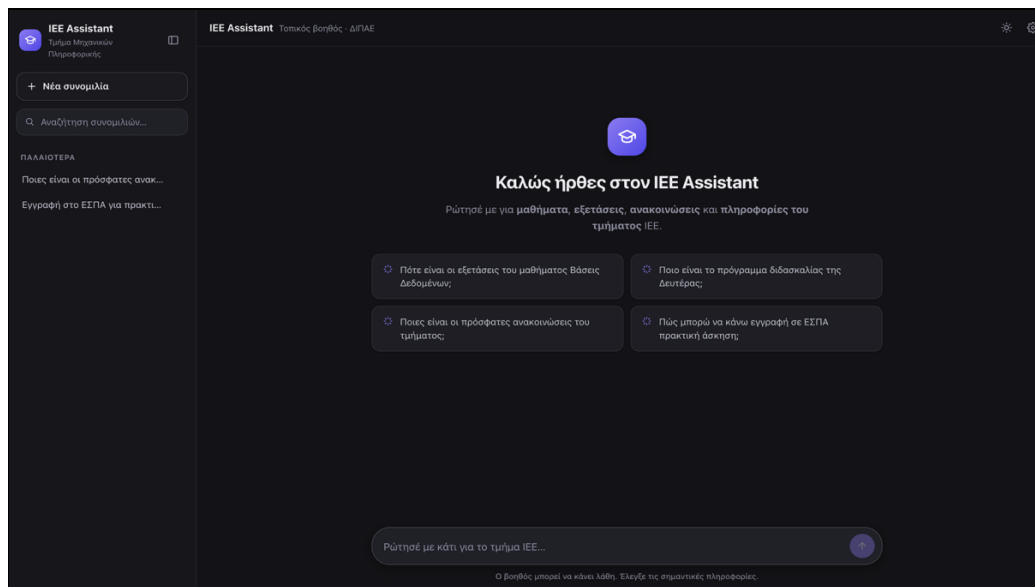


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σχεδιασμός και υλοποίηση Συστήματος RAG για την
αυτοματοποίηση Γραμματειακών Διαδικασιών στην
τριτοβάθμια εκπαίδευση»



Του φοιτητή
Παντελεήμων Πολίτη
Αρ. Μητρώου: 2021137

Επιβλέπων
Ονοματεπώνυμο Καλλιόπη Κράβαρη
Βαθμίδα Επικ. Καθηγήτρια

Ημερομηνία 03/09/2026

Τίτλος Δ.Ε. Σχεδιασμός και Υλοποίηση Συστήματος RAG για την Αυτοματοποίηση Γραμματειακών Διαδικασιών στην Τριτοβάθμια Εκπαίδευση
Κωδικός Δ.Ε. 26136
Ονοματεπώνυμο φοιτητή Παντελεήμων Πολίτης
Ονοματεπώνυμο εισηγητή Καλλιόπη Κράβαρη
Ημερομηνία ανάληψης Δ.Ε. 05/03/2026
Ημερομηνία περάτωσης Δ.Ε. 03/09/2026

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 Παντελεήμων Πολίτη 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 οποιονδήποτε 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.

Πρόλογος

Η επιλογή του θέματος της συγκεκριμένης διπλωματικής εργασίας δεν ήταν τυχαία. Κατά τη διάρκεια των σπουδών μου άρχισε να με ενδιαφέρει ιδιαίτερα ο τομέας της Μηχανικής Μάθησης και γενικότερα οι δυνατότητες που προσφέρουν τα σύγχρονα συστήματα τεχνητής νοημοσύνης. Ήθελα, επομένως, η διπλωματική μου εργασία να σχετίζεται με αυτόν τον χώρο και παράλληλα να δίνει λύση σε ένα πρόβλημα. Ένα τέτοιο πρόβλημα ήταν ο μεγάλος φόρτος εργασίας της γραμματείας και οι καθυστερήσεις που μπορεί να προκύψουν στην εξυπηρέτηση των φοιτητών. Μέσα από την εκπόνηση της εργασίας μου είχα την ευκαιρία να εμβαθύνω περισσότερο στην μηχανική μάθηση και ειδικότερα στα συστήματα RAG. Παράλληλα, απέκτησα εμπειρία στον σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός συστήματος τεχνητής νοημοσύνης, καθώς και στην αντιμετώπιση των προκλήσεων που προέκυψαν κατά την ανάπτυξή του.

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται τον σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός ελληνόγλωσσου συστήματος ανάκτησης με ενισχυμένη παραγωγή (Retrieval-Augmented Generation – RAG) με στόχο τη γραμματειακή υποστήριξη ακαδημαϊκού τμήματος. Ο ρόλος του συστήματος είναι να δίνει σαφείς απαντήσεις για ερωτήματα που αφορούν διοικητικά θέματα και γενικές απορίες για το τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων. Οι απαντήσεις βασίζονται αποκλειστικά στα θεσμικά έγγραφα του τμήματος και η εκτέλεσή του είναι τοπική χωρίς καμία κλήση σε εξωτερική υπηρεσία κατά τη λειτουργία του.

Η υλοποίηση βασίζεται στον συνδυασμό διαφορετικών τεχνικών ανάκτησης και επεξεργασίας του κειμένου, με στόχο την ακριβέστερη ερμηνεία του ερωτήματος του χρήστη και την επιλογή των κατάλληλων αποσπασμάτων για την παραγωγή μιας τεκμηριωμένης τελικής απάντησης. Παράλληλα, ο σχεδιασμός ντετερμινιστικών μηχανισμών και μηχανισμών ελέγχου βελτίωσε την ικανότητα του συστήματος να διαχειρίζεται διαφορετικές κατηγορίες ερωτημάτων, καθώς και αιτήματα που βρίσκονται εκτός του γνωστικού του αντικειμένου.

Η αξιολόγηση που πραγματοποιήθηκε είχε ως βασικό σκοπό να εξετάσει την ευρωστία του συστήματος σε διαφορετικές πτυχές και λειτουργίες του. Τα αποτελέσματα έδειξαν υψηλή απόδοση στο στάδιο της ανάκτησης, ενώ μεγαλύτερο περιθώριο βελτίωσης εντοπίστηκε στη σύνθεση της τελικής απάντησης και στην ορθή συσχέτιση των διαθέσιμων πληροφοριών. Βασική διαπίστωση της παρούσας εργασίας ήταν ότι η μορφή διατύπωσης του ερωτήματος επηρεάζει την απόδοση του συστήματος και ότι οι επιμέρους βελτιστοποιήσεις στα ενδιάμεσα στάδια επεξεργασίας δεν συνεπάγονται απαραίτητα αντίστοιχη βελτίωση στο τελικό αποτέλεσμα.

«Design and Implementation of a RAG System for Automating University Secretariat Workflows»

«Panteleimon Politis»

Abstract

This thesis addresses the design, implementation, and evaluation of a Greek-language Retrieval-Augmented Generation (RAG) system aimed at providing administrative support to an academic department. The system's role is to provide clear answers to questions regarding administrative matters and general inquiries about the Department of Information and Electronic Engineering. The answers are based exclusively on the department's institutional documents, and the system operates locally without making any calls to external services during its operation.

The implementation relies on a combination of different text retrieval and processing techniques, with the goal of more accurately interpreting the user's query and selecting the appropriate excerpts to generate a well-documented final answer. At the same time, the design of various deterministic and control mechanisms improved the system's ability to handle different categories of queries, as well as requests that fall outside its domain of expertise.

The primary purpose of the evaluation was to examine the system's robustness across its various aspects and functions. The results showed high performance during the retrieval stage, while the greatest room for improvement was identified in the formulation of the final answer and the proper correlation of available information. A key finding of this study was that the way the query is phrased affects the system's performance and that individual optimizations in the intermediate processing stages do not necessarily lead to a corresponding improvement in the final result.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την καθηγήτριά μου, Καλλιόπη Κράβαρη, για την καθοδήγηση και την υποστήριξή της κατά την εκπόνηση της διπλωματικής εργασίας. Επίσης, ευχαριστώ την οικογένεια και τους φίλους μου για τη συνεχή ψυχολογική στήριξη και συμπαράστασή τους σε όλη αυτή την πορεία.

Περιεχόμενα

Πίνακας περιεχομένων

Πρόλογος	iv
Περίληψη	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Σχημάτων	xii
Κατάλογος Πινάκων	xiii
Συντομογραφίες	xiv
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Εισαγωγή	1
1.2 Στόχος και ερευνητικά ερωτήματα	1
1.3 Δομή της διπλωματικής εργασίας	2
Κεφάλαιο 2ο: Θεωρητικό υπόβαθρο	3
2.1 Εισαγωγή	3
2.2 Το πρόβλημα της πρόσβασης σε θεσμική πληροφορία	3
2.2.1 Η φύση του προβλήματος	3
2.2.2 Υπάρχουσες προσεγγίσεις	4
2.2.3 Προστασία δεδομένων και τοπική επεξεργασία	4
2.3 Γλωσσική επεξεργασία και προεπεξεργασία κειμένου	5
2.3.1 Τεμαχισμός σε λεκτικές μονάδες	5
2.3.2 Κανονικοποίηση και Unicode	5
2.3.3 Μορφολογία, στελέχωση και λημματοποίηση	6
2.3.4 Ιδιαιτερότητες της ελληνικής	7
2.3.5 Προσεγγιστική αντιστοίχιση συμβολοσειρών	8
2.4 Αναπαραστάσεις κειμένου	8
2.4.1 Λεξικές αναπαραστάσεις	8
2.4.2 Πυκνές αναπαραστάσεις	10
2.5 Μεγάλα γλωσσικά μοντέλα	11
2.5.1 Ο μετασχηματιστής	11
2.5.2 Προεκπαίδευση και προσαρμογή	12
2.5.3 Παραγωγή κειμένου και παράμετροι αποκωδικοποίησης	12
2.5.4 Κβαντισμός και απαιτήσεις εκτέλεσης	14

2.5.5	Παραισθήσεις	14
2.6	<i>Ανάκτηση πληροφορίας</i>	15
2.6.1	Αραιή και πυκνή ανάκτηση	15
2.6.2	Προσεγγιστική αναζήτηση και βάσεις διανυσμάτων	16
2.6.3	Υβριδική ανάκτηση και σύντηξη	17
2.6.4	Επανακατάταξη	17
2.6.5	Μετρικές αξιολόγησης	18
2.7	<i>Ανάκτηση με ενισχυμένη παραγωγή</i>	20
2.7.1	Το σκεπτικό και η αρχιτεκτονική	20
2.7.2	Τεμαχισμός εγγράφων	21
2.7.3	Θεμελίωση, παραπομπές και άρνηση	22
2.7.4	Αποτίμηση συστημάτων ανάκτησης με ενισχυμένη παραγωγή	23
2.8	<i>Σχετικές εργασίες</i>	24
2.8.1	Συστήματα απάντησης ερωτημάτων σε ακαδημαϊκό περιβάλλον	24
2.8.2	Ανάκτηση με ενισχυμένη παραγωγή σε θεσμικά και νομικά σώματα κειμένων	24
Κεφάλαιο 3ο:	Τεχνολογίες και εργαλεία υλοποίησης	26
3.1	<i>Εισαγωγή</i>	26
3.2	<i>Σχεδιαστικοί περιορισμοί</i>	26
3.2.1	Τοπική εκτέλεση χωρίς υπηρεσία νέφους	26
3.2.2	Περιορισμός διαθέσιμης μνήμης	26
3.3	<i>Γλώσσες προγραμματισμού</i>	27
3.3.1	Python	27
3.3.2	TypeScript	28
3.3.3	SQL	28
3.3.4	Bash	28
3.4	<i>Βιβλιοθήκες πρόσκτησης δεδομένων</i>	29
3.4.1	Requests	29
3.4.2	Beautiful Soup	29
3.4.3	Markdownify	30
3.4.4	Playwright	30
3.4.5	Certifi και η επαλήθευση των πιστοποιητικών	30
3.5	<i>Εξαγωγή και επεξεργασία κειμένου</i>	30
3.5.1	Docling	30
3.5.2	mlx-vlm	31
3.5.3	Pdflumber και PyMuPDF	32
3.5.4	Python-docx	32
3.5.5	Pandas	32
3.5.6	Τεμαχισμός κειμένου	33
3.6	<i>Μοντέλα και περιβάλλοντα εκτέλεσης</i>	34
3.6.1	Το πλαίσιο MLX	34
3.6.2	Το γλωσσικό μοντέλο Qwen3-8B	35
3.6.3	PyTorch και NumPy	35
3.6.4	Η βιβλιοθήκη sentence-transformers και το μοντέλο ενσωμάτωσης BGE-M3	36

3.7	<i>Ανάκτηση και αποθήκευση διανυσμάτων</i>	37
3.7.1	Το σύστημα Qdrant	37
3.7.2	LangChain	38
3.7.3	Η βιβλιοθήκη snowballstemmer και η ελληνική μορφολογία στο αραιό σκέλος	39
3.8	<i>Τεχνολογίες διαδικτυακής εφαρμογής</i>	40
3.8.1	FastAPI και Unicorn	40
3.8.2	Server-Sent Events	40
3.8.3	Pydantic και διαχείριση παραμέτρων	41
3.8.4	Next.js και React	41
3.8.5	HTML και CSS	42
3.9	<i>Συστήματα αποθήκευσης της εφαρμογής</i>	43
3.9.1	PostgreSQL, SQLAlchemy και psycorg	43
3.9.2	SQLite σε λειτουργία WAL	44
3.10	<i>Εργαλεία ανάπτυξης και υποδομής</i>	44
3.10.1	Docker και Docker Compose	44
3.10.2	launchd	45
3.10.3	Pytest	45
3.10.4	Git	45
3.10.5	Loguru	46
Κεφάλαιο 4ο:	Σχεδίαση και υλοποίηση του συστήματος	46
4.1	<i>Λειτουργικές απαιτήσεις</i>	46
4.1.1	Συνολική αρχιτεκτονική	47
4.1.2	Αρχές σχεδίασης	48
4.1.3	Οργάνωση του κώδικα	49
4.2	<i>Το σώμα και ο αγωγός ευρετηρίασης</i>	50
4.2.1	Πηγές και συλλογή εγγράφων	50
4.2.2	Εξαγωγή κειμένου και διαδοχή αναλυτών	52
4.2.3	Μεταδεδομένα και χαρακτηρισμός	53
4.2.4	Ενσωμάτωση, αποθήκευση και επανευρετηρίαση	55
4.3	<i>Οι ντετερμινιστικές διαδρομές</i>	56
4.3.1	Έτοιμες απαντήσεις και αναδιατύπωση με ιστορικό	56
4.3.2	Το επίπεδο γεγονότων: δομή και σειρά χειριστών	57
4.3.3	Η πύλη εισόδου	58
4.3.4	Η σημασιολογική διαδρομή	59
4.3.5	Κανονικοποίηση του ερωτήματος	59
4.3.6	Δρομολόγηση πρόθεσης και κατασκευή της δεξαμενής	61
4.3.7	Επανακατάταξη και πολιτική εγκυρότητας	63
4.3.8	Διαμόρφωση των συμφραζομένων	64
4.3.9	Κλιμάκωση σε ασθενή ανάκτηση	65
4.4	<i>Αποσύνθεση σύνθετων ερωτημάτων</i>	66
4.5	<i>Παραγωγή της απάντησης</i>	68
4.5.1	Η πύλη απάντησης	68
4.5.2	Δομή της προτροπής και απόδοση των συμφραζομένων	69

4.5.3	Προσωρινή μνήμη προθέματος και ροή εξόδου	70
4.5.4	Έλεγχος μετά την παραγωγή	71
4.6	<i>Παράδοση: η διαδικτυακή εφαρμογή</i>	71
4.6.1	Διεπαφή προγραμματισμού, ροή συμβάντων και κατάσταση συνομιλίας	71
Κεφάλαιο 5ο: Μεθοδολογία και όργανα αξιολόγησης		73
5.1	<i>Μεθοδολογία και όργανα αξιολόγησης</i>	73
5.1.1	Όργανο Α — Πύλη ανάκτησης	74
5.1.2	Όργανο Β — Ευρωστία του ερωτήματος	74
5.1.3	Όργανο Γ — Ερωτήματα εκτός εμβέλειας	74
5.1.4	Όργανο Δ — Ευρωστία της απάντησης	75
5.1.5	Καθεστώς των σφραγισμένων συνόλων ελέγχου	75
5.1.6	Πειθαρχία και διαδικασία των μετρήσεων	76
5.2	<i>Μετρικές, ορισμοί και διαδικασία βαθμολόγησης</i>	77
5.2.1	Μετρικές ανάκτησης [45, 46]	77
5.2.2	Κανονικοποιημένη ανάκληση γεγονότων (nFR)	78
5.2.3	Ορθότητα συμπεριφοράς (behavior_ok)	79
5.2.4	Διαρροή	80
5.2.5	Το όριο του λεξιλογικού ελέγχου	80
5.2.6	Η διαδικασία μιας μέτρησης	81
5.3	<i>Απόδοση της ανάκτησης</i>	82
5.3.1	Η τρέχουσα πύλη	82
5.3.2	Η εξέλιξη της πύλης	83
5.3.3	Ευρωστία της ανάκτησης σε παραλλαγές του ερωτήματος	85
5.3.4	Η παράμετρος αναζήτησης του ευρετηρίου	86
5.3.5	Περιορισμοί της τιμής Recall@8 = 1,0000	86
5.4	<i>Απόδοση της απάντησης και της άρνησης</i>	87
5.4.1	Το σφραγισμένο αποτέλεσμα	87
5.4.2	Απόδοση ανά τύπο διατύπωσης	88
5.4.3	Δύο μητρώα απόκρισης	89
5.4.4	Άρνηση σε ερωτήματα εκτός πεδίου	89
5.4.5	Χειροκίνητη απογραφή ορθότητας	91
5.4.6	Καθυστέρηση	92
5.5	<i>Παρεμβάσεις που βελτίωσαν το σύστημα</i>	93
5.5.1	Φίλτρα εισόδου στη συλλογή	94
5.5.2	Αναδιατύπωση σε ακαδημαϊκό ύφος	94
5.5.3	Τεμαχισμός ανά εγγραφή	95
5.5.4	Στένεμα του ντετερμινιστικού επιπέδου γεγονότων	96
5.5.5	Οι δύο πύλες αποχής στον χειριστή επικοινωνίας	97
5.5.6	Αποκλιτισμός της γενικής πτώσης	98
5.6	<i>Παρεμβάσεις που απορρίφθηκαν και το εύρημα της απορρόφησης</i>	98
5.6.1	Η απορρόφηση των βελτιώσεων από τον επανακατατάκτη	98
5.6.2	Παρεμβάσεις με μετρήσιμο κόστος	99
5.6.3	Οι κριτές ποιότητας απάντησης	100
5.6.4	Υπόλοιπες απορρίψεις	101

