερπαραμέτρων top_p και temperature

5.5 Αξιολόγηση των Μεγάλων Γλωσσικών Μοντέλων

Στα πλαίσια των πειραμάτων και μέσω του grid search για κάθε ένα από τα 4 μοντέλα υπολογίστηκαν 18 διαφορετικές τιμές 9 για το dataset των ελληνικών με τα αγγλικά και 9 για το dataset του οποίου οι ελληνικές εγγραφές έχουν μεταφραστεί στα αγγλικά από το seamless4t μοντέλο. Για την ευκολότερη κατανόηση των αποτελεσμάτων σε κάποια γραφήματα έχει επιλεγεί η χρήση του μέσου όρου των αποτελεσμάτων μεταξύ των δύο datasets, ενώ όπου δεν έχει επιλεγεί αυτό αναγράφονται τα datasets ως “type”. Για την δημιουργία των γραφημάτων έγινε χρήση των βιβλιοθηκών matplotlib και της plotly της rython, καθώς και της pandas.plotting για την δημιουργία του πίνακα των αποτελεσμάτων.

Τα αποτελέσματα έδειξαν ξεκάθαρη υπεροχή των fine-tuned στα ελληνικά μοντέλων στο μεικτό dataset όπως και ήταν αναμενόμενο, με αρκετά καλύτερη διαχείριση των εγγραφών στην ελληνική γλώσσα με το Krikri 7B μοντέλο να έχει τα καλύτερα αποτελέσματα τόσο στο function calling όσο και στην αρχιτεκτονική RAG. Ωστόσο, η μετάφραση των δεδομένων στην αγγλική γλώσσα βελτίωσε σημαντικά τις αποδόσεις των αρχικών μοντέλων, και ειδικότερα του Mistral 7B μοντέλου το οποίο είχε την καλύτερη επίδοση τόσο στην κλήση συναρτήσεων όσο και στην αρχιτεκτονική RAG.

Model	Temperature	Top-p	Accuracy	BLEU
mistral	0.1	0.7	0.70	0.38
meltemi	0.1	0.7	0.70	0.41
llama	0.1	0.7	0.80	0.39
krikri	0.1	0.7	0.80	0.44
mistral	0.1	0.8	0.70	0.38
meltemi	0.1	0.8	0.70	0.40
llama	0.1	0.8	0.80	0.39
krikri	0.1	0.8	0.80	0.45
mistral	0.1	0.9	0.70	0.40
meltemi	0.1	0.9	0.70	0.41
llama	0.1	0.9	0.80	0.41
krikri	0.1	0.9	0.80	0.45
mistral	0.5	0.7	0.70	0.33
meltemi	0.5	0.7	0.60	0.38
llama	0.5	0.7	0.80	0.37
krikri	0.5	0.7	0.80	0.40
mistral	0.5	0.8	0.70	0.37
meltemi	0.5	0.8	0.70	0.39
llama	0.5	0.8	0.80	0.38
krikri	0.5	0.8	0.80	0.41
mistral	0.5	0.9	0.70	0.40
meltemi	0.5	0.9	0.70	0.41
llama	0.5	0.9	0.80	0.41
krikri	0.5	0.9	0.80	0.44
mistral	1.0	0.7	0.60	0.27
meltemi	1.0	0.7	0.60	0.29
llama	1.0	0.7	0.80	0.33
krikri	1.0	0.7	0.80	0.34
mistral	1.0	0.8	0.70	0.29
meltemi	1.0	0.8	0.60	0.33
llama	1.0	0.8	0.80	0.37
krikri	1.0	0.8	0.80	0.39
mistral	1.0	0.9	0.70	0.35
meltemi	1.0	0.9	0.70	0.39
llama	1.0	0.9	0.80	0.41
krikri	1.0	0.9	0.80	0.43