5.7	<i>Περιορισμοί της αξιολόγησης</i>	102
5.7.1	<i>Κάλυψη των συνόλων δοκιμών</i>	102
5.7.2	<i>Τεκμηρίωση και αναπαραγωγικότητα</i>	102
5.7.3	<i>Περιορισμοί των μετρήσεων</i>	102
5.7.4	<i>Διαστάσεις που δεν αξιολογήθηκαν</i>	103
Κεφάλαιο 6ο:	Συμπεράσματα και μελλοντικές επεκτάσεις	104
6.1	<i>Σύνοψη και απαντήσεις στα ερευνητικά ερωτήματα</i>	104
6.2	<i>Συνεισφορά</i>	104
6.3	<i>Μελλοντικές επεκτάσεις</i>	105
ΒΙΒΛΙΟΓΡΑΦΙΑ		106
Παράρτημα Α:	Ρυθμίσεις, παράμετροι και εκδόσεις	110
Παράρτημα Β:	Το ντετερμινιστικό επίπεδο και η πύλη εισόδου	113
Παράρτημα Γ:	Ανάκτηση, σύντηξη και επανακατάταξη	114
Παράρτημα Δ:	Παραγωγή της απάντησης και έλεγχοι	116
Παράρτημα Ε:	Ευρετηρίαση	118
Παράρτημα ΣΤ:	Ελληνικός λεξικός ανακτητής	119
Παράρτημα Ζ:	Κανονικοποίηση ελληνικού ερωτήματος	121
Παράρτημα Η:	Το σύνολο ερωτημάτων αξιολόγησης	122

Κατάλογος Σχημάτων

ΣΧΗΜΑ 4.1: ΣΥΝΟΛΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	48
--	----

Κατάλογος Πινάκων

ΠΙΝΑΚΑΣ 3.1: ΠΛΑΙΣΙΑ ΕΚΤΕΛΕΣΗΣ ΠΟΥ ΕΞΕΤΑΣΤΗΚΑΝ ΚΑΙ ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ.....	34
ΠΙΝΑΚΑΣ 4.1: ΚΑΤΗΓΟΡΙΕΣ ΠΗΓΩΝ, ΜΕΘΟΔΟΙ ΑΝΑΚΤΗΣΗΣ ΚΑΙ ΤΥΠΟΙ ΕΓΓΡΑΦΩΝ.....	50
ΠΙΝΑΚΑΣ 4.2: ΒΑΘΜΙΔΕΣ ΕΞΑΓΩΓΗΣ ΚΕΙΜΕΝΟΥ ΑΠΟ ΑΡΧΕΙΑ PDF	52
ΠΙΝΑΚΑΣ 4.3: ΣΤΡΑΤΗΓΙΚΗ ΤΕΜΑΧΙΣΜΟΥ ΚΑΙ ΠΑΡΑΜΕΤΡΟΙ ΑΝΑ ΤΥΠΟ ΕΓΓΡΑΦΟΥ	54
ΠΙΝΑΚΑΣ 4.4: ΤΑ ΔΥΟ ΣΚΕΛΗ ΤΗΣ ΥΒΡΙΔΙΚΗΣ ΑΝΑΚΤΗΣΗΣ ΚΑΙ ΤΑ ΒΑΡΗ ΤΟΥΣ.....	62
ΠΙΝΑΚΑΣ 4.5: ΠΡΟΣΑΡΜΟΓΕΣ ΤΗΣ ΒΑΘΜΟΛΟΓΙΑΣ ΕΠΑΝΑΚΑΤΑΤΑΞΗΣ	64
ΠΙΝΑΚΑΣ 5.1: ΣΥΓΚΡΙΤΙΚΗ ΕΙΚΟΝΑ ΤΩΝ ΤΕΣΣΑΡΩΝ ΟΡΓΑΝΩΝ ΑΞΙΟΛΟΓΗΣΗΣ.....	73
ΠΙΝΑΚΑΣ 5.2: ΚΑΘΕΣΤΩΣ ΤΩΝ ΣΦΡΑΓΙΣΜΕΝΩΝ ΣΥΝΟΛΩΝ ΕΛΕΓΧΟΥ	75
ΠΙΝΑΚΑΣ 5.3: ΤΑΒΑΝΙΑ ΤΗΣ ΚΑΝΟΝΙΚΟΠΟΙΗΜΕΝΗΣ ΑΝΑΚΛΗΣΗΣ ΓΕΓΟΝΟΤΩΝ	79
ΠΙΝΑΚΑΣ 5.4: ΤΙΜΕΣ ΤΗΣ ΠΥΛΗΣ ΑΝΑ ΔΙΑΙΡΕΣΗ ΤΟΥ ΟΡΓΑΝΟΥ Α (K = 8)	82
ΠΙΝΑΚΑΣ 5.5: ΚΥΡΙΑ ΣΗΜΕΙΑ ΕΞΕΛΙΞΗΣ ΤΗΣ ΠΥΛΗΣ ΑΝΑΚΤΗΣΗΣ	83
ΠΙΝΑΚΑΣ 5.6: ΑΠΟΔΟΣΗ ΤΗΣ ΑΝΑΚΤΗΣΗΣ ΣΤΙΣ ΔΙΑΙΡΕΣΕΙΣ ΤΟΥ ΟΡΓΑΝΟΥ Β.....	85
ΠΙΝΑΚΑΣ 5.7: ΣΑΡΩΣΗ ΤΗΣ ΠΑΡΑΜΕΤΡΟΥ ΑΝΑΖΗΤΗΣΗΣ ΕΝΑΝΤΙ ΤΗΣ ΠΥΛΗΣ.....	86
ΠΙΝΑΚΑΣ 5.8: ΣΦΡΑΓΙΣΜΕΝΟ ΣΥΝΟΛΟ ΕΛΕΓΧΟΥ ΕΝΑΝΤΙ ΣΥΝΟΛΟΥ ΑΝΑΠΤΥΞΗΣ.....	87
ΠΙΝΑΚΑΣ 5.9: ΟΡΘΟΤΗΤΑ ΣΥΜΠΕΡΙΦΟΡΑΣ ΚΑΙ NFR ΑΝΑ ΤΥΠΟ ΔΙΑΤΥΠΩΣΗΣ	88
ΠΙΝΑΚΑΣ 5.10: ΟΡΘΟΤΗΤΑ ΣΥΜΠΕΡΙΦΟΡΑΣ ΑΝΑ ΑΝΑΜΕΝΟΜΕΝΟ ΤΥΠΟ ΑΠΟΚΡΙΣΗΣ.....	89
ΠΙΝΑΚΑΣ 5.11: ΠΟΣΟΣΤΟ ΑΡΝΗΣΗΣ ΑΝΑ ΚΑΤΗΓΟΡΙΑ, ΣΥΝΟΛΟ ΑΝΑΠΤΥΞΗΣ (N = 30).....	90
ΠΙΝΑΚΑΣ 5.12: ΠΟΣΟΣΤΟ ΑΡΝΗΣΗΣ ΑΝΑ ΚΑΤΗΓΟΡΙΑ, ΣΦΡΑΓΙΣΜΕΝΟ ΣΥΝΟΛΟ (N = 23).....	90
ΠΙΝΑΚΑΣ 5.13: ΧΕΙΡΟΚΙΝΗΤΗ ΤΑΞΙΝΟΜΗΣΗ 497 ΑΠΑΝΤΗΣΕΩΝ ΑΝΑ ΤΥΠΟ ΔΙΑΤΥΠΩΣΗΣ	91
ΠΙΝΑΚΑΣ 5.14: ΑΝΑΛΥΣΗ ΤΩΝ 160 ΛΑΝΘΑΣΜΕΝΩΝ ΑΠΑΝΤΗΣΕΩΝ	91
ΠΙΝΑΚΑΣ 5.15: ΜΕΤΡΗΣΕΙΣ ΚΑΘΥΣΤΕΡΗΣΗΣ ΚΑΙ ΚΑΘΕΣΤΩΣ ΤΕΚΜΗΡΙΩΣΗΣ.....	92
ΠΙΝΑΚΑΣ 5.16: ΣΥΓΚΕΝΤΡΩΤΙΚΗ ΕΙΚΟΝΑ ΤΩΝ ΠΑΡΕΜΒΑΣΕΩΝ.....	93
ΠΙΝΑΚΑΣ 5.17: ΕΠΙΔΡΑΣΗ ΤΗΣ ΑΝΑΔΙΑΤΥΠΩΣΗΣ ΣΤΗ ΜΕΣΗ ΑΝΑΚΛΗΣΗ ΤΟΥ ΟΡΓΑΝΟΥ Β	94
ΠΙΝΑΚΑΣ 5.18: ΘΕΣΗ ΚΑΙ ΒΑΘΜΟΛΟΓΙΑ ΤΟΥ ΦΟΡΕΑ ΤΗΣ ΑΠΑΝΤΗΣΗΣ ΠΡΙΝ ΚΑΙ ΜΕΤΑ ΤΗΝ ΠΑΡΕΜΒΑΣΗ	95
ΠΙΝΑΚΑΣ 5.19: ΔΙΑΔΟΧΙΚΕΣ ΣΤΕΝΕΥΣΕΙΣ ΤΟΥ ΝΤΕΡΜΙΝΙΣΤΙΚΟΥ ΕΠΙΠΕΔΟΥ ΓΕΓΟΝΟΤΩΝ....	96
ΠΙΝΑΚΑΣ 5.20: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ ΔΥΟ ΠΥΛΩΝ ΑΠΟΧΗΣ	97
ΠΙΝΑΚΑΣ 5.21: ΒΕΛΤΙΩΣΕΙΣ ΕΠΙΜΕΡΟΥΣ ΣΚΕΛΩΝ ΚΑΙ ΕΠΙΔΡΑΣΗ ΣΤΗΝ ΤΕΛΙΚΗ ΠΥΛΗ.....	98
ΠΙΝΑΚΑΣ 5.22: ΕΠΙΔΡΑΣΗ ΤΗΣ ΔΙΕΥΡΥΝΣΗΣ ΤΟΥ ΠΑΡΑΘΥΡΟΥ ΕΙΣΟΔΟΥ ΤΟΥ ΕΠΑΝΑΚΑΤΑΤΑΚΤΗ	99
ΠΙΝΑΚΑΣ 5.23: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ ΔΥΟ ΚΡΙΤΩΝ.....	100
ΠΙΝΑΚΑΣ Α.1: ΣΤΑΘΕΡΕΣ ΠΟΥ ΚΑΤΟΝΟΜΑΖΟΝΤΑΙ ΣΤΑ ΚΕΦΑΛΑΙΑ 4 ΚΑΙ 5	110
ΠΙΝΑΚΑΣ Α.2: ΠΛΗΘΟΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΑΝΑ ΣΤΑΔΙΟ ΤΗΣ ΑΝΑΚΤΗΣΗΣ.....	110
ΠΙΝΑΚΑΣ Α.3: ΣΤΡΑΤΗΓΙΚΗ ΤΕΜΑΧΙΣΜΟΥ ΚΑΙ ΕΠΙΚΑΛΥΨΗ ΑΝΑ ΤΥΠΟ ΕΓΓΡΑΦΟΥ	111
ΠΙΝΑΚΑΣ Α.4: ΠΑΡΑΜΕΤΡΟΙ ΑΠΟΚΩΔΙΚΟΠΟΙΗΣΗΣ ΤΟΥ ΓΛΩΣΣΙΚΟΥ ΜΟΝΤΕΛΟΥ	111
ΠΙΝΑΚΑΣ Α.5: ΕΚΔΟΣΕΙΣ ΤΩΝ ΒΙΒΛΙΟΘΗΚΩΝ ΠΟΥ ΑΝΑΦΕΡΟΝΤΑΙ ΣΤΟ ΚΕΦΑΛΑΙΟ 3	112
ΠΙΝΑΚΑΣ Α.6: ΠΛΑΤΦΟΡΜΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΥΠΟΔΟΜΗΣ	112
ΠΙΝΑΚΑΣ Α.7: ΜΕΤΑΒΟΛΗ ΤΟΥ ΜΕΓΕΘΟΥΣ ΤΗΣ ΣΥΛΛΟΓΗΣ ΚΑΤΑ ΤΗΝ ΠΕΡΙΟΔΟ ΤΩΝ ΜΕΤΡΗΣΕΩΝ.....	112
ΠΙΝΑΚΑΣ Η.1: ΤΑ ΒΑΣΙΚΑ ΕΡΩΤΗΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ, Η ΑΝΑΜΕΝΟΜΕΝΗ ΚΑΙ Η ΠΡΑΓΜΑΤΙΚΗ ΑΠΑΝΤΗΣΗ.....	122

Συντομογραφίες

Στον πίνακα που ακολουθεί παρατίθενται οι συντομογραφίες και τα αρκτικόλεξα που χρησιμοποιούνται στο κείμενο της εργασίας.

Συντομογραφία	Πλήρης όρος
ANN	Approximate Nearest Neighbor
API	Application Programming Interface
ASGI	Asynchronous Server Gateway Interface
BGE	Beijing Academy of Artificial Intelligence General Embedding
BM25	Best Matching 25
CORS	Cross-Origin Resource Sharing
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DCG	Discounted Cumulative Gain
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
HNSW	Hierarchical Navigable Small World
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDF	Inverse Document Frequency
IVF	Inverted File Index
JSON	JavaScript Object Notation
KV Cache	Key-Value Cache
LLM	Large Language Model
LSH	Locality-Sensitive Hashing
MLX	Machine Learning for Apple Silicon
MPS	Metal Performance Shaders
MRR	Mean Reciprocal Rank
nDCG	Normalized Discounted Cumulative Gain
NFC / NFD	Normalization Form Composed / Decomposed
nFR	Normalized Fact Recall
NLI	Natural Language Inference
RAG	Retrieval-Augmented Generation
REST	Representational State Transfer
RLHF	Reinforcement Learning from Human Feedback
RRF	Reciprocal Rank Fusion
SFT	Supervised Fine-Tuning
SQL	Structured Query Language
SSE	Server-Sent Events
URL	Uniform Resource Locator

Συντομογραφία

Πλήρης όρος

WAL

Write-Ahead Logging

ΔΙ.ΠΑ.Ε.

Διεθνές Πανεπιστήμιο της Ελλάδος

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η καθημερινή λειτουργία της γραμματείας ενός ακαδημαϊκού τμήματος βασίζεται σε έναν μεγάλο όγκο θεσμικών εγγράφων όπως ανακοινώσεις, κανονισμούς, προγράμματα σπουδών, έντυπα και πληροφορίες σχετικά με προθεσμίες. Ο κατακερματισμός αυτών των δεδομένων σε διαφορετικές πηγές, σε συνδυασμό με τους ασύγχρονους ρυθμούς ανανέωσής τους, δυσκολεύει την πρόσβαση του φοιτητή στην πληροφορία. Παράλληλα, απουσιάζει ένα ενιαίο σημείο στο οποίο ο φοιτητής να μπορεί να απευθύνει ένα ερώτημα διατυπωμένο σε φυσική γλώσσα και να λαμβάνει συγκεντρωμένη απάντηση.

Η αντιμετώπιση του προβλήματος ωστόσο δεν περιορίζεται στον εντοπισμό ενός σχετικού εγγράφου. Τα περισσότερα έγγραφα του συστήματος έχουν περιορισμένη χρονική ισχύ, ενώ παλαιότερες εκδόσεις συνυπάρχουν μαζί με τις νεότερες. Επιπλέον, η διατύπωση που χρησιμοποιούν οι φοιτητές για ένα ερώτημα είναι πιο καθημερινή και είναι πιο δύσκολο να αναγνωριστεί το ζητούμενο ερώτημα. Ιδιαίτερη σημασία έχει και η αξιοπιστία της απάντησης, καθώς πρέπει να βασίζεται σε έγκυρα δεδομένα και να μη δημιουργεί ψευδή βεβαιότητα, οδηγώντας σε λανθασμένη ενέργεια.

Επομένως, ένα σύστημα γραμματειακής υποστήριξης για να λειτουργεί σωστά χρειάζεται να εντοπίζει την ισχύουσα πληροφορία, να τη μετατρέπει σε κατανοητή μορφή φυσικής γλώσσας και όταν δεν υπάρχει επαρκής τεκμηρίωση να αποφεύγει την παραγωγή μη υποστηριζόμενης απάντησης.

Αυτήν ακριβώς την ανάγκη καλύπτει η αρχιτεκτονική Ανάκτησης με Ενισχυμένη Παραγωγή (Retrieval-Augmented Generation – RAG) καθώς συνδυάζει την ανάκτηση των σχετικών εγγράφων από συγκεκριμένες πηγές με τη σύνθεση μιας απάντησης μέσω γλωσσικού μοντέλου. Ωστόσο, η εφαρμογή του συστήματος σε ένα ακαδημαϊκό ίδρυμα επιφυλάσσει αρκετές προκλήσεις και ζητήματα που πρέπει να επιλυθούν [1, 2].

1.2 Στόχος και ερευνητικά ερωτήματα

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός, η υλοποίηση και η αξιολόγηση ενός ελληνόγλωσσου συστήματος RAG για τη γραμματειακή υποστήριξη ακαδημαϊκού τμήματος. Βασικός στόχος του συστήματος είναι να απαντά σε ερωτήσεις που σχετίζονται με διοικητικά θέματα βασισμένο σε πηγές του τμήματος και να εκτελείται αποκλειστικά τοπικά, χωρίς να καλεί εξωτερικές υπηρεσίες κατά τη λειτουργία του.

Η πλήρως τοπική λειτουργία αποτελεί σχεδιαστικό περιορισμό από την αρχή, με στόχο να αποφεύγεται η μεταφορά δεδομένων σε εξωτερικές υπηρεσίες. Η συγκεκριμένη επιλογή περιορίζει το εύρος των διαθέσιμων τεχνολογικών λύσεων και απαιτεί τη λειτουργία του συστήματος με βάση τους διαθέσιμους υπολογιστικούς πόρους.