Σχήμα 38: Αποτελέσματα Grid Search σε δεδομένα σε Ελληνικά και Αγγλικά

Model	Temperature	Top-p	Accuracy	BLEU
mistral	0.1	0.7	0.90	0.41
meltemi	0.1	0.7	0.70	0.27
llama	0.1	0.7	0.90	0.40
krikri	0.1	0.7	0.90	0.41
mistral	0.1	0.8	0.90	0.43
meltemi	0.1	0.8	0.70	0.29
llama	0.1	0.8	0.90	0.41
krikri	0.1	0.8	0.90	0.41
mistral	0.1	0.9	0.90	0.44
meltemi	0.1	0.9	0.70	0.33
llama	0.1	0.9	0.90	0.42
krikri	0.1	0.9	0.90	0.43
mistral	0.5	0.7	0.90	0.35
meltemi	0.5	0.7	0.60	0.28
llama	0.5	0.7	0.90	0.33
krikri	0.5	0.7	0.90	0.35
mistral	0.5	0.8	0.90	0.39
meltemi	0.5	0.8	0.70	0.29
llama	0.5	0.8	0.90	0.36
krikri	0.5	0.8	0.90	0.37
mistral	0.5	0.9	0.90	0.43
meltemi	0.5	0.9	0.70	0.30
llama	0.5	0.9	0.90	0.40
krikri	0.5	0.9	0.90	0.40
mistral	1.0	0.7	0.80	0.32
meltemi	1.0	0.7	0.60	0.22
llama	1.0	0.7	0.90	0.30
krikri	1.0	0.7	0.90	0.33
mistral	1.0	0.8	0.90	0.35
meltemi	1.0	0.8	0.70	0.26
llama	1.0	0.8	0.90	0.33
krikri	1.0	0.8	0.90	0.34
mistral	1.0	0.9	0.90	0.40
meltemi	1.0	0.9	0.70	0.27
llama	1.0	0.9	0.90	0.39
krikri	1.0	0.9	0.90	0.40

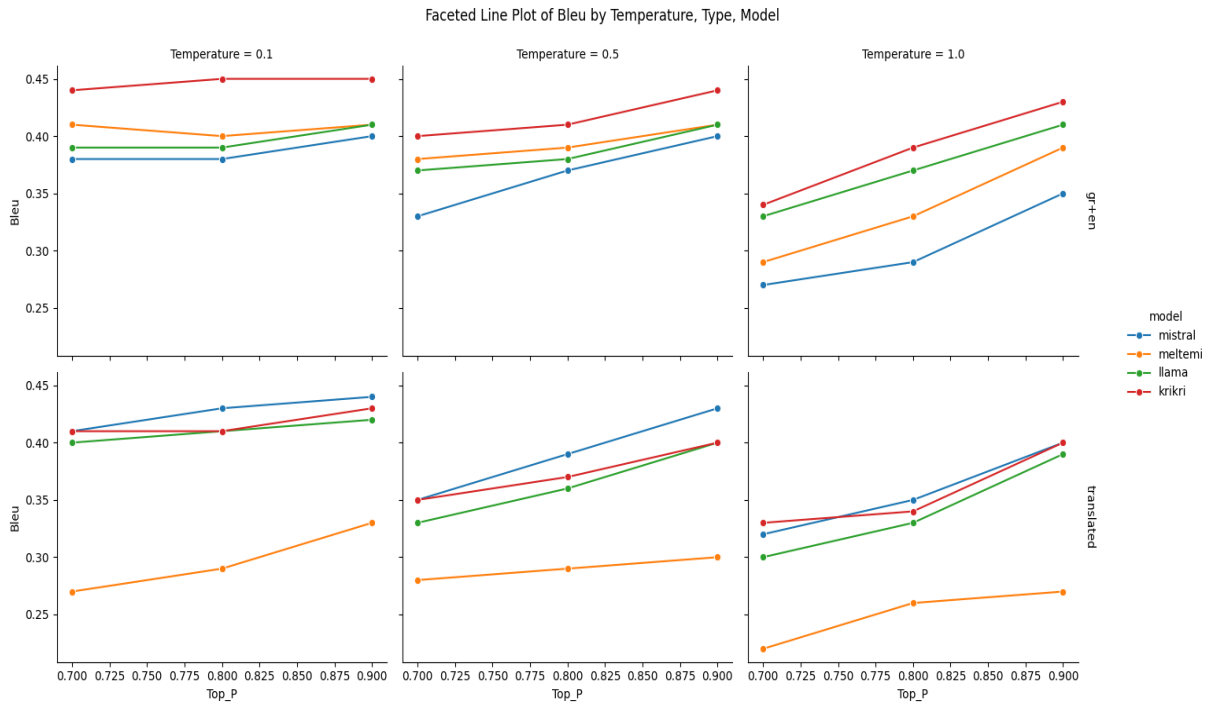
Σχήμα 39: Αποτελέσματα Grid Search σε μεταφρασμένα δεδομένα

Τα αποτελέσματα αναδεικνύουν το Mistral 7B Instruct μοντέλο με τιμές 0.1 για το temperature και 0.9 για το top_p ως το καλύτερο σε απόδοση. Το Krikri μοντέλο στα bilingual δεδομένα κατάφερε να ξεπεράσει το 0.44 το οποίο έκανε το Mistral μοντέλο στα μεικτά δεδομένα ωστόσο αυτό συνδυαζόταν με 0.8 στο accuracy για το function calling καθιστώντας το χειρότερο συνολικά.

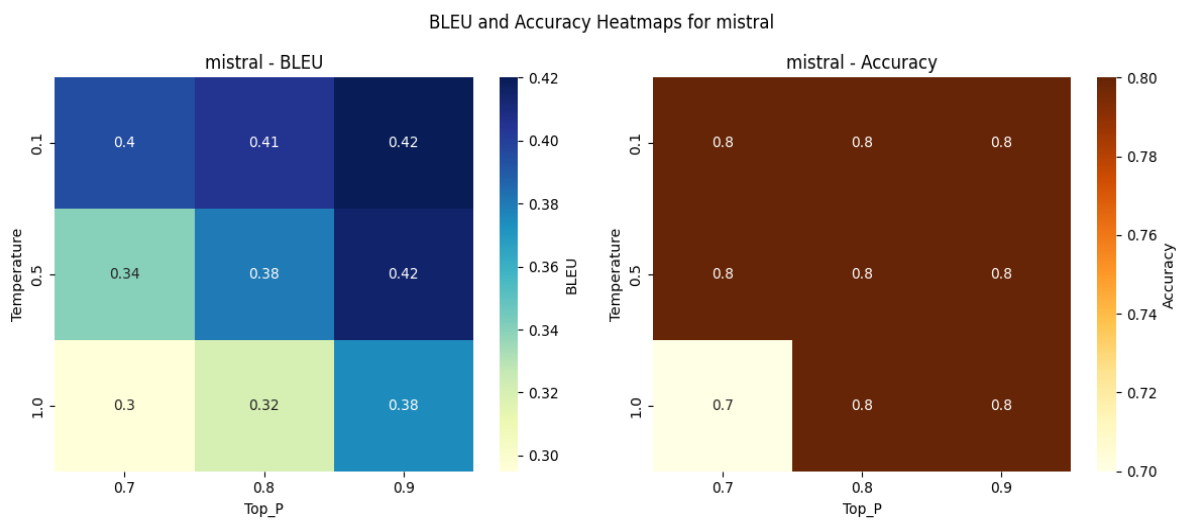
Ενδιαφέρον παρουσιάζει και το γεγονός ότι παρατηρείται γενικότερη βελτίωση των αποδόσεων των μοντέλων και στα δύο datasets όταν αυτά γίνονται περισσότερο ντετερμινιστικά, όταν δηλαδή το temperature μειώνεται και το top_p αυξάνεται. Αυτό το φαινόμενο μπορεί να εξηγηθεί εύκολα λόγω της φύσης του dataset το οποίο επιβραβεύει την ακολούθηση των κανόνων και την πιστότητα στα δεδομένα που δίνονται ως reference καθώς και την τιμωρία ακόμα και των μικρότερων αλλαγών στη μορφή και τη δομή των απαντήσεων. Το φαινόμενο αυτό είναι εμφανές στα παρακάτω διαγράμματα, τόσο για το κάθε μοντέλο ξεχωριστά, με τις υψηλότερες τιμές να εμφανίζονται όσο αυτό γίνεται ντετερμινιστικό, όσο και συνολικά ως τάση σε όλα τα μοντέλα.

Τέλος, είναι σημαντικό να αναφερθεί πως σε κάθε περίπτωση η χρήση κανενός από τα υπόλοιπα μοντέλα δε θα ήταν λανθασμένη καθώς οι διαφορές και οι αποκλίσεις στην απόδοση δεν είναι μεγάλες, όπως και εξίσου σωστή θα ήταν η χρήση ενός pipeline χωρίς μετάφραση, με το krikri μοντέλο σε περίπτωση όπου η ταχύτητα απόκρισης του συστήματος ήταν σημαντικότερη από την απόδοση.