Η μελέτη οργανώνεται γύρω από τέσσερα ερευνητικά ερωτήματα:

- E1. Έχει τη δυνατότητα ένα σύστημα RAG, το οποίο εκτελείται τοπικά σε υλικό καταναλωτικών προδιαγραφών, να παρέχει τεκμηριωμένες απαντήσεις σε ελληνόγλωσσα διοικητικά ερωτήματα, αντλώντας πληροφορίες από ετερογενείς θεσμικές πηγές;
- E2. Πώς επηρεάζει η διατύπωση του ερωτήματος την απόδοση του συστήματος, όταν χρησιμοποιούνται καθομιλουμένη γλώσσα, greeklish, ορθογραφικά λάθη ή ελλιπείς διατυπώσεις;

- E3. Σε ποιο στάδιο εμφανίζονται κυρίως οι αστοχίες του συστήματος, στην ανάκτηση ή στην παραγωγή της απάντησης, ποιο είναι το προφίλ των σφαλμάτων του και πώς συμπεριφέρεται όταν η διαθέσιμη πληροφορία δεν επαρκεί για τεκμηριωμένη απάντηση;
- E4. Ποιες παρεμβάσεις στα επιμέρους στάδια του συστήματος αποδίδουν μετρήσιμο όφελος και ποιες όχι, και ποια συμπεράσματα προκύπτουν από τα αποτελέσματα αυτά για την αρχιτεκτονική του συστήματος;

1.3 Δομή της διπλωματικής εργασίας

Στο 1ο Κεφάλαιο παρουσιάζεται το πλαίσιο του προβλήματος, ο στόχος της εργασίας και τα ερευνητικά ερωτήματα. Στο 2ο Κεφάλαιο αναλύεται το πρόβλημα της πρόσβασης σε θεσμική πληροφορία, παρουσιάζονται οι βασικές έννοιες που υποστηρίζουν θεωρητικά την υλοποίηση —γλωσσική επεξεργασία, διανυσματικές αναπαραστάσεις, γλωσσικά μοντέλα, ανάκτηση πληροφορίας και αρχιτεκτονική RAG— και εξετάζονται οι σχετικές εργασίες. Στο 3ο Κεφάλαιο παρουσιάζονται οι σχεδιαστικοί περιορισμοί και οι τεχνολογίες που επιλέχθηκαν. Στο 4ο Κεφάλαιο περιγράφεται η υλοποίηση του συστήματος, από τη συλλογή και την ευρετηρίαση των εγγράφων έως την παραγωγή και την παράδοση της απάντησης. Στο 5ο Κεφάλαιο παρουσιάζεται η μεθοδολογία που ακολουθήθηκε για την αξιολόγηση, οι μετρικές και τα αποτελέσματα του συστήματος, καθώς και η επίδραση των παρεμβάσεων που εξετάστηκαν κατά την ανάπτυξή του. Τέλος, στο 6ο Κεφάλαιο συνοψίζονται τα βασικά συμπεράσματα, παρουσιάζεται η συνεισφορά της εργασίας και προτείνονται κατευθύνσεις για μελλοντική επέκταση.

Κεφάλαιο 2ο: Θεωρητικό υπόβαθρο

2.1 Εισαγωγή

Ο σκοπός αυτού του κεφαλαίου είναι η παρουσίαση του θεωρητικού υπόβαθρου που απαιτείται για την κατανόηση του συστήματος που υλοποιείται. Αρχικά, εξετάζεται η δυσκολία ανάκτησης της θεσμικής πληροφορίας καθώς και προκλήσεις που αφορούν την εκτέλεση του συστήματος και την προστασία των δεδομένων που επεξεργάζεται. Στη συνέχεια, αναλύονται βασικές έννοιες της επεξεργασίας φυσικής γλώσσας, καθώς και ο τρόπος λειτουργίας των μεγάλων γλωσσικών μοντέλων. Στο τέλος παρουσιάζεται ο βασικός μηχανισμός που είναι η αρχιτεκτονική της ανάκτησης με ενισχυμένη παραγωγή (Retrieval-Augmented Generation – RAG) και οι βασικές αρχές λειτουργίας της, διαμορφώνοντας το θεωρητικό πλαίσιο για τις τεχνολογικές και σχεδιαστικές επιλογές που ακολουθούν στα επόμενα κεφάλαια.

2.2 Το πρόβλημα της πρόσβασης σε θεσμική πληροφορία

2.2.1 Η φύση του προβλήματος

Η πληροφορία που διαχειρίζεται μια γραμματεία πανεπιστημιακού τμήματος έχει μεγάλο όγκο δεδομένων και η οργάνωση της είναι ιδιαίτερα απαιτητική. Ειδικότερα, στο σύστημα υπάρχουν ανακοινώσεις, κανονισμοί, προγράμματα σπουδών, προθεσμίες και ένα ευρύ φάσμα διοικητικών εγγράφων. Κάθε τύπος εγγράφου παράγεται από ξεχωριστή διοικητική διαδικασία και έχει διαφορετικό ρυθμό ανανέωσης. Παρόλα αυτά, το αποτέλεσμα είναι ότι ο χρήστης πρέπει να ενημερώνεται από μόνος του για κάθε πληροφορία που του χρειάζεται καθώς δεν υπάρχει ένα κεντρικό σημείο ενημέρωσης στο οποίο να μπορεί να υποβάλει ένα ερώτημα διατυπωμένο σε φυσική γλώσσα.

Παράλληλα, μία ακόμη σημαντική παράμετρος είναι η μεταβλητότητα. Ένα μεγάλο μέρος της θεσμικής πληροφορίας διαθέτει περιορισμένη χρονική ισχύ που όταν παρέλθει αντικαθίσταται από νέα έκδοση ή σταματάει να ισχύει. Μετά το πέρας της λήξης τους όμως παραμένουν στο σύστημα και συνυπάρχουν με τα ισχύοντα, φέροντας συχνά πανομοιότυπη μορφή και τίτλο. Σε αυτό το πλαίσιο, η ανάκτηση ενός αρχείου που δεν ισχύει το περιεχόμενό του πλέον αποτελεί χειρότερο σενάριο από την αποτυχία ανάκτησης του. Η δεύτερη κατευθύνει τον χρήστη στην αναζήτηση εναλλακτικής πηγής, ενώ η πρώτη του παρέχει ψευδή βεβαιότητα.

Το κόστος αυτού του σφάλματος είναι εξίσου ασύμμετρο. Η απλή άρνηση σε ένα ερώτημα συνεπάγεται απλώς χρονική καθυστέρηση, ενώ η λανθασμένη πληροφόρηση μπορεί να οδηγήσει σε λάθος συμπεράσματα ή ενδέχεται να οδηγήσει σε εσφαλμένες ενέργειες. Η ασυμμετρία αυτή διαφοροποιεί τις απαιτήσεις ενός συστήματος θεσμικής ενημέρωσης από εκείνες ενός γενικού συστήματος αναζήτησης, όπου η επιστροφή πληροφορίας μέτριας συνάφειας έχει συγκριτικά μικρό κόστος.

Ένα σημαντικό ζήτημα αφορά την κατανομή των ίδιων των ερωτημάτων. Τα περισσότερα ερωτήματα προς μία γραμματεία κατανέμονται ανομοιόμορφα σε διάφορες χρονικές στιγμές. Κατά χρονικές περιόδους όπως αυτές των εγγράφων, των δηλώσεων μαθημάτων και των εξετάσεων, τα ερωτήματα παρουσιάζουν υψηλό βαθμό επανάληψης και έντονη χρονική συγκέντρωση. Αυτό έχει ως αποτέλεσμα να δημιουργείται ένας μεγάλος φόρτος εργασίας ο οποίος είναι δυσανάλογος προς το πλήθος των διακριτών πληροφοριών που ζητούνται.

Τέλος, παρατηρείται σημαντική διαφορά ανάμεσα στη διατύπωση των ερωτημάτων των φοιτητών και στο λεκτικό των επίσημων εγγράφων. Η διατύπωση που έχουν αυτά τα έγγραφα είναι κανονιστική και

επίσημη, ενώ οι φοιτητές διατυπώνουν τα ερωτήματά τους σε καθημερινή γλώσσα, και προσπαθούν να περιγράψουν την κατάσταση στην οποία βρίσκονται. Η απόσταση αυτή, γνωστή στη βιβλιογραφία της Ανάκτησης Πληροφορίας ως «χάσμα λεξιλογίου» (vocabulary mismatch), καθιστά εγγενώς ανεπαρκή οποιονδήποτε παραδοσιακό μηχανισμό αναζήτησης που στηρίζεται αποκλειστικά στη λεξική επικάλυψη (lexical overlap) των όρων [3].

2.2.2 Υπάρχουσες προσεγγίσεις

Μια συνηθισμένη προσέγγιση στο πρόβλημα της θεσμικής ενημέρωσης είναι η προσθήκη σελίδας συχνών ερωτήσεων και απαντήσεων. Με αυτόν τον τρόπο καλύπτεται ένα σημαντικό μέρος των αποριών που ενδέχεται να προκύψουν από την πλευρά των φοιτητών. Το βασικό πλεονέκτημα είναι ότι οι απαντήσεις συντάσσονται απευθείας από τον αρμόδιο φορέα και είναι, επομένως, έγκυρες. Ωστόσο, υπάρχει η πιθανότητα να μην καλύπτει όλο το εύρος ερωτήσεων που μπορεί να προκύψουν και, επομένως, όταν η περίπτωση του χρήστη δεν αντιστοιχεί σε καμία από τις διαθέσιμες τότε πρέπει να αναζητήσει άλλη προσέγγιση.

Παράλληλα, οι παραδοσιακές μηχανές αναζήτησης εντός των ιστότοπων αντιμετωπίζουν εν μέρει το πρόβλημα του εντοπισμού εγγράφων αλλά δεν παράγουν κάποια φυσική απάντηση. Επιστρέφουν αποτελέσματα ταξινομημένα βάσει αλγορίθμων συνάφειας, αφήνοντας στον χρήστη το έργο της ανάγνωσης, της εξαγωγής της σχετικής παραγράφου και της τελικής κρίσης περί της εγκυρότητας του εγγράφου. Σε περιπτώσεις όπου υπάρχουν εκτενή κανονιστικά κείμενα, η δυσκολία εύρεσης του ζητούμενου στοιχείου το οποίο καταλαμβάνει ελάχιστες γραμμές εντείνεται σημαντικά.

Τέλος τα συστήματα διαλόγου με προκαθορισμένες προθέσεις είναι μία λύση καθώς παρέχουν διεπαφή φυσικής γλώσσας, αλλά η συντήρησή τους είναι εξαιρετικά κοστοβόρα. Για την απάντηση ενός απλού ερωτήματος, χρειάζεται να πραγματοποιηθεί εκτενής εκπαίδευση σε δεδομένα και κανόνες, καθώς και συνεχόμενη ενημέρωση των πληροφοριών που μεταβάλλονται [4, 5].

2.2.3 Προστασία δεδομένων και τοπική επεξεργασία

Τα ερωτήματα προς μία γραμματεία συνήθως περιλαμβάνουν προσωπικά στοιχεία του ερωτώντος όπως αριθμός μητρώου, ονοματεπώνυμο, έτος εισαγωγής και άλλα. Τα στοιχεία αυτά είναι προσωπικά δεδομένα και χρειάζονται αντίστοιχη μεταχείριση. Σύμφωνα με τον Γενικό Κανονισμό για την Προστασία Δεδομένων, το ίδρυμα που θα λειτουργήσει ένα τέτοιο σύστημα ενεργεί ως υπεύθυνος επεξεργασίας και φέρει τις αντίστοιχες υποχρεώσεις, ανεξάρτητα από το ποιος παρέχει την τεχνική υποδομή [6].

Σε περίπτωση που κάποιος φορέας εκτός πανεπιστημίου αναλάβει την διαχείριση αυτών των στοιχείων μεταβάλλεται η νομική εικόνα με δύο τρόπους. Αρχικά, η ανάθεση της επεξεργασίας σε εξωτερικό πάροχο μπορεί να απαιτεί τη σύναψη σχετικής σύμβασης μεταξύ του υπευθύνου επεξεργασίας και του παρόχου. Δεύτερον, όταν ο πάροχος είναι εγκατεστημένος εκτός του Ευρωπαϊκού Οικονομικού Χώρου (ΕΟΧ), η αποστολή του ερωτήματος συνιστά διαβίβαση δεδομένων προς τρίτη χώρα και υπάγεται στους περιορισμούς του Κεφαλαίου V του Κανονισμού (άρθρα 44-49). Η νομιμότητα τέτοιων διαβιβάσεων στηρίζεται ιστορικά σε «αποφάσεις επάρκειας» (άρθρο 45), η νομική ισχύς των οποίων έχει αμφισβητηθεί δικαστικά κατ' επανάληψη (π.χ. ΔΕΕ C-311/18, Schrems II) και παραμένει αντικείμενο εκκρεμούς κρίσης (π.χ., εκκρεμής αναίρεση C-703/25 P). Ένα σύστημα που βασίζεται σε εξωτερικούς παρόχους ενδέχεται, επομένως, να εξαρτάται από νομικές και κανονιστικές προϋποθέσεις οι οποίες δεν ελέγχονται από τον σχεδιαστή του. Συγχρόνως, πάροχοι τέτοιου είδους επιτρέπουν τη διατήρηση των υποβαλλόμενων δεδομένων για λόγους βελτίωσης της υπηρεσίας, γεγονός που έρχεται

σε αντίθεση με την αρχή του περιορισμού του σκοπού και με την αρχή της ελαχιστοποίησης των δεδομένων.

Αξίζει να σημειωθεί ότι η εκτέλεση του συνόλου της επεξεργασίας τοπικά, εντός των υποδομών του πανεπιστημίου, αποτελεί μια ιδιαίτερα προσιτή λύση όσον αφορά την προστασία των δεδομένων. Η προσέγγιση αυτή δικαιολογείται από το γεγονός ότι αποτρέπει πλήρως τη διαβίβαση προσωπικών δεδομένων σε τρίτους και αποκλείει κάθε ενδεχόμενο δευτερογενούς χρήσης. Αυτό, ωστόσο, εισάγει την αναγκαιότητα ενός τεχνικού συμβιβασμού, καθώς το σύστημα χρειάζεται να λειτουργήσει με τους περιορισμένους υπολογιστικούς πόρους που διαθέτει το ίδρυμα. Αυτό, αναπόφευκτα, περιορίζει σημαντικά τις διαθέσιμες επιλογές σε γλωσσικά μοντέλα καθώς οι υπολογιστικοί πόροι είναι περιορισμένοι και χρειάζεται αντίστοιχη αρχιτεκτονική.

2.3 Γλωσσική επεξεργασία και προεπεξεργασία κειμένου

2.3.1 Τεμαχισμός σε λεκτικές μονάδες

Τα σύγχρονα μοντέλα παραγωγής φυσικής γλώσσας δεν επεξεργάζονται χαρακτήρες με τη μορφή που τους αντιλαμβάνονται οι χρήστες αλλά μια ακολουθία διακριτών συμβόλων (tokens), τα οποία προέρχονται από ένα πεπερασμένο λεξιλόγιο. Η διαδικασία μετατροπής του κειμένου σε αυτή την ακολουθία ονομάζεται τεμαχισμός ή tokenization και αποτελεί απαραίτητο στάδιο για την εισαγωγή του κειμένου στο μοντέλο [7].

Η πιο απλή προσέγγιση θα ήταν κάθε λέξη να αντιστοιχίζεται σε μια μονάδα. Η μέθοδος αυτή παρουσιάζει σημαντικούς περιορισμούς, καθώς η κάλυψη όλων των διαφορετικών μορφών μιας λέξης και, γενικότερα, του συνόλου των πιθανών λέξεων θα απαιτούσε ένα ιδιαίτερα μεγάλο λεξιλόγιο. Επιπλέον, κάθε λέξη που δεν παρατηρήθηκε κατά την εκπαίδευση παραμένει άγνωστη και αντιστοιχίζεται σε ένα ειδικό σύμβολο που δεν παρέχει πληροφορία για το περιεχόμενό της.

Για την επίλυση του προβλήματος οι λέξεις χωρίζονται σε μικρότερα τμήματα που ονομάζονται subword tokens. Ειδικότερα, πολλά τμήματα μπορούν να συνδυαστούν για να αναπαραστήσουν μια κανονική λέξη. Αλγόριθμοι που ακολουθούν αυτή την προσέγγιση είναι, μεταξύ άλλων, η κωδικοποίηση ζευγών ψηφιολέξεων, η μέθοδος WordPiece και η SentencePiece — διαφέρουν ως προς το κριτήριο με το οποίο επιλέγουν ποια τεμάχια θα συγχωνεύσουν, αλλά μοιράζονται την ίδια αρχή: το λεξιλόγιο μαθαίνεται από τα δεδομένα και κάθε συμβολοσειρά μπορεί να αναπαρασταθεί [8, 9].

Αυτό έχει ως συνέπεια ένα πολύγλωσσο μοντέλο να μη χρειάζεται ξεχωριστό λεξιλόγιο για να αναγνωρίσει κάθε γλώσσα όπως τα ελληνικά ακριβώς επειδή αναλύει τις λέξεις σε τεμάχια που έχουν παρατηρηθεί στο πολύγλωσσο σώμα εκπαίδευσης. Επίσης, μια ακόμη παρατήρηση που πρέπει να γίνει είναι ότι το μήκος ενός κειμένου σε λεκτικές μονάδες δεν συμπίπτει με το μήκος του σε χαρακτήρες, και ο λόγος των δύο εξαρτάται από τη γλώσσα: γλώσσες που δεν χρησιμοποιούν το λατινικό αλφάβητο και υποεκπροσωπούνται στο σώμα εκπαίδευσης τεμαχίζονται σε περισσότερες και μικρότερες μονάδες ανά λέξη. Δεδομένου ότι τα όρια χωρητικότητας των συστημάτων ανάκτησης και των γλωσσικών μοντέλων (context window) μετρώνται σε λεκτικές μονάδες, η απόκλιση αυτή επηρεάζει άμεσα την αποδοτικότητα του συστήματος [10].