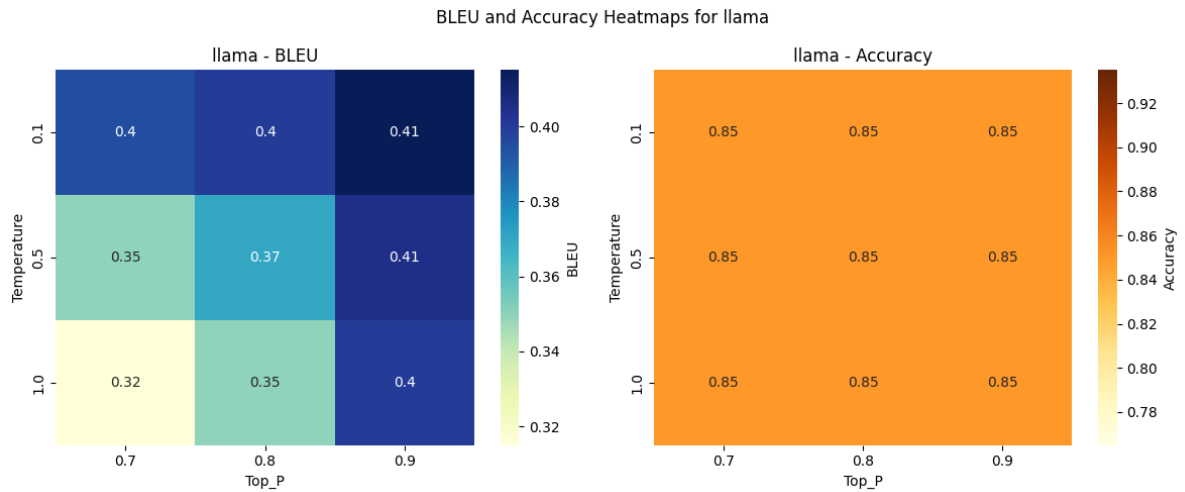
Ακολουθούν τα διαγράμματα για την τάση των μοντέλων με τις αλλαγές στα top_p και temperature συνολικά και για το καθένα ξεχωριστά.



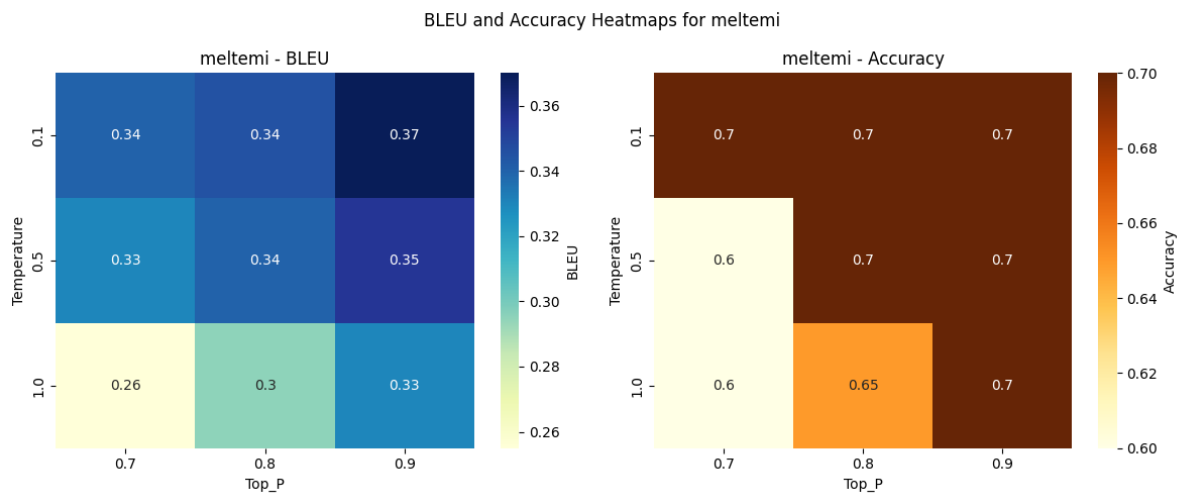
Σχήμα 40: Διάγραμμα Bleu Score προς Top_p, Temperature και τύπου dataset