2.3.2 Κανονικοποίηση και Unicode

Το πρότυπο Unicode αναλαμβάνει να αντιστοιχίσει κάθε χαρακτήρα σε ένα κωδικοποιημένο σύμβολο, διαφορετικό από αυτό που βλέπει ο χρήστης στο τελικό αποτέλεσμα. Αυτό έχει ιδιαίτερη σημασία καθώς μπορούν να δημιουργηθούν πολλοί συνδυασμοί με τον ίδιο χαρακτήρα. Για την ακρίβεια, ένα

φωνήεν μπορεί να αναπαρασταθεί ως ένα κωδικοποιημένο σύμβολο είτε ως ακολουθία του άτονου γράμματος και ενός συνδυαστικού διακριτικού. Παρόλο που οι δύο αναπαραστάσεις μπορεί να ταιριάζουν οπτικά, στην πραγματικότητα όμως διαφέρουν σε επίπεδο ακολουθίας χαρακτήρων [11].

Για τον λόγο αυτό, το Unicode ορίζει τέσσερις μορφές κανονικοποίησης, οι οποίες επιτρέπουν την αναγωγή κανονικά ή συμβατά ισοδύναμων ακολουθιών χαρακτήρων σε συνεπή μορφή. Ειδικότερα, για τη σύνθεση των χαρακτήρων αξιοποιείται η μορφή NFC ενώ για την αποσύνθεσή τους η μορφή NFD. Στον αντίποδα, οι μορφές NFKC και NFKD βασίζονται στη συμβατή ισοδυναμία (compatibility equivalence), οι οποίες εξομοιώνουν χαρακτήρες που ενδέχεται να διαφέρουν τυπογραφικά ή μορφολογικά αλλά διατηρούν κοινό σημασιολογικό περιεχόμενο. Η δεύτερη κατηγορία εφαρμόζει πιο επιθετική κανονικοποίηση και μπορεί να προκαλέσει απώλεια ορισμένων διαφορών της αρχικής αναπαράστασης. Για τον λόγο αυτό δεν προτιμάται όταν έχει σημασία η ακριβής μορφή του κειμένου [12].

Είναι σημαντικό να σημειωθεί ότι η κανονικοποίηση από μόνη της δεν είναι αρκετή, καθώς πρέπει να εφαρμόζεται με συνέπεια και συμμετρικά σε όλα τα στάδια της επεξεργασίας. Εν προκειμένω, στο πλαίσιο ενός συστήματος ανάκτησης πληροφοριών, το κείμενο που καταχωρίζεται στο ευρετήριο και το ερώτημα του χρήστη θα πρέπει να υποβάλλονται στην ίδια μορφή κανονικοποίησης. Σε διαφορετική περίπτωση, η αναντιστοιχία που η κανονικοποίηση υποτίθεται ότι εξαλείφει επανεισάγεται από την ίδια τη διαδικασία.

Ανάλογο ζήτημα προκύπτει και κατά τη διαδικασία της πεζοποίησης. Θεωρείται γενικά μια εύκολη και ασφαλής μετατροπή μεταξύ κεφαλαίων και πεζών, ωστόσο δεν είναι πάντα ένα προς ένα και εξαρτάται από το γραφικό σύστημα και τη γλώσσα. Στο ελληνικό αλφάβητο ειδικότερα, η διαδικασία αυτή μπορεί να οδηγήσει σε απώλεια του τονισμού με αποτέλεσμα να μην είναι δυνατό να ανακτηθεί με βεβαιότητα κατά την αντίστροφη μετατροπή σε πεζά.

2.3.3 Μορφολογία, στελέχωση και λημματοποίηση

Μια λέξη μπορεί να εμφανίζεται σε διαφορετικές μορφές όπως άλλους τύπους και κλίσεις οι οποίοι διατηρούν το βασικό νόημα και γίνονται εύκολα αντιληπτοί από τον αναγνώστη. Ωστόσο, ένα μηχανήμα δεν μπορεί να αντιληφθεί με τον ίδιο τρόπο τη σχέση των διαφορετικών τύπων όπως ένας άνθρωπος. Με αυτό ακριβώς το θέμα ασχολείται η μορφολογία, δηλαδή με τη μελέτη της δομής των λέξεων και τον τρόπο που σχηματίζονται από μορφήματα. Τα μορφήματα διακρίνονται σε δύο βασικές κατηγορίες: τα κλιτικά μορφήματα, τα οποία εκφράζουν γραμματικές κατηγορίες, όπως η πτώση, ο αριθμός και ο χρόνος, χωρίς να μεταβάλλουν τη λεξική κατηγορία της λέξης, και τα παραγωγικά μορφήματα, τα οποία χρησιμοποιούνται για τον σχηματισμό νέων λέξεων. Σε έναν μηχανισμό κάθε αλλαγή μορφήματος αντιμετωπίζεται διαφορετικά παρότι από γλωσσική άποψη αποτελούν μορφές της ίδιας λέξης.

Για το πρόβλημα αυτό υπάρχουν δύο διαφορετικές προσεγγίσεις. Η πρώτη είναι η στελέχωση, κατά την οποία ο αλγόριθμος κόβει τις καταλήξεις των λέξεων μηχανικά ακολουθώντας μόνο συγκεκριμένους κανόνες, με σκοπό να κρατήσει μόνο το βασικό τους στέλεχος. Ο αλγόριθμος του Porter αποτέλεσε μία από τις βασικές προσεγγίσεις αυτής της μεθόδου για την αγγλική γλώσσα [13] και αργότερα εντάχθηκε στην οικογένεια Snowball, η οποία παρέχει κανόνες στελέχωσης για πολλές γλώσσες μέσα από ένα κοινό πλαίσιο. Το αποτέλεσμα της μεθόδου προκύπτει αφαιρώντας τμήματα της λέξης χωρίς να είναι απαραίτητο να είναι υπαρκτή λέξη. Η λημματοποίηση, από την άλλη, ακολουθεί διαφορετική προσέγγιση καθώς μετατρέπει κάθε τύπο της λέξης στο αντίστοιχο λήμμα, δηλαδή στη λεξικογραφική

μορφή με την οποία θα αναζητούνταν σε ένα λεξικό. Ειδικότερα, παράγει ένα υπαρκτό λήμμα καθώς χρειάζεται τη χρήση λεξικών και σε πολλές περιπτώσεις γραμματικής ανάλυσης των συμφραζομένων. Εν κατακλείδι, η στελέχωση είναι η πιο απλή μέθοδος και απαιτεί λιγότερους πόρους καθώς εφαρμόζεται πιο εύκολα, ενώ η λημματοποίηση προσφέρει μεγαλύτερη γλωσσική ακρίβεια, αλλά εξαρτάται περισσότερο από τη διαθεσιμότητα κατάλληλων γλωσσικών πόρων [13, 14].

Είναι κρίσιμο να αναφερθεί ότι επειδή η στελέχωση εφαρμόζει κανόνες μηχανικά, οδηγεί συχνά σε δύο αντίρροπα σφάλματα. Το πρώτο είναι η υπερστελέχωση (over-stemming), όπου το σύστημα αποκόπτει υπερβολικά μεγάλο τμήμα, με αποτέλεσμα διαφορετικές λέξεις να καταλήγουν στο ίδιο στέλεχος και να συγχέονται μεταξύ τους. Το δεύτερο είναι η υποστελέχωση (under-stemming), όπου ο αλγόριθμος δεν αποκόπτει επαρκές τμήμα, με συνέπεια διαφορετικοί τύποι της ίδιας λέξης να μην αναγνωρίζονται ως όμοιοι [3].

2.3.4 Ιδιαιτερότητες της ελληνικής

Η εφαρμογή των παραπάνω μεθόδων στα ελληνικά μπορεί να περιπλέξει τη διαδικασία και να δυσκολέψει την παραμετροποίηση. Αυτό οφείλεται στο γεγονός ότι υπάρχουν ορισμένα χαρακτηριστικά της γλώσσας που επηρεάζουν άμεσα τον σχεδιασμό και τη λειτουργία ενός συστήματος ανάκτησης πληροφοριών.

Το πρώτο χαρακτηριστικό είναι το πλούσιο κλιτικό σύστημα της ελληνικής γλώσσας. Ένα ουσιαστικό μπορεί να έχει πολλούς συνδυασμούς και πιο συγκεκριμένα μπορεί να εμφανιστεί σε τέσσερις πτώσεις και δύο αριθμούς, ενώ ένα επίθετο συνδυάζεται επιπλέον με τα τρία γένη. Αντίστοιχα τα ρήματα έχουν πολλές μορφές που μπορεί να εμφανιστούν. Για έναν αλγόριθμο, διαφορετικοί μορφολογικοί τύποι, όπως «φοιτητής», «φοιτητή», «φοιτητές» και «φοιτητών» αντιμετωπίζονται ως διαφορετικοί όροι. Αυτό έχει ως αποτέλεσμα, η στατιστική βαρύτητα μιας λέξης να μπορεί να κατανέμεται στους διαφορετικούς κλιτικούς της τύπους, με συνέπεια ένα ερώτημα που περιέχει μία μορφή να μην αντιστοιχίζεται σε ένα έγγραφο που περιέχει κάποια άλλη.

Ένα ακόμη χαρακτηριστικό είναι ο τονισμός των λέξεων, ο οποίος πολλές φορές παραλείπεται κατά την πληκτρολόγηση, ιδιαίτερα σε ερωτήματα που εισάγονται γρήγορα ή από φορητές συσκευές. Έτσι, μία λέξη μπορεί να εμφανιστεί με διαφορετικές εκδοχές τονισμού, δηλαδή σε τονισμένη και άτονη μορφή. Το γεγονός αυτό μπορεί να δημιουργήσει αναντιστοιχίες μεταξύ του ερωτήματος του χρήστη και του κειμένου των εγγράφων, όταν αυτά χρησιμοποιούν διαφορετική μορφή.

Το τελικό σίγμα αποτελεί και αυτό ένα ιδιαίτερο χαρακτηριστικό που πρέπει να ληφθεί υπόψη. Το γράμμα σίγμα διαθέτει ειδική μορφή όταν βρίσκεται στο τέλος μιας λέξης («ς»), η οποία έχει διαφορετικό κωδικοσημείο από τη μορφή «σ». Επομένως, κατά το στάδιο της σύγκρισης καταλήξεων και της εφαρμογής κανόνων στελέχωσης πρέπει να λαμβάνεται υπόψη η διαφορετική αναπαράσταση του ίδιου φωνήματος ανάλογα με τη θέση του στη λέξη. Η εξομοίωση των δύο μορφών μπορεί επίσης να επηρεάσει την κατάληξη που αναγνωρίζει ένας μηχανισμός στελέχωσης.

Το τέταρτο ζήτημα αφορά τη χρήση λατινικών χαρακτήρων για την απόδοση ελληνικών λέξεων, πράγμα που συμβαίνει συχνά στην ηλεκτρονική επικοινωνία. Δεν υπάρχει ένα ενιαίο πρότυπο μεταγραφής λέξεων καθώς πολλά λατινικά γράμματα μπορεί να αντιστοιχίζονται με ελληνικά που έχουν παρόμοια μορφή. Η δυσκολία αυξάνεται όταν η μεταγραφή γίνεται άλλοτε φωνητικά και άλλοτε με βάση την οπτική ομοιότητα των χαρακτήρων, ώστε το ίδιο ελληνικό γράφημα να αποδίδεται διαφορετικά από πολλούς χρήστες.

Πέρα από τα δομικά αυτά χαρακτηριστικά, η ελληνική γλώσσα κατατάσσεται στις γλώσσες χαμηλότερων πόρων: τα διαθέσιμα σώματα κειμένων, τα σύνολα αξιολόγησης και τα προεκπαιδευμένα μοντέλα είναι σημαντικά λιγότερα από τα αντίστοιχα της αγγλικής. Αυτό σημαίνει ότι τα μοντέλα που υποστηρίζουν τη γλώσσα είναι πολύ περιορισμένα και η μέση απόδοση του δεν είναι ίδια και για την ελληνική [15].

Ιδιαίτερη περίπτωση διατύπωσης αποτελεί η χρήση λατινικών χαρακτήρων για την απόδοση ελληνικών λέξεων, πρακτική γνωστή ως greeklish. Δεν έχει τυποποιημένη αντιστοίχιση γραμμάτων καθώς ο ίδιος ελληνικός φθόγγος αποδίδεται σε πολλούς λατινικούς χαρακτήρες ανάλογα με το αν ακολουθείται οπτική ή φωνητική λογική.

Σε συστήματα που πραγματοποιούν ανάκτηση κειμένου αυτή η περίπτωση είναι κρίσιμη καθώς ένα ερώτημα γραμμένο με λατινικούς χαρακτήρες δεν παρουσιάζει καμία λεξική επικάλυψη με έγγραφα γραμμένα στο ελληνικό αλφάβητο, ανεξάρτητα από το πόσο μοιάζει σημασιολογικά. Μία λύση για αυτή τη περίπτωση θα ήταν η αναδιατύπωση του ερωτήματος πριν από την ανάκτηση μέσω ενός μηχανισμού μεταγραφής, είτε θα μπορούσε να χρησιμοποιηθεί ένα λεξικό αντιστοίχισης που δημιουργείται από το ίδιο το σώμα κειμένων, ώστε οι greeklish μορφές να αντιστοιχίζονται στις αντίστοιχες ελληνικές λέξεις.

2.3.5 Προσεγγιστική αντιστοίχιση συμβολοσειρών

Οι προηγούμενοι μηχανισμοί που περιεγράφηκαν προηγουμένως για να λειτουργήσουν σωστά προϋποθέτουν να μην κάνει κάποιο λάθος ο χρήστης κατά την πληκτρολόγηση. Ακόμη και με την κανονικοποίηση της λέξης χρειάζεται να υπάρχει στο σώμα κειμένων για να αντιστοιχιστεί διαφορετικά χρειάζεται να υπάρχει ένας μηχανισμός ανοχής.

Για τον σκοπό αυτό χρησιμοποιείται η απόσταση επεξεργασίας Levenshtein. Πιο συγκεκριμένα, ορίζεται ως το ελάχιστο πλήθος εισαγωγών, διαγραφών και αντικαταστάσεων χαρακτήρων που απαιτούνται για τη μετατροπή μιας συμβολοσειράς σε μια άλλη. Επειδή υπάρχει εξάρτηση μεταξύ της απόλυτης τιμής και του μήκους των λέξεων, συνήθως κανονικοποιείται ως προς αυτό, ώστε το ίδιο κατώφλι να είναι εφαρμόσιμο σε σύντομες και σε εκτενείς λέξεις.

Ο υπολογισμός της απόστασης Levenshtein ανάμεσα στην λέξη που γράφει ο χρήστης και σε κάθε λέξη του λεξιλογίου απαιτεί πολύ μεγάλο αριθμό συγκρίσεων και χρειάζεται αρκετό χρόνο για να ολοκληρωθεί όταν το λεξιλόγιο είναι μεγάλο. Για τον λόγο αυτό μπορεί να χρησιμοποιηθεί ένα δέντρο Burkhard-Keller (BK-tree), το οποίο οργανώνει τις λέξεις με βάση το πόσο διαφέρουν μεταξύ τους. Έτσι, το σύστημα μπορεί να βρει πολύ πιο άνετα μια λέξη που μοιάζει με αυτή στο λεξιλόγιο του. Επιπλέον, έχει την δυνατότητα να αποκλείει ολόκληρες ομάδες λέξεων που βρίσκονται έξω από το επιτρεπόμενο όριο απόστασης, μειώνοντας σημαντικά τον αριθμό των απαραίτητων συγκρίσεων.

2.4 Αναπαραστάσεις κειμένου

2.4.1 Λεξικές αναπαραστάσεις

Το μοντέλο του «σάκου λέξεων» (bag-of-words) αναπαριστά κάθε έγγραφο ως ένα διάνυσμα του οποίου οι διαστάσεις αντιστοιχούν στο συνολικό λεξιλόγιο. Κάθε διάσταση αντιστοιχεί σε έναν συγκεκριμένο όρο και η τιμή της μπορεί να εκφράζει είτε την παρουσία είτε τη συχνότητα εμφάνισής του. Η σειρά με την οποία εμφανίζονται οι λέξεις στο έγγραφο δεν λαμβάνεται υπόψη. Η αναπαράσταση είναι συνήθως αραιή, καθώς αναπαριστά μόνο ένα μικρό μέρος του λεξιλογίου [16].

Ωστόσο, η απλή συχνότητα εμφάνισης ενός όρου δεν συνεπάγεται ότι έχει και μεγάλη αξία. Συχνά επαναλαμβανόμενες λέξεις όπως, για παράδειγμα, το “και” ή το “είναι”, εμφανίζονται με υψηλή συχνότητα στην ελληνική γλώσσα και, συνεπώς, δεν συμβάλλουν σημαντικά στη διάκριση των εγγράφων μεταξύ τους. Σε τέτοιες περιπτώσεις εφαρμόζεται η αντίστροφη συχνότητα εγγράφου (Inverse Document Frequency – IDF), η οποία λαμβάνει υπόψη το πλήθος των εγγράφων στα οποία εμφανίζεται ένας όρος. Ειδικότερα, ένας όρος που εμφανίζεται σε μικρότερο αριθμό εγγράφων έχει μεγαλύτερο βάρος, καθώς θεωρείται ότι προσφέρει περισσότερη διακριτική πληροφορία από έναν όρο που εμφανίζεται σχεδόν σε ολόκληρο το σώμα κειμένων.

Η συνάρτηση κατάταξης BM25 συνδυάζει τη συχνότητα εμφάνισης ενός όρου με την αντίστροφη συχνότητα εγγράφου και χρησιμοποιεί τις παραμέτρους k_1 και b για τον έλεγχο του κορεσμού της συχνότητας και της κανονικοποίησης ως προς το μήκος του εγγράφου. Για ένα ερώτημα q με όρους t_i και ένα έγγραφο d , η βαθμολογία υπολογίζεται ως [17]:

$$\text{score}(q, d) = \sum_{i=1}^{|q|} \text{IDF}(t_i) \cdot \frac{f(t_i, d) \cdot (k_1 + 1)}{f(t_i, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)} \quad (2.1)$$

όπου $f(t_i, d)$ είναι η συχνότητα εμφάνισης του όρου t_i στο έγγραφο d , $|q|$ το πλήθος των όρων του ερωτήματος, $|d|$ το μήκος του εγγράφου και avgdl το μέσο μήκος των εγγράφων του σώματος κειμένων.