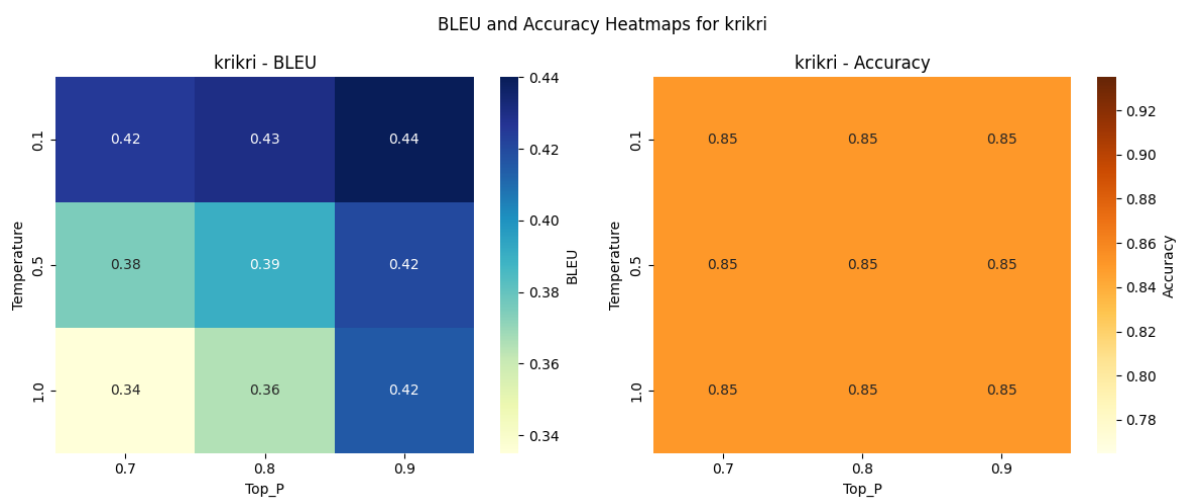
Σχήμα 41: Heatmap Μέσου όρου τιμών του Mistral 7B μοντέλου



Σχήμα 42: Heatmap μέσου όρου τιμών LLama μοντέλου



Σχήμα 43: Heatmap μέσου όρου τιμών Meltemi μοντέλου



Σχήμα 44: Heatmap μέσου όρου τιμών Krikri μοντέλου

5.6 Σύνοψη

Με την δημιουργία ενός dataset ερωτήσεων και απαντήσεων κατάλληλης μορφής ώστε να μπορούν να αξιολογηθούν τα μεγάλα γλωσσικά μοντέλα και την υλοποίηση μίας αναζήτησης πλέγματος για τις βέλτιστες τιμές των υπερπαραμέτρων, ελέγχθηκε η επιρροή την οποία έχει το fine-tuning και οι υπερπαραμέτροι temperature και top_p στα μεγάλα γλωσσικά μοντέλα. Σε εφαρμογές ανάκτησης πληροφορίας και κλήσης συναρτήσεων αποδείχτηκε πως τα περισσότερα ντετερμινιστικά μοντέλα αποδίδουν καλύτερα. Τέλος, παρά την βελτίωση της απόδοσης των μοντέλων μετά από το fine-tuning στην ελληνική γλώσσα τα αποτελέσματα όταν το dataset περιλαμβάνει μόνο αγγλικά δείχνει υπεροχή των αρχικών μεγάλων γλωσσικών μοντέλων.

Κεφάλαιο 6ο: Συμπεράσματα και Προτάσεις Βελτίωσης

6.1 Συμπεράσματα

Η χρήση μεγάλων γλωσσικών μοντέλων μικρού μεγέθους συγκριτικά με τα καλύτερα σε απόδοση μοντέλα με κατάλληλο τρόπο μέσω της αρχιτεκτονικής Retrieval Augmented Generation και την δημιουργία σωστών prompts με few-shot examples μπορεί να δημιουργήσει έναν βοηθό με πολύ ικανοποιητική απόδοση, ικανό να βοηθά τους φοιτητές στις καθημερινές τους δραστηριότητες.

Για την δημιουργία μίας τέτοια εφαρμογής, χρειάστηκε αρκετή μελέτη και κατανόηση του θεωρητικού υποβάθρου σε έναν κλάδο συνεχώς εξελισσόμενο με καινούργιες τεχνολογίες, μοντέλα και αρχιτεκτονικές να αναδύονται συνεχώς στην επιφάνεια. Ο κλάδος της τεχνητής νοημοσύνης είναι ίσως ο ταχύτερα εξελισσόμενος αυτή τη στιγμή κάνοντας βιβλιογραφία που εκδόθηκε 2 χρόνια πριν, πολλές φορές να φαντάζει απαρχαιωμένη. Λόγω της ταχύτατης αυτής εξέλιξης, οι απαιτήσεις που έχουμε από αυτά τα συστήματα συνεχώς αυξάνονται, ωστόσο τη δεδομένη στιγμή τα αποτελέσματα της εφαρμογής είναι πολύ ικανοποιητικά.