Οι δύο παράμετροι της συνάρτησης επηρεάζουν διαφορετικά τον τρόπο με τον οποίο υπολογίζεται η βαθμολογία. Η k_1 ελέγχει τον κορεσμό της συχνότητας εμφάνισης ενός όρου. Έτσι, οι πρώτες εμφανίσεις ενός όρου έχουν μεγαλύτερη επίδραση από τις υπόλοιπες, καθώς η συνεισφορά των επόμενων εμφανίσεων μειώνεται σταδιακά. Αντίστοιχα, η b χρησιμοποιείται για την κανονικοποίηση του μήκους των εγγράφων. Σε ένα μεγαλύτερο έγγραφο είναι πιθανότερο να εμφανίζεται ένας όρος περισσότερες φορές απλώς λόγω του μεγέθους του χωρίς αυτό να σημαίνει απαραίτητα ότι το έγγραφο είναι περισσότερο συναφές με το ερώτημα. Η παράμετρος b περιορίζει επομένως αυτή την επίδραση λαμβάνοντας υπόψη την απόκλιση του μήκους του εγγράφου από το μέσο μήκος του σώματος.

Η λεξική ανάκτηση είναι ιδιαίτερα αποτελεσματική σε περιπτώσεις όπου απαιτείται ακριβής αντιστοίχιση όρων, όπως σε κωδικούς μαθημάτων, αριθμούς νόμων, ακρωνύμια ή επώνυμα. Σε αυτές τις περιπτώσεις η παρουσία του συγκεκριμένου όρου αποτελεί ισχυρή ένδειξη συνάφειας. Αντίθετα, η λεξική ανάκτηση παρουσιάζει περιορισμένη αποτελεσματικότητα όταν το ερώτημα και το έγγραφο εκφράζουν την ίδια έννοια με διαφορετικές λέξεις, όπως αναφέρθηκε στην υποενότητα 2.2.1 σχετικά με το λεξιλογικό χάσμα. Επιπλέον, η ίδια λέξη μπορεί να χρησιμοποιείται με διαφορετική σημασία ανάλογα με το πλαίσιο, κάτι που επίσης δεν μπορεί να αντιμετωπιστεί αποτελεσματικά μόνο μέσω της λεξικής αντιστοίχισης.

Τέλος, δεν μπορεί να παραλειφθεί η συνηθισμένη πρακτική της αφαίρεσης των λέξεων υψηλής συχνότητας (stop-words). Βασίζεται στη λογική ότι αυτές οι λέξεις δεν έχουν διακριτική ικανότητα, παρόλα αυτά η διαγραφή τους μπορεί να επιφέρει κινδύνους ανάλογα με το είδος του κειμένου. Για παράδειγμα, σε κανονιστικά ή νομικά κείμενα, οι προθέσεις έχουν τεράστια σημασία για τη σωστή ερμηνεία μιας διάταξης. Εφόσον ο υπολογισμός IDF αποδίδει ανεξαρτήτως χαμηλό βάρος σε αυτούς τους όρους, η πλήρης αφαίρεσή τους αυξάνει τον κίνδυνο χωρίς να προσφέρει κάποιο ουσιαστικό πλεονέκτημα στην αναζήτηση [3].

2.4.2 Πυκνές αναπαραστάσεις

Οι πυκνές αναπαραστάσεις (dense representations) βασίζονται στην κατανομητική υπόθεση, η οποία ορίζει ότι λέξεις που εμφανίζονται σε παρόμοια συμφραζόμενα τείνουν να έχουν παρόμοια σημασία. Αν η υπόθεση ισχύει, το νόημα μιας λέξης μπορεί να προσεγγιστεί από την κατανομή των συμφραζομένων της και επομένως μπορεί να αναπαρασταθεί σε έναν πυκνό διανυσματικό χώρο, στον οποίο η σχετική θέση των διανυσμάτων αντικατοπτρίζει σημασιολογικές σχέσεις [18].

Οι πρώτες μέθοδοι αυτής της κατηγορίας βασίζονταν σε στατικές αναπαραστάσεις, στις οποίες κάθε λέξη αντιστοιχιζόταν σε ένα σταθερό διάνυσμα. Πιο συγκεκριμένα, αυτό επιτυγχανόταν είτε προσπαθώντας να προβλέψουν τις γειτονικές λέξεις είτε μέσω παραγοντοποίησης ενός πίνακα συνεμφανίσεων [19, 20]. Ως αποτέλεσμα, μία πολυσημική λέξη, δηλαδή μία λέξη με δύο ή περισσότερες σημασίες, αποκτούσε ένα μόνο διάνυσμα, το οποίο αποτελούσε ουσιαστικά έναν συμβιβασμό μεταξύ των διαφορετικών σημασιών της. Το πρόβλημα αυτό λύθηκε χρησιμοποιώντας τις αναπαραστάσεις βάσει συμφραζομένων (contextualized embeddings), οι οποίες παράγουν δυναμικά ένα διαφορετικό διάνυσμα για κάθε εμφάνιση της λέξης, ανάλογα με την πρόταση στην οποία εντάσσεται.

Για την εύρεση της ομοιότητας μεταξύ δύο διανυσμάτων χρησιμοποιείται η ομοιότητα συνημιτόνου (cosine similarity), η οποία υπολογίζεται από τον ακόλουθο τύπο:

$$\cos(u, v) = \frac{u \cdot v}{\|u\| \|v\|} \quad (2.2)$$

Η συγκεκριμένη μέτρηση βασίζεται στη γωνία μεταξύ των δύο διανυσμάτων και όχι στο απόλυτο μέγεθός τους. Αυτό είναι χρήσιμο σε περιπτώσεις σύγκρισης κειμένων διαφορετικής έκτασης, καθώς η ομοιότητα εξαρτάται κυρίως από την κατεύθυνση των διανυσμάτων. Όταν τα διανύσματα κανονικοποιούνται ώστε να έχουν μοναδιαίο μήκος, η ομοιότητα συνημιτόνου ισοδυναμεί με το εσωτερικό τους γινόμενο, γεγονός που απλοποιεί τον σχετικό υπολογισμό.

Ωστόσο, για την ανάκτηση πληροφοριών δεν αρκεί η αναπαράσταση μεμονωμένων λέξεων, καθώς απαιτείται ένα διάνυσμα που αντιπροσωπεύει το νόημα ολόκληρου του κειμένου. Η απλή συγκέντρωση των διανυσμάτων για κάθε λέξη μειώνει την απόδοση του αλγόριθμου, καθώς τα μοντέλα δεν έχουν εκπαιδευτεί ώστε το αποτέλεσμα αυτής της συγκέντρωσης να αποτελεί κατάλληλη σημασιολογική αναπαράσταση. Για αυτό τον σκοπό χρησιμοποιούνται μοντέλα ενσωμάτωσης προτάσεων, στα οποία ο κωδικοποιητής εκπαιδεύεται ώστε κείμενα με παρόμοιο νόημα να αποκτούν κοντινά διανύσματα, ενώ κείμενα που δεν σχετίζονται να τοποθετούνται σε μεγαλύτερη απόσταση [21, 22].

Ο τρόπος με τον οποίο αντιμετωπίζονται το ερώτημα και το κείμενο προς ανάκτηση από αυτά τα μοντέλα είναι ιδιαίτερα αξιοσημείωτος. Αυτό συμβαίνει διότι μερικές φορές εκπαιδεύονται με ασύμμετρο τρόπο με αποτέλεσμα το ερώτημα και το απόσπασμα να επεξεργάζονται διαφορετικά ή να χρησιμοποιούν διαφορετικά προθέματα που δηλώνουν τον ρόλο τους. Άλλα μοντέλα ακολουθούν συμμετρική προσέγγιση και επεξεργάζονται και τα δύο κείμενα με τον ίδιο τρόπο. Στα ασύμμετρα μοντέλα, η λανθασμένη τοποθέτηση ή παράλειψη του κατάλληλου προθέματος μπορεί να επηρεάσει την τελική θέση του κειμένου στο διανυσματικό χώρο και να οδηγήσει σε ανάκτηση χαμηλότερης ποιότητας, χωρίς να εμφανιστεί κάποιο σφάλμα κατά την εκτέλεση.

Παράλληλα είναι σημαντικό να σημειωθεί ότι τα πολύγλωσσα μοντέλα τα οποία εκπαιδεύονται σε δεδομένα από διαφορετικές γλώσσες, μπορούν να αναπαριστούν κείμενα διαφορετικών γλωσσών σε έναν κοινό διανυσματικό χώρο. Με αυτόν τον τρόπο, λέξεις που έχουν παρόμοιο ή ίδιο νόημα ακόμη και όταν είναι γραμμένες σε διαφορετικές γλώσσες, μπορούν να αποκτούν κοντινές αναπαραστάσεις. Ωστόσο δεν συνεπάγεται ότι η απόδοσή τους κατανέμεται ομοιόμορφα για όλες τις γλώσσες, καθώς επηρεάζεται από την ποσότητα και την ποιότητα των δεδομένων που χρησιμοποιήθηκαν για κάθε γλώσσα κατά την εκπαίδευση. Κατά συνέπεια, η μέση συνολική επίδοση ενός τέτοιου μοντέλου δεν αποτελεί εγγύηση για το πόσο αποτελεσματικά θα λειτουργήσει σε μια συγκεκριμένη γλώσσα, όπως η ελληνική [10, 23].

2.5 Μεγάλα γλωσσικά μοντέλα

2.5.1 Ο μετασχηματιστής

Η αρχιτεκτονική του μετασχηματιστή «transformers» αποτέλεσε τομή στην επεξεργασία ακολουθιών και περιόρισε σημαντικά τη χρήση των αναδρομικών νευρωνικών δικτύων (RNNs). Το στοιχείο που την έκανε να ξεχωρίζει έναντι άλλων αρχιτεκτονικών είναι η αντικατάσταση της σειριακής επεξεργασίας με την ταυτόχρονη συσχέτιση κάθε στοιχείου της ακολουθίας με τα υπόλοιπα στοιχεία [24].

Βασικό χαρακτηριστικό αυτής της αρχιτεκτονικής είναι ο μηχανισμός της αυτοπροσοχής (self-attention). Πιο αναλυτικά, παράγονται τρία διανύσματα για κάθε στοιχείο της εισόδου: ένα ερώτημα (query - Q), ένα κλειδί (key - K) και μία τιμή (value - V). Η συνάφεια μεταξύ δύο στοιχείων υπολογίζεται μέσω του εσωτερικού γινομένου του ερωτήματος του ενός με το κλειδί του άλλου. Στη συνέχεια, οι παραγόμενες συνάφειες κανονικοποιούνται σε μια κατανομή βαρών και η νέα αναπαράσταση κάθε στοιχείου προκύπτει ως ο σταθμισμένος συνδυασμός των τιμών όλων των στοιχείων, βάσει του εξής τύπου:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.3)$$

όπου d_k είναι η διάσταση των διανυσμάτων Key. Η διαίρεση με τη $\sqrt{d_k}$ περιορίζει την αύξηση των τιμών του εσωτερικού γινομένου για μεγαλύτερες διαστάσεις και βοηθά στην αποφυγή κορεσμού της softmax. Επιπροσθέτως, η προσοχή πολλαπλών κεφαλών (multi-head attention) εκτελεί τον παραπάνω υπολογισμό παράλληλα, σε πολλούς διαφορετικούς υποχώρους, επιτρέποντας έτσι στο μοντέλο να αποτυπώνει ταυτόχρονα διαφορετικά είδη σημασιολογικών και συντακτικών σχέσεων.

Ο μηχανισμός αυτοπροσοχής δεν αναγνωρίζει εγγενώς τη σειρά των στοιχείων, παρά μόνο διακρίνει ταυτόχρονα όλες τις λέξεις και τις συγκρίνει. Η πληροφορία της θέσης εισάγεται ρητά μέσω της κωδικοποίησης θέσης (positional encoding), η οποία προστίθεται στις αρχικές αναπαραστάσεις εισόδου. Ωστόσο, υπάρχει όριο στον όγκο δεδομένων που μπορεί να επεξεργαστεί ο μηχανισμός, καθώς η υπολογιστική πολυπλοκότητα της αυτοπροσοχής αυξάνεται τετραγωνικά ως προς το μήκος της ακολουθίας. Για αυτόν τον λόγο, είναι αναγκαία η εισαγωγή ενός παραθύρου συμφραζομένων (context window), το οποίο αποτελεί έναν αυστηρό σχεδιαστικό περιορισμό για τον όγκο δεδομένων που μπορούν να τροφοδοτηθούν στα συστήματα RAG.

Η αρχική αρχιτεκτονική Transformer συνδύαζε έναν κωδικοποιητή (encoder) και έναν αποκωδικοποιητή (decoder). Μετά από αυτό το σημείο προέκυψαν δύο ξεχωριστές οικογένειες μοντέλων με διακριτές χρήσεις. Τα μοντέλα “μόνο-κωδικοποιητή” (encoder-only) επεξεργάζονται ολόκληρη την είσοδο και χρησιμοποιούνται κυρίως σε εργασίες κατανόησης και αναπαράστασης

κειμένου, όπως η ταξινόμηση και η παραγωγή διανυσματικών αναπαραστάσεων. Αντίθετα, τα μοντέλα «μόνο-αποκωδικοποιητή» (decoder-only) παράγουν κείμενο διαδοχικά (αυτοπαλινδρομικά) και αποτελούν τη θεμελιώδη βάση των σύγχρονων μεγάλων γλωσσικών μοντέλων (LLMs). Η διάκριση αυτή δεν είναι απλώς δομική, αλλά καθαρά λειτουργική, γιατί καθορίζει ποιες διεργασίες μπορούν να προϋπολογιστούν και ποιες απαιτούν δυναμική παραγωγή [21, 25].

2.5.2 Προεκπαίδευση και προσαρμογή

Η εκπαίδευση ενός μεγάλου γλωσσικού μοντέλου απαιτεί πολλά στάδια περιλαμβάνει πολλά στάδια, μέσα από τα οποία το μοντέλο διαμορφώνεται σταδιακά. Το μοντέλο αρχικά μαθαίνει να προβλέπει την επόμενη λεκτική μονάδα (token) που θα παραχθεί κατά το στάδιο της προεκπαίδευσης (pre-training) πάνω σε μεγάλης κλίμακας, μη επισημειωμένα σώματα κειμένου. Στη συνέχεια, ακολουθεί η εποπτευόμενη προσαρμογή (Supervised Fine-Tuning - SFT) σε επιλεγμένα ζεύγη οδηγιών και επιθυμητών αποκρίσεων. Πολλές φορές το τελικό στάδιο είναι η ενισχυτική μάθηση μέσω ανθρώπινης ανατροφοδότησης (RLHF), όπου οι αποκρίσεις του μοντέλου ευθυγραμμίζονται με τις ανθρώπινες προτιμήσεις [26].

Είναι σημαντικό να αναφερθεί ότι η ικανότητα του μοντέλου να ακολουθεί πιστά τις οδηγίες που του δόθηκαν διαμορφώνεται από τα ενδιάμεσα στάδια και όχι αυτό της προεκπαίδευσης. Πρόκειται, συνεπώς, για μια ιδιότητα εντελώς ανεξάρτητη από τη γλωσσική του επάρκεια: δύο μοντέλα μπορεί να διαθέτουν την ίδια εξαιρετική γνώση μιας γλώσσας, αλλά να επιδεικνύουν εντελώς διαφορετική ικανότητα στην τήρηση κανόνων. Για ένα σύστημα RAG, το οποίο απαιτεί από το μοντέλο να απαντά αποκλειστικά βάσει του παρεχόμενου κειμένου και να αρνείται να απαντήσει όταν αυτό δεν επαρκεί (προς αποφυγή παραισθήσεων), η δεύτερη ιδιότητα της πειθαρχίας είναι εξίσου καθοριστική με την πρώτη [27].

2.5.3 Παραγωγή κειμένου και παράμετροι αποκωδικοποίησης

Η παραγωγή κειμένου χαρακτηρίζεται αυτοπαλινδρομη όταν το μοντέλο υπολογίζει την κατανομή πιθανοτήτων για την παραγωγή του επόμενου token βάσει όλων των προηγούμενων. Στη συνέχεια, επιλέγεται ένα token κατόπιν εισάγεται στην ακολουθία και επαναλαμβάνει τη διαδικασία με την επανξιμένη είσοδο [25].

Ειδικότερα, στο τελευταίο στάδιο του δικτύου παράγεται ένα διάνυσμα πραγματικών αριθμών με τόσες συνιστώσες όσες και οι μονάδες των λέξεων του. Οι τιμές αυτές δεν αποτελούν απευθείας πιθανότητες, αλλά μετατρέπονται σε κατανομή μέσω της συνάρτησης softmax, η οποία τις εκθετικοποιεί και τις κανονικοποιεί ώστε να αθροίζουν στη μονάδα. Οι παράμετροι που ακολουθούν δεν επηρεάζουν τον τρόπο που «σκέφτεται» το μοντέλο αλλά παρεμβαίνουν ακριβώς στο τελευταίο βήμα: εφαρμόζονται αφού το δίκτυο έχει βαθμολογήσει τις υποψήφιες λέξεις, αλλά λίγο πριν κάνει την τελική του επιλογή.

Οι δύο φάσεις παραγωγής

Η διαδικασία αποτελείται από δύο φάσεις με διαφορετικό υπολογιστικό προφίλ. Η πρώτη είναι η προπλήρωση (prefill) και έχει τη δυνατότητα να επεξεργάζεται ολόκληρη την είσοδο. Επειδή όλες οι μονάδες της εισόδου είναι ήδη γνωστές, ο μηχανισμός προσοχής μπορεί να υπολογιστεί για όλες τις θέσεις ταυτόχρονα μέσω πράξεων πινάκων. Είναι αντιληπτό ότι ανάλογα με το μήκος της εισόδου είναι αντίστοιχη και η υπολογιστική ισχύς που απαιτείται για να παραχθεί ένα αποτέλεσμα καθώς οι πόροι δεσμεύονται από την αρχή. Η αποκωδικοποίηση (decoding), αντίθετα, παράγει μία μονάδα κάθε φορά και δεν μπορεί να παραλληλοποιηθεί, καθώς κάθε βήμα προϋποθέτει το αποτέλεσμα του προηγούμενου. Κατά το στάδιο παραγωγής διαβάζονται σε κάθε βήμα οι παράμετροι του μοντέλου με αποτέλεσμα ο

επεξεργαστής να παραμένει σε μεγάλο ποσοστό αδρανής αναμένοντας δεδομένα, αυτό απαιτεί μεγάλο εύρος ζώνης της μνήμης (memory bandwidth).