Η τάση της βιομηχανίας και της έρευνας τον τελευταίο χρόνο έχει στραφεί στην δημιουργία ισχυρών αλλά μικρών σε μέγεθος μοντέλων, ικανών να τρέχουν σε περισσότερα συστήματα με σκοπό την αύξηση της δημοφιλίας και της χρήσης εφαρμογών που χρησιμοποιούν μεγάλα γλωσσικά μοντέλα, μέσα σε αυτό το κλίμα η διπλωματική εργασία και εφαρμογή που την συνοδεύει αποτελεί ένα βήμα προς την επίκαιρη κατεύθυνση.

6.2 Προτάσεις για βελτίωση και αδυναμίες

Μελλοντικά θα ήταν χρήσιμη η δημιουργία μίας διεπαφής χρήστη αποκλειστικά στοχευμένη στην παρούσα εφαρμογή, είτε με κάποια εφαρμογή για κινητά τηλέφωνα είτε κάποια ιστοσελίδα με σκοπό την εξατομικευμένη για την εφαρμογή διεπαφή.

Σημαντική βελτίωση αποτελεί η χρήση και η σύγκριση με καινούργια μοντέλα, λίγο μεγαλύτερα και λίγο μικρότερα για την διερεύνηση της επίδρασης του μεγέθους των μεγάλων γλωσσικών μοντέλων.

Για την χρήση της εφαρμογής και την δημοσίευση της για χρήση από τους φοιτητές ιδανική θα ήταν μία cloud υποδομή με κάποιο σύστημα διαχείρισης όπως το kubernetes ώστε να τρέχει γρήγορα και χωρίς πρόβλημα.

Τέλος, η προσθήκη περισσότερων συναρτήσεων και εργαλείων προς χρήση για το μεγάλο γλωσσικό μοντέλο είτε με περισσότερα και διαφορετικά APIs είτε με την σύνδεση με υπηρεσιών της σχολής θα μπορούσε να επεκτείνει εύκολα και δραστικά τις δυνατότητες της εφαρμογής.

6.3 Συζήτηση

Τα μεγάλα γλωσσικά μοντέλα αποτελούν το μέλλον της επιστήμης των υπολογιστών διευκολύνοντας την ζωή των χρηστών με την απλοποίηση ενεργειών που διαφορετικά θα έπρεπε να κάνουν χειροκίνητα ενώ πλέον μπορούν να ζητήσουν από την μηχανή να κάνει για αυτούς όσο εκείνοι επενδύουν τον χρόνο τους διαφορετικά. Η δυνατότητες χρήσης μεγάλων γλωσσικών μοντέλων σε μικρούς υπολογιστές με ικανοποιητικό βαθμό εξαιτίας της αρχιτεκτονικής RAG και την δημιουργία σωστών prompts και παραδειγμάτων έχει κάνει προσιτά τα μεγάλα γλωσσικά μοντέλα στο κοινό. Εφαρμογές όπως αυτή αποτελούν πολύ χρήσιμο εργαλείο τόσο για τους χρήστες όσο και για τους δημιουργούς εφαρμογών οι οποίοι πλέον δε θα ανησυχούν για την δυσκολία εύρεσης πληροφοριών μέσα σε μία ιστοσελίδα, την σειρά την οποία θα πρέπει να εμφανίζεται το περιεχόμενο στο χρήστη κ.ο.κ. καθώς ο χρήστης θα μπορεί να συνομιλεί με το σύστημα και να λαμβάνει πίσω την χρήσιμη για αυτόν πληροφορία. Τα μεγάλα γλωσσικά μοντέλα τα οποία δεν διαθέτουν τεράστια υπολογιστική δυνατότητα ή πολύ μεγάλο αριθμό δεδομένων εκπαίδευσης μπορεί κατά την χρήση τους σε γενικά πεδία χωρίς κάποια βοήθεια να μην αποδίδουν καλά, ωστόσο με την χρήση της αρχιτεκτονικής RAG και την δυνατότητα χρήσης εργαλείων τα αποτελέσματα είναι πολύ ικανοποιητικά. Τέλος, η αξιολόγηση των μεγάλων γλωσσικών μοντέλων

Αρχιτεκτονική της Εφαρμογής και Υλοποίηση

απαιτεί την πλήρη κατανόηση της λειτουργίας τους, την δημιουργία κατάλληλων συνόλων δεδομένων αλλά και συστήματα με μεγάλες υπολογιστικές ικανότητες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ο τρόπος γραφής των βιβλιογραφικών αναφορών γίνεται σύμφωνα με τα παρακάτω παραδείγματα (IEEE style):