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί η σημασία που έχει η κρυφή μνήμη των κλειδιών-τιμών (KV Cache) σε αυτή τη διαδικασία. Κατά το στάδιο αποκωδικοποίησης, ο μηχανισμός αυτοπροσοχής χρειάζεται τα κλειδιά και τις τιμές όλων των προηγούμενων θέσεων, όμως τα μεγέθη αυτά δεν εξαρτώνται από τη νέα μονάδα και δεν μεταβάλλονται. Επιπλέον, αν δεν αποθηκευτούν, θα πρέπει να επανυπολογίζονται σε κάθε βήμα, καθιστώντας το κόστος παραγωγής τετραγωνικό αντί για γραμμικό. Για αυτό η αποθήκευσή τους καθίσταται αναγκαία καθώς μετατρέπει τον επανυπολογισμό σε απλή ανάγνωση αντί για συνεχή επανυπολογισμό. Μια χρήση αυτής της μεθόδου είναι ότι ένα σταθερό αρχικό τμήμα προτροπής (όπως οι γενικές οδηγίες του συστήματος) μπορεί να επεξεργαστεί μία φορά και το αποτέλεσμα να επαναχρησιμοποιηθεί στα επόμενα αιτήματα [28].

Η επιλογή της επόμενης μονάδας

Η άπληστη επιλογή (greedy decoding) διαλέγει το επόμενο token με βάση αυτό που έχει τη μεγαλύτερη πιθανότητα. Είναι μια απόλυτα ντετερμινιστική μέθοδος, ωστόσο η βελτιστοποίηση που προσφέρει είναι τοπική. Μια άλλη εναλλακτική προσέγγιση είναι η δειγματοληψία (sampling) από την κατανομή, η οποία εισάγει τυχαιότητα. Η βασική παράμετρος που ρυθμίζει αυτή τη μορφή είναι η θερμοκρασία (Temperature - T). Πριν από τη συνάρτηση softmax, οι παραγόμενες τιμές (z) διαιρούνται με τη θετική σταθερά T [29]:

$$p(t_i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \quad (2.4)$$

Επειδή η εκθετική συνάρτηση δεν είναι γραμμική, η διαίρεση αυτή μεταβάλλει τη μορφή της κατανομής. Για $T < 1$, οι διαφορές μεταξύ των τιμών μεγεθύνονται, η κατανομή οξύνεται και η πιθανότητα συγκεντρώνεται στις επικρατέστερες μονάδες. Στο όριο, καθώς η θερμοκρασία τείνει στο μηδέν, η δειγματοληψία εκφυλίζεται στην άπληστη επιλογή — γεγονός που εξηγεί γιατί η μηδενική θερμοκρασία χρησιμοποιείται συμβατικά ως δήλωση ντετερμινιστικής παραγωγής.

Πέραν της θερμοκρασίας, μέθοδοι όπως το top-k και το top-p χρησιμοποιούνται για να περιορίσουν τη δειγματοληψία στις πιθανότερες επιλογές, εξαλείφοντας την "ουρά" της κατανομής. Το top-k διατηρεί σταθερά τις k επικρατέστερες μονάδες αγνοώντας τη μορφή της κατανομής, ενώ το top-p θέτει ένα κατώφλι αθροιστικής πιθανότητας, επιτρέποντας στο πλήθος των υποψηφίων να προσαρμόζεται δυναμικά: αν η κατανομή είναι οξεία το σύνολο συρρικνώνεται, ενώ αν είναι επίπεδη, διευρύνεται. Σε περιπτώσεις, ωστόσο, όπου ο ντετερμινισμός αποτελεί βασική προϋπόθεση, αποφεύγεται η χρήση τους, καθώς η σταθερή επιλογή της πιθανότερης λέξης τις καθιστά πρακτικά ανενεργές [29, 30].

Ποινή επανάληψης

Η ποινή επανάληψης (repetition penalty) μειώνει την πιθανότητα των μονάδων που έχουν ήδη εμφανιστεί στην ακολουθία, εφαρμόζοντας έναν μειωτικό συντελεστή στις αντίστοιχες τιμές πριν από τη softmax [31].

Ο μηχανισμός αυτός αποδεικνύεται χρήσιμος σε περιπτώσεις που το μοντέλο εγκλωβίζεται σε ατέρμονους βρόγχους, ωστόσο αντενδείκνυται για εργασίες εξαγωγής πληροφορίας από δεδομένο κείμενο. Σε τέτοια συστήματα, οι απαντήσεις απαιτούν συχνά την επανάληψη συγκεκριμένων λέξεων ή φράσεων (π.χ. ενός ονόματος ή ενός κωδικού) προκειμένου να αποδοθεί το σωστό νόημα. Η επιβολή ποινής επανάληψης, όμως, αναγκάζει το μοντέλο να επιλέξει διαφορετική λέξη. Συνεπώς, το μοντέλο

δεν παραλείπει απλώς τον όρο, αλλά τον υποκαθιστά με άλλον, συνήθως σημασιολογικά γειτονικό. Το τελικό αποτέλεσμα είναι ένα κείμενο που μεν διατηρεί τη ρευστότητα και τη φυσική του ροή, αλλά υστερεί σε ποιότητα ως προς το περιεχόμενό του, εισάγοντας ένα εξαιρετικά δυσδιάκριτο σφάλμα ουσίας.

Η μνήμη επομένως είναι ο καθοριστικός παράγοντας για τις δυνατότητες του συστήματος που μπορεί να υλοποιηθεί. Το συνολικό αποτύπωμα μνήμης ενός μοντέλου μπορεί να προσεγγιστεί από το γινόμενο του πλήθους των παραμέτρων επί τον αριθμό των ψηφίων ανά παράμετρο, προσαναξημένο κατά τον χώρο που απαιτεί η κρυφή μνήμη κλειδιών-τιμών (KV Cache), ο οποίος αυξάνεται γραμμικά με το μήκος της ακολουθίας.

2.5.4 Κβαντισμός και απαιτήσεις εκτέλεσης

Οι παράμετροι ή αλλιώς τα βάρη ενός νευρωνικού δικτύου αποθηκεύονται σε μορφή κινητής υποδιαστολής 16 δυαδικών ψηφίων (FP16/BF16). Ο κβαντισμός (quantization) μειώνει αυτή την ακρίβεια σε οκτώ (8 bit) ή τέσσερα bit (4 bit), αντιστοιχίζοντας τις τιμές των βαρών σε ένα μικρότερο σύνολο διακριτών επιπέδων. Το άμεσο και προφανές όφελος είναι η μείωση της απαιτούμενης μνήμης [32, 33].

Η μείωση του μεγέθους των παραμέτρων μπορεί να επηρεάσει θετικά και την ταχύτητα εκτέλεσης. Κατά τη διάρκεια της αποκωδικοποίησης, μεταφέρονται συνεχώς οι παράμετροι του μοντέλου, επομένως ο μικρότερος όγκος δεδομένων μπορεί να περιορίσει το κόστος αυτών των μεταφορών και να επιταχύνει την παραγωγή κειμένου. Ωστόσο, μειώνοντας την αριθμητική ακρίβεια επηρεάζεται και η ποιότητα του μοντέλου. Η μείωση της αριθμητικής ακρίβειας δεν συνεπάγεται απαραίτητα αντίστοιχη υποβάθμιση της ποιότητας του μοντέλου, καθώς το αποτέλεσμα εξαρτάται από παράγοντες όπως το ίδιο το μοντέλο και η μέθοδος κβαντισμού που χρησιμοποιείται. Επομένως, η επιλογή του κατάλληλου επιπέδου κβαντισμού αποτελεί κυρίως εμπειρική διαδικασία, καθώς απαιτείται αξιολόγηση με γνώμονα τόσο την απαιτούμενη μνήμη και την ταχύτητα εκτέλεσης, όσο και την ποιότητα του παραγόμενου κειμένου [34].

Σε επίπεδο συστήματος, οι απαιτήσεις μνήμης αποτελούν βασικό περιορισμό για την τοπική εκτέλεση ενός γλωσσικού μοντέλου. Η συνολική μνήμη που απαιτείται δεν εξαρτάται μόνο από το μέγεθος των παραμέτρων, αλλά και από τον χώρο που δεσμεύει η κρυφή μνήμη κλειδιών-τιμών (KV Cache), ο οποίος αυξάνεται ανάλογα με το μήκος της ακολουθίας. Παράλληλα, πρέπει να λαμβάνεται υπόψη η αρχιτεκτονική μνήμης του συστήματος. Σε συστήματα με διακριτή μνήμη, η μνήμη του επιταχυντή διαχωρίζεται από τη μνήμη του επεξεργαστή, ενώ σε αρχιτεκτονικές ενοποιημένης μνήμης τα δύο υποσυστήματα χρησιμοποιούν κοινό χώρο μνήμης. Η δεύτερη αρχιτεκτονική ευνοεί την κοινή χρήση των δεδομένων, αλλά η διαθέσιμη μνήμη αποτελεί κοινό και περιορισμένο πόρο.

2.5.5 Παραισθήσεις

Οι παραισθήσεις είναι ένα φαινόμενο των γλωσσικών μοντέλων που το παραγόμενο περιεχόμενο παρουσιάζεται ως έγκυρο, χωρίς να διαθέτει την απαραίτητη τεκμηρίωση ή να ανταποκρίνεται στα διαθέσιμα δεδομένα. Η πρώτη αφορά πληροφορίες που δεν έχουν σχέση με την πραγματικότητα, ενώ η δεύτερη αφορά περιπτώσεις όπου το μοντέλο παράγει μια απάντηση που δεν συμφωνεί με την παρεχόμενη πηγή ή αποκλίνει από το περιεχόμενο που του δόθηκε. Η δεύτερη περίπτωση είναι ιδιαίτερα σημαντική για συστήματα RAG, καθώς μπορεί να εμφανιστεί ακόμη και όταν η απαιτούμενη πληροφορία υπάρχει ήδη στο ανακτημένο κείμενο [35].

Αυτό το φαινόμενο οφείλεται στον τρόπο με τον οποίο εκπαιδεύονται τα γλωσσικά μοντέλα. Πιο συγκεκριμένα, μαθαίνουν να αποδίδουν υψηλή πιθανότητα σε ακολουθίες tokens που είναι πιθανές με βάση τα δεδομένα εκπαίδευσης και όχι να ελέγχουν άμεσα αν κάθε πληροφορία που παράγει είναι αληθής. Μ. Οι δύο στόχοι μπορεί σε ορισμένες περιπτώσεις να οδηγούν στο ίδιο αποτέλεσμα, χωρίς όμως να ταυτίζονται. Ως αποτέλεσμα, το μοντέλο μπορεί να παράγει μια γλωσσικά φυσική και πειστική διατύπωση χωρίς να βασίζεται στα δεδομένα και στην πραγματικότητα [36].

Μια πιθανή λύση για τον μετριασμό αυτού του φαινομένου είναι η αύξηση του μεγέθους των γλωσσικών μοντέλων. Ωστόσο, το πρόβλημα δεν εξαλείφεται τελείως. Παράλληλα, όσο πιο μεγάλο και ανεπτυγμένο είναι ένα μοντέλο, τόσο πιο συνεκτικές και πειστικές απαντήσεις παράγει, με αποτέλεσμα οι παραισθήσεις που ενδεχομένως προκύπτουν να μην είναι πάντοτε εύκολο να αναγνωριστούν από τον χρήστη.

Η επιλογή ενός ισχυρότερου γλωσσικού μοντέλου δεν αρκεί από μόνο της για να αντιμετωπιστούν οι παραισθήσεις. Για τον περιορισμό των παραισθήσεων απαιτείται η παροχή σχετικών πηγών κατά την εκτέλεση του συστήματος, η παραγωγή απάντησης με βάση το ανακτημένο περιεχόμενο, καθώς και η ικανότητα του συστήματος να αρνείται να απαντήσει σε περιπτώσεις αβεβαιότητας (όταν οι διαθέσιμες πληροφορίες δεν επαρκούν). Με τον τρόπο αυτό η αξιοπιστία της τελικής απάντησης δεν καθορίζεται αποκλειστικά από το γλωσσικό μοντέλο, αλλά και από τον τρόπο με τον οποίο αυτό εντάσσεται στη συνολική αρχιτεκτονική του συστήματος [1].

2.6 Ανάκτηση πληροφορίας

2.6.1 Αραιή και πυκνή ανάκτηση

Η αραιή ανάκτηση (sparse retrieval) στηρίζεται στις λεξικές αναπαραστάσεις της ενότητας 2.4.1. Το βασικό στοιχείο της είναι το αντεστραμμένο ευρετήριο (inverted index), το οποίο αντιστοιχίζει κάθε όρο του λεξιλογίου με τα έγγραφα στα οποία εμφανίζεται. Κατά το στάδιο της αναζήτησης κάθε όρος του ερωτήματος εξετάζεται ξεχωριστά και στη συνέχεια βαθμολογούνται τα αντίστοιχα έγγραφα που περιέχουν αυτούς τους όρους. Με τον τρόπο αυτό αποφεύγεται η χρονοβόρα αναζήτηση ολόκληρου του σώματος κειμένων για κάθε ερώτημα [3]. Η πυκνή ανάκτηση από την άλλη, βασίζεται στις διανυσματικές αναπαραστάσεις της ενότητας 2.4.2. Πιο συγκεκριμένα τα έγγραφα μετατρέπονται εξ αρχής σε διανύσματα, ενώ το ερώτημα του χρήστη μετατρέπεται δυναμικά σε διάνυσμα κατά τον χρόνο εκτέλεσης. Η διαδικασία της ανάκτησης πραγματοποιείται βάσει της ομοιότητας μεταξύ των διανυσμάτων και αποσκοπεί στον εντοπισμό των εγγράφων που βρίσκονται πλησιέστερα στο ερώτημα [37].

Ανάλογα την περίπτωση και τον σκοπό χρησιμοποιείται διαφορετική μέθοδος ανάκτησης. Η αραιή ανάκτηση είναι ιδιαίτερα αποτελεσματική όταν απαιτείται ακριβής αντιστοίχιση όρων, όπως κωδικοί μαθημάτων, ονόματα, αριθμοί νόμων. Ωστόσο, παρουσιάζει χαμηλότερη αποτελεσματικότητα όταν το ερώτημα και το σχετικό έγγραφο χρησιμοποιούν διαφορετικές λέξεις για την ίδια έννοια. Η πυκνή ανάκτηση μπορεί να αντιμετωπίσει καλύτερα τέτοιες περιπτώσεις, καθώς βασίζεται στη σημασιολογική ομοιότητα, αλλά μπορεί να παρουσιάσει χαμηλότερη αποτελεσματικότητα όταν η αναζήτηση βασίζεται σε συγκεκριμένες συμβολοσειρές, όπως κωδικοί ή αριθμοί. Τα πλεονεκτήματα και των δύο μπορούν να αξιοποιηθούν ακόμη περισσότερο όταν λειτουργούν συνδυαστικά, οδηγώντας στην υβριδική ανάκτηση που εξετάζεται στην υποενότητα 2.6.3.

2.6.2 Προσεγγιστική αναζήτηση και βάσεις διανυσμάτων

Η εξαντλητική αναζήτηση πλησιέστερου γείτονα (exact k-NN) απαιτεί τη σύγκριση του διανύσματος του ερωτήματος με κάθε μεμονωμένο διάνυσμα του σώματος. Καθώς αυξάνεται ο αριθμός των εγγράφων και η διάσταση των διανυσμάτων, η διαδικασία γίνεται ολοένα και πιο απαιτητική σε υπολογιστικούς πόρους. Για την αντιμετώπιση αυτού του προβλήματος χρησιμοποιείται η μέθοδος της προσεγγιστικής αναζήτησης πλησιέστερου γείτονα (Approximate Nearest Neighbor – ANN) περιορίζοντας την αναζήτηση σε ένα μέρος του συνολικού χώρου. Με αυτόν τον τρόπο μειώνεται αισθητά ο χρόνος αναζήτησης με το τίμημα ότι δεν υπάρχει εγγύηση πως θα επιστραφούν οι πραγματικά πλησιέστεροι γείτονες [38].

Βασικό χαρακτηριστικό των μεθόδων ANN είναι ο συμβιβασμός μεταξύ ταχύτητας και ανάκλησης. Με τη ρύθμιση των υπερπαραμέτρων της μεθόδου μπορεί να καθοριστεί το πόσο εκτενώς εξερευνάται ο χώρος. Δηλαδή, μια ευρύτερη εξερεύνηση πλησιάζει το εξαντλητικό (ακριβές) αποτέλεσμα με το ανάλογο υπολογιστικό κόστος, ενώ μια στενότερη εξερεύνηση επιταχύνει δραστικά την αναζήτηση, με τον κίνδυνο απώλειας κάποιων πραγματικών γειτόνων.

Είναι σημαντικό να σημειωθεί ότι υπάρχουν τρεις κατηγορίες μεθόδων ANN. Ο κατακερματισμός με ευαισθησία τοπικότητας (Locality-Sensitive Hashing - LSH) χρησιμοποιεί συναρτήσεις που αντιστοιχίζουν, με μεγάλη πιθανότητα, κοντινά διανύσματα στον ίδιο "κάδο". Οι μέθοδοι διαμέρισης του χώρου ομαδοποιούν τα διανύσματα και περιορίζουν την αναζήτηση σε λίγες ομάδες (π.χ. ανεστραμμένα ευρετήρια αρχειοθέτησης - IVF) [39], ενώ οι δομές δέντρων διαμερίζουν αναδρομικά τον χώρο μέσω υπερεπιπέδων. Τέλος, οι μέθοδοι που βασίζονται σε γράφους κατασκευάζουν έναν γράφο γειτνίασης, όπου κάθε διάνυσμα συνδέεται με τους κοντινότερους του, και η αναζήτηση πραγματοποιείται μέσω άπληστης διάσχισης (greedy traversal) προς τον στόχο.