Βιβλία

- [1] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv preprint arXiv:2005.11401*, 2020. [Online]. Available: <https://arxiv.org/abs/2005.11401>.
- [2] A. Singh *et al.*, "Agentic RAG Systems for Improving Adaptability and Performance in AI-Driven Information Retrieval" (December 06, 2024), Available at SSRN: <https://ssrn.com/abstract=5188363>
- [3] M. Lewis *et al.* "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", Facebook AI 2019, [Online]. Available: <https://arxiv.org/abs/1910.13461>
- [4] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", *Journal of Machine Learning Research* 21 (2020), Available: <https://jmlr.org/papers/volume21/20-074/20-074.pdf>
- [5] X. Wang *et al.*, "Searching for Best Practices in Retrieval-Augmented Generation", School of Computer Science Fudan University (2024), [Online]. Available: <https://arxiv.org/abs/2407.01219>
- [6] R. Nogueira *et al.*, "Document Ranking with a Pretrained Sequence-to-Sequence Model", In *Findings of the Association for Computer Linguistics: EMNLP 2020*, [Online]. Available: <https://aclanthology.org/2020.findings-emnlp.63/>
- [7] A. Vaswani *et al.*, "Attention is All You Need", 2017, [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [8] T. Guo *et al.*, "Large Language Model based Multi-Agents: A Survey of Progress and Challenges", *arXiv preprint arXiv:2402.01680*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.01680>
- [9] A. Singh *et al.*, "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG", *arXiv preprint arXiv:2501.09136*, 2025. [Online]. Available: <https://arxiv.org/abs/2501.09136>
- [10] T. M. Mitchell, *Machine Learning*, vol. 1, no. 9. McGraw-hill New York, 1997.
- [11] A. Karthikeyan *et al.*, "Artificial Intelligence: Machine Learning for Chemical Sciences", *Journal of Chemical Sciences*, 2022. [Online]. Available: https://www.researchgate.net/publication/357234810_Artificial_intelligence_machine_learning_for_chemical_sciences
- [12] J. Devlin *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv preprint arXiv:1810.04805*, 2018, [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [13] Y. Liu *et al.*, "A Survey of Visual Transformers", in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 6, pp. 7478-7498, doi: 10.1109/TNNLS.2022.3227717
- [14] Q. Wen *et al.*, "Transformers in Time Series: A Survey", *arXiv preprint arXiv:2202.07125*, 2022. [Online]. Available: <https://arxiv.org/abs/2202.07125>

- [15] W. Li et al., “Approximate Nearest Neighbor Search on High Dimensional Data - Experiments, Analyses, and Improvement”, in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 8, pp 1475-1488, 1 Aug. 2020, doi: 10.1109/TKDE.2019.2909204
- [16] L. Barrault et al., “SeamlessM4T: Massively Multilingual & Multimodal Machine Translation”, arXiv: 2308.11596, 2023. [Online]. Available: <https://arxiv.org/abs/2308.11596>
- [17] R. C. Martin, “Clean Architecture: A Craftsman’s Guide to Software Structure and Design”, Prentice Hall, 2017
- [18] A. Q. Jiang et al., “Mistral 7B”, arXiv: 2310.06825, 2023. [Online]. Available: <https://arxiv.org/abs/2310.06825>
- [19] L. Voukoutis et al., “Meltemi: The first open Large Language Model for Greek”, eprint arXiv: 2407.20743, 2024. [Online]. Available: <https://arxiv.org/abs/2407.20743>
- [20] A. Pal et al., “MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering”, Proceedings of the Conference on Health, Inference, and Learning in Proceedings of Machine Learning Research 174:248-260. [Online]. Available: <https://proceedings.mlr.press/v174/pal22a.html>
- [21] L. Bandarkar et al., “The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants”, arXiv: 2308.16884, 2023. [Online]. Available: <https://arxiv.org/abs/2308.16884v2>
- [22] R. Zellers et al., “HellaSwag: Can a Machine Really Finish Your Sentence?”, arXiv: 1905.07830, 2019. [Online]. Available: <https://arxiv.org/abs/1905.07830v1>
- [23] P. Clark et al., “Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge”, arXiv: 1803.05457, 2018. [Online]. Available: <https://arxiv.org/abs/1803.05457v1>
- [24] S. Lin et al., “TruthfulQA: Measuring How Models Mimic Human Falsehoods”, arXiv:2109.07985, 2021. [Online]. Available: <https://arxiv.org/abs/2109.07958v2>
- [25] D. Hendrycks et al., “Measuring massive Multitask Language Understanding”, arXiv: 2009.03300, 2020. [Online]. Available: <https://arxiv.org/abs/2009.03300v3>
- [26] Meta AI blog, “Introducing Llama 3.1: Our most capable models to date”, 2024, [Online]. Available: <https://ai.meta.com/blog/meta-llama-3-1/>
- [27] J. Zhou et al., “Instruction-Following Evaluation for Large Language Models”, arXiv: 2311.07911, 2023. [Online]. Available: <https://arxiv.org/abs/2311.07911>
- [28] M. Chen et al., “Evaluating Large Language Models Trained on Code”, arXiv: 2107.03374, 2021. [Online]. Available: <https://arxiv.org/abs/2107.03374v2>
- [29] J. Liu et al., “Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation”, Thirty-seventh Conference on Neural Information Processing Systems, 2023. [Online]. Available: <https://openreview.net/forum?id=1qvx610Cu7>
- [30] K. Cobbe et al., “Training Verifiers to Solve Math Word Problems”, arXiv:2110.14168, 2021. [Online]. Available: <https://arxiv.org/abs/2110.14168v2>
- [31] D. Hendrycks et al., “Measuring Mathematical Problem Solving With the MATH Dataset”, arXiv:2103.03874, 2021. [Online]. Available: <https://arxiv.org/abs/2103.03874v2>