Στα σύγχρονα συστήματα συναντάται πολύ η μέθοδος με γράφους και ειδικότερα η δομή του ιεραρχικά προσπελάσιμου γράφου μικρού κόσμου (Hierarchical Navigable Small World - HNSW). Η δομή αυτή οργανώνεται σε επίπεδα. Το ανώτερο επίπεδο περιέχει λίγους κόμβους και συνδέσεις που έχουν μεγάλες αποστάσεις μεταξύ τους (μακρο-πλοήγηση), ενώ τα κατώτερα περιέχουν όλους τους κόμβους με πυκνές, τοπικές συνδέσεις. Η αναζήτηση ξεκινάει από το ανώτερο επίπεδο, προσεγγίζοντας ταχύτατα την ευρύτερη περιοχή του στόχου διανύοντας μεγάλες αποστάσεις σε λίγα βήματα, και "κατεβαίνει" σταδιακά επίπεδα για τον ακριβέστερο εντοπισμό των διανυσμάτων. Παράλληλα, η αναζήτηση διαθέτει παραμέτρους που ρυθμίζουν τον συμβιβασμό ανάκλησης και ταχύτητας, ανάλογα με τον εκάστοτε στόχο του συστήματος [40].

Μία βάση δεδομένων διανυσμάτων (vector database) παρέχει λειτουργίες που επεκτείνουν τις δυνατότητες ενός απλού ευρετηρίου διανυσμάτων. Εκτός από την αποθήκευση διανυσμάτων μπορεί να διατηρεί και πρόσθετα μεταδεδομένα ή ωφέλιμο φορτίο (payload) και να εφαρμόζει φίλτρα κατά την αναζήτηση. Η λειτουργία αυτή είναι ιδιαίτερα σημαντική, καθώς δίνεται η δυνατότητα να εφαρμοστεί δυναμικό φιλτράρισμα κατά τη διαδικασία της αναζήτησης (pre-filtering) και όχι εκ των υστέρων, στα ήδη ανακτημένα αποτελέσματα (post-filtering). Η προσέγγιση αυτή διασφαλίζει ότι το σύστημα θα επιστρέψει τον επιθυμητό αριθμό k αποτελεσμάτων και δεν θα καταλήξει με λιγότερα έγγραφα (ή και κανένα), γεγονός που αποτελεί κίνδυνο στο post-filtering όταν το κριτήριο είναι αυστηρά επιλεκτικό. Επιπροσθέτως, μια βάση διανυσμάτων παρέχει πλήρη διαχείριση του κύκλου ζωής των δεδομένων (εισαγωγή, ενημέρωση, διαγραφή, αντίγραφα ασφαλείας), διεργασίες τις οποίες ένα καθαρό ευρετήριο θεωρεί εκτός της αρμοδιότητάς του [41].

Σε αυτό το σημείο πρέπει να σημειωθεί ότι υπάρχουν δύο βασικοί τρόποι ανάπτυξης. Η πρώτη εκτελείται εντός της διεργασίας της εφαρμογής ως τοπική βιβλιοθήκη, χωρίς ξεχωριστή υπηρεσία. Η άλλη επιλογή είναι μέσω εξυπηρετητή (client-server mode), εκτελείται ως αυτόνομη διεργασία, με την οποία η εφαρμογή επικοινωνεί μέσω δικτύου (π.χ. REST/gRPC). Η πρώτη περίπτωση μπορεί να απλοποιεί δραστηριότητες την εγκατάσταση, συνήθως δεν επιτρέπει ταυτόχρονη πρόσβαση από πολλαπλές διεργασίες, γεγονός που καθιστά τις υπόλοιπες λειτουργίες αμοιβαία αποκλειόμενες, για αυτό συνήθως επιλέγεται το σχήμα εξυπηρετητή.

2.6.3 Υβριδική ανάκτηση και σύντηξη

Η αραιή και η πυκνή ανάκτηση ακολουθούν διαφορετικές προσεγγίσεις, όπως διαπιστώθηκε στην υποενότητα 2.6.1, προσφέροντας διακριτά πλεονεκτήματα. Ωστόσο, η συνδυαστική τους χρήση οδηγεί στην υβριδική ανάκτηση, η οποία συνενώνει τα επιμέρους αποτελέσματά τους. Το κύριο ζήτημα, επομένως, αφορά τον τρόπο με τον οποίο συνδυάζονται τα αποτελέσματα των δύο μεθόδων.

Μία προσέγγιση είναι η σύντηξη των βαθμολογιών τους (score-level fusion), κατά την οποία οι βαθμολογίες των δύο μεθόδων συνδυάζονται, συνήθως μέσω ενός σταθμισμένου αθροίσματος. Εντούτοις, η διαδικασία αυτή είναι ιδιαίτερα σύνθετη λόγω της διαφορετικής κλίμακας κάθε βαθμολογίας και για αυτό δεν μπορούν να συγκριθούν άμεσα καθώς η πυκνή ανάκτηση υπολογίζεται συνήθως ως συνημίτονο σε ένα αυστηρά φραγμένο διάστημα (π.χ. $[-1, 1]$), ενώ η αραιή ανάκτηση (π.χ. BM25) αποτελεί άθροισμα σταθμισμένων συχνοτήτων χωρίς άνω φράγμα. Οι κατανομές τους, συνεπώς, διαφέρουν ριζικά και μεταβάλλονται δυναμικά ανάλογα με το εκάστοτε ερώτημα. Επιπλέον, σε προσπάθεια κανονικοποίησης των αποτελεσμάτων εισάγονται αυθαίρετες υποθέσεις που δεν ισχύουν ομοιόμορφα σε όλα τα δεδομένα.

Η σύντηξη σε επίπεδο κατάταξης (rank-level fusion) παρακάμπτει αυτό το πρόβλημα αγνοώντας πλήρως τις βαθμολογίες και λαμβάνοντας υπόψη μόνο τη σειρά (θέση) των εγγράφων. Η πιο καθιερωμένη μέθοδος για αυτή τη διαδικασία είναι η σύντηξη αντίστροφης κατάταξης (Reciprocal Rank Fusion - RRF) [42]:

$$RF(d) = \sum_i \frac{w_i}{K + r_i(d)} \quad (2.5)$$

όπου $r_i(d)$ είναι η θέση του εγγράφου d στην κατάταξη του σκέλους i , w_i το βάρος του σκέλους (εφόσον επιθυμούμε να δώσουμε προτεραιότητα σε ένα από τα δύο), και K μια σταθερά εξομάλυνσης.

Η μορφή αυτής της συνάρτησης διαθέτει δύο ιδιότητες με μεγάλο αντίκτυπο. Η πρώτη είναι ότι η επίδραση μειώνεται όσο χαμηλώνει η θέση του στην κατάταξη, διασφαλίζοντας ότι οι υψηλές κατατάξεις θα επηρεάζουν περισσότερο το τελικό αποτέλεσμα. Δεύτερον, η σταθερά K περιορίζει τη δυσανάλογη διαφορά μεταξύ των πρώτων θέσεων. Χωρίς αυτήν η πρώτη θέση έχει πολύ μεγαλύτερη αξία από τις υπόλοιπες γεγονός, που θα καθιστά το τελικό αποτέλεσμα υπερβολικά ευαίσθητο σε μία μόνο κατάταξη. Χάρη στον παραπάνω μηχανισμό, ένα έγγραφο που εμφανίζεται σε σταθερά καλή θέση (π.χ. στην πρώτη δάδα) σε αμφότερα τα σκέλη, τελικά υπερτερεί ενός εγγράφου που τερματίζει στην πρώτη θέση στο ένα σκέλος, αλλά αγνοείται εντελώς από το άλλο.

2.6.4 Επανακατάταξη

Αφού ανακτηθούν μερικά υποψήφια αποσπάσματα για το ερώτημα από τις μεθόδους ανάκτησης που αναφέρθηκαν μπορεί να βελτιωθεί ακόμη περισσότερο η ποιότητα των απαντήσεων και η ακρίβειά τους μέσω ενός δεύτερου σταδίου επανακατάταξης (reranking). Σε αυτό το στάδιο ένα δεύτερο μοντέλο

επανεξετάζει και αναδιατάσσει ένα περιορισμένο σύνολο υποψηφίων, λειτουργώντας ως φίλτρο για την τελική απάντηση [43].

Υπάρχουν δύο βασικές αρχιτεκτονικές κωδικοποίησης. Ο δι-κωδικοποιητής κωδικοποιεί το ερώτημα και το έγγραφο εντελώς ανεξάρτητα, παράγοντας ένα ξεχωριστό διάνυσμα για το καθένα, και εκτιμά τη συνάφεια ως συνάρτηση ομοιότητας (π.χ. συνημίτονο) των δύο αυτών διανυσμάτων. Αντίθετα, ο διασταυρούμενος κωδικοποιητής κωδικοποιεί μαζί το ερώτημα και το έγγραφο ως ενιαίο ζεύγος και παράγει απευθείας μια βαθμολογία συνάφειας [22, 43].

Η διαφορά των δύο προσεγγίσεων είναι κυρίως αρχιτεκτονική. Στην περίπτωση του bi-encoder η αναπαράσταση του εγγράφου δημιουργείται ανεξάρτητα από το ερώτημα. Έτσι, η αλληλεπίδρασή τους μπορεί να αποτυπωθεί αποκλειστικά ως γεωμετρική απόσταση στον διανυσματικό χώρο. Στην αρχιτεκτονική του cross-encoder ο μηχανισμός προσοχής επεξεργάζεται το έγγραφο και το ερώτημα από κοινού επιτρέποντας σε κάθε μονάδα (token) του ερωτήματος να συσχετίζεται και να αλληλεπιδρά απευθείας με κάθε μονάδα του εγγράφου.

Ανάλογα με την προσέγγιση που υιοθετείται, διαμορφώνεται αντιστοίχως και το υπολογιστικό κόστος. Στην περίπτωση του bi-encoder, οι αναπαραστάσεις ολόκληρου του σώματος εγγράφων μπορούν να προϋπολογιστούν κατά το στάδιο της ευρετηρίωσης και στη συνέχεια να αποθηκευτούν. Κατά τον χρόνο εκτέλεσης απαιτείται μόνο η κωδικοποίηση του ερωτήματος και η σύγκρισή του με τις διαθέσιμες αναπαραστάσεις. Στον cross-encoder όμως δεν είναι εφικτός ο προ-υπολογισμός, αφού η αναπαράσταση εξαρτάται αναπόσπαστα και από τα δύο μέλη του ζεύγους. Κάθε έγγραφο απαιτείται να αναγνωστεί σε συνδυασμό με το ερώτημα, για αυτό και η εφαρμογή του σε ολόκληρο το κείμενο από την αρχή καθίσταται απαγορευτική.

Από αυτή την ανάλυση προκύπτει η αρχιτεκτονική δύο σταδίων (two-stage retrieval pipeline). Πιο συγκεκριμένα, ένα ταχύτατο πρώτο στάδιο ανάκτησης περιορίζει το αχανές σώμα κειμένων σε ένα μικρό σύνολο υποψηφίων, και στη συνέχεια, ο διασταυρούμενος κωδικοποιητής αναδιατάσσει τα υποψήφια τμήματα. Αυτό έχει έναν σημαντικό περιορισμό που πρέπει να ειπωθεί. Ένα έγγραφο που δεν πέρασε το σύνολο των υποψηφίων εγγράφων στο πρώτο στάδιο δεν μπορεί να ανακτηθεί από το δεύτερο στάδιο. Επομένως, η ποιότητα του δεύτερου σταδίου εξαρτάται σε σημαντικό βαθμό από την ικανότητα του πρώτου να συμπεριλάβει τα πραγματικά σχετικά έγγραφα στους υποψηφίους. Από την άλλη πλευρά, η τελική κατάταξη δεν εξαρτάται απαραίτητα από την αρχική ακριβή θέση ενός εγγράφου, αρκεί αυτό να έχει συμπεριληφθεί στο σύνολο που εξετάζει ο επανακατατάκτης [44].

2.6.5 Μετρικές αξιολόγησης

Η αξιολόγηση ενός συστήματος ανάκτησης πληροφοριών αποτελεί ένα αναπόσπαστο κομμάτι της διαδικασίας ανάπτυξης καθώς απαιτεί ένα σύνολο ερωτημάτων για τα οποία είναι γνωστά τα έγγραφα που θεωρούνται συναφή. Το σύνολο αυτό αποτελεί σημείο σύγκρισης με τα αποτελέσματα που παράγει το σύστημα καθώς έτσι αξιολογείται η επίδοσή του. Ανάλογα με τη διαδικασία αξιολόγησης, η συνάφεια μπορεί να εκφράζεται με διάφορους τρόπους και τιμές, ώστε να αποτυπώνεται ο βαθμός συνάφειας μεταξύ διαφορετικών αποτελεσμάτων.

Αρχικά, οι δύο πιο βασικές μετρικές αξιολόγησης είναι η ακρίβεια (precision) και η ανάκληση (recall). Η ακρίβεια εκφράζει το ποσοστό των εγγράφων που ανακτήθηκαν και σχετίζονται με το ερώτημα. Για ένα σύνολο ανακτημένων εγγράφων R και το υποσύνολο των συναφών $Rel \cap R$, ορίζεται ως [3]:

$$\text{Precision} = \frac{|Rel \cap R|}{|R|} \quad (2.6)$$

Αποτυπώνει δηλαδή την «καθαρότητα» των αποτελεσμάτων. Υψηλή τιμή precision σημαίνει ότι μεγάλο μέρος των εγγράφων που επιστρέφονται από το σύστημα είναι σχετικά με το ερώτημα, ενώ χαμηλή τιμή υποδηλώνει την παρουσία περισσότερων άσχετων αποτελεσμάτων.

Η ανάκληση μετρά το ποσοστό των συνολικά συναφών εγγράφων που κατάφερε να ανακτήσει το σύστημα και ορίζεται ως:

$$\text{Recall} = \frac{|Rel \cap R|}{|Rel|} \quad (2.7)$$

Σε αντίθεση με την ακρίβεια, η ανάκληση αποτυπώνει πόσα σχετικά έγγραφα δεν χάθηκαν κατά τη διαδικασία αναζήτησης. Επομένως, οι δύο μετρικές αποτυπώνουν διαφορετικές οπτικές της απόδοσης. Ένα σύστημα μπορεί να εμφανίζει υψηλή ακρίβεια επιστρέφοντας λίγα αλλά κυρίως σχετικά έγγραφα, χωρίς όμως να ανακτά μεγάλο μέρος του συνόλου των συναφών εγγράφων.

Ωστόσο, σε μία εφαρμογή ανάκτησης πληροφορίας δεν αρκεί ο απλός εντοπισμός των εγγράφων που ανακτήθηκαν, καθώς τα αποτελέσματα επιστρέφονται σε συγκεκριμένη σειρά κατάταξης. Η θέση ενός συναφούς εγγράφου επηρεάζει άμεσα τη χρησιμότητα του για τον χρήστη, καθώς ένα έγγραφο που εμφανίζεται στην πρώτη θέση είναι πιθανότερο να αξιοποιηθεί από ένα αντίστοιχο έγγραφο που βρίσκεται πολύ χαμηλότερα. Αυτό δεν αποτυπώνεται από το precision και το recall καθώς παράγουν ένα συνολικό αποτέλεσμα, για αυτό απαιτούνται επιπλέον μετρικές που λαμβάνουν υπόψη τη σειρά κατάταξης.

Η ανάκληση καλύπτει αυτή την ανάγκη, δηλαδή η ανάκληση για τις πρώτες k θέσεις (Recall@k) μετρά το ποσοστό των συνολικά συναφών εγγράφων που εμφανίζονται μέσα στα k πρώτα αποτελέσματα. Η συγκεκριμένη μετρική είναι ιδιαίτερα σημαντική σε αρχιτεκτονικές με πολλαπλά στάδια καθώς αποτυπώνει την ικανότητα του πρώτου σταδίου να συμπεριλαμβάνει τα πραγματικά συναφή έγγραφα στο σύνολο των υποψηφίων που θα εξεταστεί από το επόμενο στάδιο. Ένα έγγραφο που είναι εκτός πρώτων k υποψηφίων δεν μπορεί να αξιοποιηθεί από τον επανακατατάκτη, ανεξάρτητα από την ποιότητα του δεύτερου σταδίου.

Η μέση αντίστροφη κατάταξη (Mean Reciprocal Rank – MRR) εστιάζει στη θέση του πρώτου συναφούς αποτελέσματος. Για ένα σύνολο ερωτημάτων Q, υπολογίζεται ως [45]:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q} \quad (2.8)$$

όπου rank_q είναι η θέση του πρώτου συναφούς αποτελέσματος για το ερώτημα q. Όσο υψηλότερα βρίσκεται το πρώτο συναφές έγγραφο τόσο μεγαλύτερη είναι η τιμή του MRR. Για παράδειγμα, όταν το πρώτο συναφές αποτέλεσμα βρίσκεται στην πρώτη θέση, η αντίστροφη κατάταξη ισούται με 1, ενώ όταν βρίσκεται στη δέκατη θέση ισούται με 0,1. Είναι ιδιαίτερα χρήσιμο όταν το βασικό ζητούμενο είναι ο άμεσος εντοπισμός του πρώτου συναφούς αρχείου, ωστόσο δεν λαμβάνει υπόψη τα υπόλοιπα σχετικά έγγραφα που μπορεί να εμφανίζονται χαμηλότερα στην κατάταξη.

Μία πιο αναλυτική μετρική είναι η κανονικοποιημένη προεξοφλημένη αθροιστική ωφέλεια (normalized Discounted Cumulative Gain – nDCG), η οποία λαμβάνει υπόψη τόσο τη θέση των αποτελεσμάτων όσο

και τον βαθμό συνάφειάς τους. Επιτρέπει τη διαβαθμισμένη αξιολόγηση όπως για παράδειγμα ένα έγγραφο μπορεί να χαρακτηριστεί ως ιδιαίτερα σχετικό, μερικώς σχετικό ή μη σχετικό και να λάβει αντίστοιχη βαθμολογία rel_i [46].