- [32] D. Rein et al., “A Graduate-Level Google-Proof Q&A Benchmark”, arXiv:2311.12022, 2023. [Online]. Available: <https://arxiv.org/abs/2311.12022v1>
- [33] H. Mao et. al., “BFCL V3 Multi-Turn & Multi-Step Function Calling Evaluation”, Berkeley.edu Blog, 2024. [Online]. Available: https://gorilla.cs.berkeley.edu/blogs/13_bfcl_v3_multi_turn.html
- [34] Nexusflow.ai team, “NexusRaven-V2: Surpassing GPT-4 for Zero-Shot Function Calling”, Online, 2023. Available: <https://nexusflow.ai/blogs/ravenv2>
- [35] U. Shaham et al., “ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding”, arXiv:2305.14196, 2023. [Online]. Available: <https://arxiv.org/abs/2305.14196>
- [36] X. Zhang et al., “InfiniteBench: Extending Long Context Evaluation Beyond 100k Tokens”, arXiv:2402.13718, 2024. [Online]. Available: <https://arxiv.org/abs/2402.13718>
- [37] H. Wang et al., “Multimodal Needle in a Haystack: Benchmarking Long-Context Capability of Multimodal Large Language Models”, arXiv:2406.11230, 2024. [Online]. Available: <https://arxiv.org/abs/2406.11230>
- [38] F. Shi et al., “Language Models are Multilingual Chain-of-Thought Reasoners”, arXiv:2210.03057, 2022. [Online]. Available: <https://arxiv.org/abs/2210.03057v1>
- [39] R. Nallapati et al., “Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond”, eprint arXiv:1602.06023, 2016. [Online]. Available: <https://arxiv.org/abs/1602.06023v5>
- [40] G. Bai et al., “MT-Bench-101: A Fine-Grained Benchmark for Evaluating Large Language Models in Multi-Turn Dialogues”, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 7421-7454, Association for Computational Linguistics, 2024. [Online]. Available: <https://arxiv.org/abs/2402.14762>
- [41] gTTS, google Text-To-Speech, “gTTS Documentation”, Online, Available: <https://pypi.org/project/gTTS/>
- [42] Wang et al., “Multilingual E5 Text Embeddings: A Technical Report”, arXiv preprint arXiv:2402.05672, 2024. [Online]. Available: <https://arxiv.org/pdf/2402.05672>
- [43] K. Enevoldsen et al., “MMTEB: Massive Multilingual Text Embedding Benchmark”, eprint arXiv:2502.13595, 2025. [Online]. Available: <https://arxiv.org/abs/2502.13595>
- [44] I. Kavathekar et al., “Small Models, Big Tasks: An Exploratory Empirical Study on Small Language Models for Function Calling”, eprint arXiv:2504.19277, 2025. [Online]. Available: <https://arxiv.org/abs/2504.19277>