Αρχικά υπολογίζεται η Discounted Cumulative Gain στις πρώτες k θέσεις:

$$DCG@k = \sum_{i=1}^k \frac{rel_i}{\log_2(i+1)} \quad (2.9)$$

και στη συνέχεια η τιμή κανονικοποιείται ως προς την ιδανική δυνατή κατάταξη:

$$nDCG@k = \frac{DCG@k}{IDCG@k} \quad (2.10)$$

όπου $IDCG@k$ είναι η τιμή του $DCG@k$ που θα προέκυπτε αν τα αποτελέσματα είχαν τοποθετηθεί στην ιδανική δυνατή σειρά, με τα περισσότερα συναφή έγγραφα στις υψηλότερες θέσεις. Ο λογαριθμικός όρος στον παρονομαστή μειώνει σταδιακά τη συνεισφορά των χαμηλότερων θέσεων. Έτσι η διαφορά μεταξύ δύο θέσεων που βρίσκονται υψηλότερα έχει μεγαλύτερη επίδραση στην τελική βαθμολογία από μια αντίστοιχη διαφορά μεταξύ χαμηλότερων θέσεων. Η κανονικοποίηση ως προς την ιδανική κατάταξη επιτρέπει τη σύγκριση διαφορετικών ερωτημάτων και η τιμή του $nDCG@k$ κυμαίνεται από 0 έως 1. Με την κανονικοποίηση, το σκορ διαιρείται με την ιδανική κατάταξη ($IDCG$), με αποτέλεσμα η τελική τιμή του $nDCG@k$ να περιορίζεται στο διάστημα από 0 έως 1. Με αυτόν τον τρόπο είναι πιο εύκολη η σύγκρισή του με άλλα ερωτήματα.

Σε αυτό το σημείο πρέπει να σημειωθεί η ειδική περίπτωση που κάθε ερώτημα του συνόλου αξιολόγησης διαθέτει ακριβώς ένα συναφές έγγραφο, οι παραπάνω μετρικές παρουσιάζουν σημαντική επικάλυψη στην πληροφορία που παρέχουν. Ειδικότερα, το $Recall@k$ περιορίζεται σε ένδειξη επιτυχίας ή αποτυχίας, ανάλογα με το αν το μοναδικό συναφές έγγραφο βρίσκεται μέσα στις πρώτες k θέσεις. Αντίστοιχα, το $nDCG$ εξαρτάται αποκλειστικά από τη θέση του ίδιου εγγράφου και επομένως μεταφέρει πληροφορία παρόμοια με αυτή του MRR . Η ιδιαιτερότητα αυτή πρέπει να λαμβάνεται υπόψη κατά την ερμηνεία των αποτελεσμάτων, ώστε η χρήση πολλαπλών μετρικών να μην θεωρείται ότι παρέχει πάντοτε πλήρως ανεξάρτητες οπτικές.

Τέλος, η επιλογή των μετρικών που θα χρησιμοποιηθούν πρέπει να βασίζεται με την διαδικασία που αξιολογείται. Το $Recall@k$ εκφράζει το ανώτατο όριο που θέτει το πρώτο στάδιο ανάκτησης, καθώς τα έγγραφα που δεν ανακτώνται στα πρώιμα στάδια δεν μπορούν να εισαχθούν αργότερα από τον επανακατατάκτη. Αντίθετα, το MRR και το $nDCG$ εστιάζουν περισσότερο στην ποιότητα της τελικής κατάταξης. Επομένως, μία βελτίωση στη σειρά των αποτελεσμάτων του πρώτου σταδίου δεν είναι απαραίτητο να εμφανιστεί στην τελική επίδοση, εφόσον τα σχετικά έγγραφα βρίσκονται ήδη μέσα στο σύνολο υποψηφίων και ο επανακατατάκτης μπορεί να τα τοποθετήσει στις κατάλληλες θέσεις.

2.7 Ανάκτηση με ενισχυμένη παραγωγή

2.7.1 Το σκεπτικό και η αρχιτεκτονική

Η γνώση ενός γλωσσικού μοντέλου είναι παραμετρική (parametric), δηλαδή κωδικοποιείται στα βάρη του νευρωνικού δικτύου κατά την εκπαίδευση και δεν είναι προσβάσιμη ως διακριτό περιεχόμενο. Από αυτή την ιδιότητα απορρέουν τέσσερις κρίσιμοι περιορισμοί.

Πρώτον, μετά την εκπαίδευση του μοντέλου, η γνώση του παραμένει στατική, καθώς δεν είναι εύκολο να ανανεωθεί. Δεύτερον, δεν είναι ελέγξιμη, αφού δεν υπάρχει κάποιος τρόπος να διαπιστωθεί ότι κατέχει μια ορισμένη πληροφορία παρά μόνο εμμέσως, θέτοντάς του ερώτημα. Επιπροσθέτως, δεν υπάρχει κάποιος τρόπος να εξακριβωθεί η πηγή της πληροφορίας. Τέλος, για να ενημερωθεί η γνώση του απαιτείται επανεκπαίδευση ή προσαρμογή, μια διαδικασία με μεγάλο υπολογιστικό κόστος για αυτό καθίσταται ανέφικτη η συχνή ενημέρωσή του. Σε περιπτώσεις, όπου η πληροφορία μεταβάλλεται συχνά και η επιχειρησιακή ανάγκη επιβάλλει άμεση ενημέρωση, η αποκλειστική αξιοποίηση της παραμετρικής γνώσης κρίνεται ανεπαρκής.

Για τον σκοπό αυτό, χρησιμοποιείται η ανάκτηση με ενισχυμένη παραγωγή (Retrieval-Augmented Generation, RAG). Αντιμετωπίζει πλήρως το ζήτημα διαχωρίζοντας τη γνώση από το μοντέλο καθώς η πληροφορία παραμένει σε εξωτερικό σώμα κειμένων ως μη παραμετρική (non-parametric) γνώση και ανακτάται κατά τον χρόνο του ερωτήματος και παρέχεται στο μοντέλο μέσα στην είσοδό του. Ο ρόλος του μοντέλου αλλάζει αφού δεν καλείται πλέον να ανακαλέσει, αλλά να εξάγει και να διατυπώσει με βάση το κείμενο που έχει στη βάση του. Η αρχιτεκτονική αυτή αίρει και τους τέσσερις περιορισμούς που προαναφέρθηκαν [1].

Η δομή της αρχιτεκτονικής του RAG χωρίζεται σε δύο φάσεις με διαφορετικό χρονικό ορίζοντα. Η ευρετηρίαση εκτελείται εκτός σύνδεσης και επαναλαμβάνεται περιοδικά, ώστε το περιεχόμενο να παραμένει ενημερωμένο. Το πρώτο βήμα είναι η συλλογή των δεδομένων από διάφορες πηγές που θα ορίσει ο σχεδιαστής και με τον τρόπο που θέλει να εισάγει τις πληροφορίες. Η εξαγωγή του κειμένου είναι πολύ σημαντικό στάδιο καθώς δεν είναι πάντα εύκολο και ανάλογα τη μορφή της πηγής εξαρτάται και η μέθοδος που θα επιλεγεί για αυτόν τον σκοπό. Στη συνέχεια, το εξαγόμενο κείμενο κανονικοποιείται και τεμαχίζεται σε αποσπάσματα για να διευκολυνθεί η εύρεσή του αργότερα. Καθένα από αυτά μετατρέπεται στη συνέχεια σε διανύσματα μέσω του μοντέλου ενσωμάτωσης όπου είναι και το πιο υπολογιστικά δαπανηρό στάδιο της διαδικασίας για αυτό πραγματοποιείται εκ των προτέρων. Τελευταίο στάδιο της προεργασίας είναι η αποθήκευση των δεδομένων σε μια βάση, όπου θα ανακτηθούν αργότερα [2].

Μετά την ολοκλήρωση των παραπάνω βημάτων ακολουθεί η ανάκτηση και η παραγωγή (retrieval and generation) η οποία εκτελείται σε κάθε ερώτημα. Το ερώτημα κανονικοποιείται στην ίδια μορφή με το κείμενο ώστε να διασφαλίζεται η συνεπής επεξεργασία και σύγκρισή τους. Ανακτώνται στη συνέχεια τα αποσπάσματα που είναι σχετικά με το ερώτημα και συντίθενται κατόπιν μαζί με τις οδηγίες λειτουργίας του μοντέλου για να παραχθεί μία απάντηση από το γλωσσικό μοντέλο, με στόχο η απάντηση να βασίζεται κατά κύριο λόγο στο ανακτημένο περιεχόμενο και όχι σε πληροφορίες που δεν τεκμηριώνονται από αυτό.

Οι δύο φάσεις δεν είναι συμμετρικές καθώς αποτυχία η ανάκτησης είναι μη ανακτήσιμη: αν το ορθό απόσπασμα δεν βρίσκεται στην είσοδο, καμία ιδιότητα του μοντέλου δεν παράγει σωστή απάντηση. Επιτυχής ανάκτηση, αντιθέτως, δεν εγγυάται ορθή απάντηση, καθώς το μοντέλο ενδέχεται να επιλέξει λανθασμένο τμήμα του κειμένου ή να συγχέει πληροφορίες διαφορετικών αποσπασμάτων [47].

2.7.2 Τεμαχισμός εγγράφων

Στα συστήματα ανάκτησης τα έγγραφα δεν ευρετηριάζονται ολόκληρα, αλλά διασπώνται υποχρεωτικά σε μικρότερα αποσπάσματα (chunks). Η πρακτική αυτή επιβάλλεται από δύο κύριους τεχνικούς λόγους. Πρώτον, τα μοντέλα ενσωμάτωσης (embedding models) παράγουν ένα διάνυσμα ανά είσοδο, γεγονός που οδηγεί στη συμπίκνωση μεγάλου όγκου πληροφορίας, ο οποίος εμπεριέχει διακριτά και διαφορετικά μεταξύ τους θέματα, σε μία μόνο αναπαράσταση. Δεύτερον, το παράθυρο συμφραζομένων

(context window) του γλωσσικού μοντέλου που θα αναλάβει την παραγωγή της απάντησης είναι αυστηρά πεπερασμένο και δεν μπορεί να δεχθεί μεγάλο όγκο δεδομένων [2].

Οι στρατηγικές τεμαχισμού διαφέρουν κυρίως ως προς το κριτήριο τομής και την περίπτωση. Στη βιβλιογραφία συναντώνται διάφορες προσεγγίσεις, μερικές από αυτές είναι στρατηγικές σταθερού μήκους, αναδρομικός τεμαχισμός με ιεραρχία διαχωριστικών χαρακτήρων, δομικός τεμαχισμός, καθώς και σημασιολογικός τεμαχισμός, όπου η τομή πραγματοποιείται στα σημεία όπου η ομοιότητα των διαδοχικών προτάσεων μεταβάλλεται απότομα.

Κατά την επιλογή μιας μεθόδου πρέπει να γίνει αποδεκτός ο συμβιβασμός που αφορά το μέγεθος των αποσπασμάτων. Όταν τα τμήματα είναι υπερβολικά μικρά και εστιασμένα, εξασφαλίζεται υψηλή πιθανότητα αντιστοίχισης με το ερώτημα. Ωστόσο, το περιεχόμενό τους είναι τόσο περιορισμένο που απουσιάζουν τα ευρύτερα συμφραζόμενα, με αποτέλεσμα το γλωσσικό μοντέλο να στερείται του απαραίτητου νοηματικού πλαισίου για να συνθέσει μια ολοκληρωμένη απάντηση. Για παράδειγμα, ένα απόσπασμα που αναφέρει μια προθεσμία, χωρίς να ονομάζει τη διοικητική διαδικασία στην οποία αναφέρεται, μπορεί να αποτελεί ένα «ακριβές ταίριασμα» για το σύστημα ανάκτησης, αλλά είναι ανεπαρκές για τη σύνθεση ολοκληρωμένης απάντησης για το γλωσσικό μοντέλο. Από την άλλη πλευρά, τα μεγάλα αποσπάσματα αφενός διατηρούν το εκτενές πλαίσιο του ζητούμενου, αφετέρου περιλαμβάνουν τόσο ευρύ περιεχόμενο που η κρίσιμη πληροφορία αποδυναμώνεται, δυσχεραίνοντας το έργο του μοντέλου.

Μία ακόμη τεχνική που χρησιμοποιείται κατά τον τεμαχισμό είναι η επικάλυψη (overlap). Η τεχνική αυτή επιτρέπει στα διαδοχικά αποσπάσματα να μοιράζονται ένα κοινό τμήμα κειμένου, αποτρέποντας την αποκοπή μιας κρίσιμης πληροφορίας ακριβώς στο σημείο της τομής. Ωστόσο, η πρακτική αυτή αυξάνει το μέγεθος του ευρετηρίου και ενδέχεται να εισαγάγει πανομοιότυπα ή πλεονάζοντα αποσπάσματα στα τελικά αποτελέσματα της ανάκτησης.

Τέλος, ο τεμαχισμός δεν είναι αναγκαίο να εφαρμόζεται ομοιόμορφα σε όλο το σώμα του κειμένου. Όταν περιλαμβάνει διαφορετικού τύπου πληροφορίες με ανεξάρτητα θέματα, η εφαρμογή διαφορετικής στρατηγικής (adaptive chunking) ανά τύπο εγγράφου αποδίδει σταθερά καλύτερα από οποιαδήποτε ενιαία, καθολική ρύθμιση.

2.7.3 Θεμελίωση, παραπομπές και άρνηση

Θεμελίωση θεωρείται μια απάντηση όταν κάθε ισχυρισμός που εμπεριέχει υποστηρίζεται αποκλειστικά και αποδεδειγμένα από το ανακτηθέν κείμενο. Η θεμελίωση δεν ταυτίζεται πάντα με την ορθότητα, καθώς μία απάντηση μπορεί να είναι ορθή, χωρίς να τεκμηριώνεται από κάποια πηγή. Η διάκριση αυτή δεν είναι τυχαία. Μια μη θεμελιωμένη απάντηση μπορεί να ισχύει κατά σύμπτωση και ο μηχανισμός που την παρήγαγε δεν παρέχει εγγύηση για την επόμενη. Ο μόνος τρόπος επικύρωσης της πληροφορίας είναι με την εισαγωγή παραπομπών, ακόμη και έτσι όμως το μοντέλο μπορεί να μην βασιστεί σε αυτές τις πηγές και να επινοήσει στοιχεία (hallucinations) που δεν υπάρχουν στην είσοδό του. Συνεπώς, απαιτείται η αυστηρή επικύρωση κάθε παραπομπής έναντι των αποσπασμάτων που πράγματι ανακτήθηκαν [48].

Το κρίσιμο σημείο στην αξιολόγηση τέτοιων συστημάτων είναι ότι η άρνηση απάντησης αποτελεί επιθυμητή συμπεριφορά και όχι αποτυχία. Αυτό απορρέει από την ασυμμετρία κόστους που αναλύθηκε στην υποενότητα 2.2.1. Αν ένα σύστημα αρνείται να απαντήσει, απλώς μεταθέτει την επίλυση του ερωτήματος σε κάποια άλλη πηγή, όπως για παράδειγμα σε έναν άνθρωπο, προκαλώντας ενδεχομένως

μια μικρή χρονική καθυστέρηση. Είναι προτιμότερη η επιλογή αυτή από το να δώσει μια λάθος απάντηση με βεβαιότητα που μπορεί να οδηγήσει τον χρήστη σε λανθασμένες ενέργειες. Το ζητούμενο, επομένως, για ένα παραγωγικό σύστημα RAG, δεν είναι η μεγιστοποίηση του συνολικού ποσοστού των απαντήσεων, αλλά η αυστηρή ελαχιστοποίηση των εσφαλμένων.

Ιδιαίτερη κατηγορία που πρέπει να ληφθεί υπόψη είναι τα ερωτήματα εκτός πεδίου (out-of-scope), τα οποία αφορούν θέματα που το σώμα δεν καλύπτει. Η δυσκολία διαχείρισής τους έγκειται στο γεγονός ότι, επειδή η ζητούμενη πληροφορία δεν περιλαμβάνεται πουθενά στη βάση, ο μηχανισμός ανάκτησης επιστρέφει αναγκαστικά τα «λιγότερο άσχετα» αποτελέσματα. Αυτό επιφυλάσσει κίνδυνο καθώς το μοντέλο μπορεί να βρει πληροφορία που δεν είναι η ζητούμενη, όμως είναι αρκετή για να την εντάξει στην απάντηση [49].

2.7.4 Αποτίμηση συστημάτων ανάκτησης με ενισχυμένη παραγωγή

Η αξιολόγηση ενός συστήματος RAG δεν περιορίζεται μόνο από τις μετρικές ανάκτησης που αναλύθηκαν στην υποενότητα 2.6.5. Απαιτείται ξεχωριστή αξιολόγηση του σταδίου της παραγωγής, καθώς μπορεί να αποτύχει σε οποιοδήποτε σημείο και κάθε περίπτωση απαιτεί διαφορετική προσέγγιση για τη διόρθωσή της. Το σκέλος της παραγωγής αξιολογείται συνήθως βάσει τριών ανεξάρτητων κριτηρίων: τη θεμελίωση στα ανακτηθέντα αποσπάσματα (grounding), τη συνάφεια προς το ερώτημα (relevance) και την ορθότητα του παραγόμενου περιεχομένου (correctness). Για παράδειγμα, μια απάντηση μπορεί να είναι πλήρως θεμελιωμένη, ωστόσο να μην απαντά στο αρχικό ερώτημα [50].

Μια επιλογή αξιολόγησης είναι η αυτοματοποιημένη αξιολόγηση της παραγόμενης απάντησης καθώς αποτελεί μια ιδιαίτερα δυσχερή διαδικασία. Οι παραδοσιακές μετρικές, οι οποίες βασίζονται στην επιφανειακή λεκτική επικάλυψη (n-gram overlap) με μια απάντηση-αναφοράς, κρίνονται ανεπαρκείς, δεδομένου ότι η ίδια εννοιολογική πληροφορία μπορεί να διατυπωθεί με πολλαπλούς, ισοδύναμους τρόπους. Η σύγχρονη εναλλακτική που συναντάται είναι η χρήση ενός ισχυρού γλωσσικού μοντέλου σε ρόλο αυτόματου κριτή (LLM-as-a-judge) [51].

Ωστόσο, αυτή η τεχνική συνοδεύεται από σημαντικούς περιορισμούς. Ένας από αυτούς είναι ότι συχνά το μοντέλο-κριτής επιδεικνύει συστηματικές προκαταλήψεις (biases) ως προς τη μορφή ή την έκταση των απαντήσεων. Επιπλέον, η επιλογή του ίδιου μοντέλου που παρήγαγε την απάντηση οδηγεί στην επανάληψη των ίδιων σφαλμάτων, καθώς ο "κριτής" μοιράζεται ακριβώς τα ίδια «τυφλά σημεία» στον συλλογισμό του με τον "παραγωγό".

Για τον λόγο αυτό, η αξιολόγηση ενός συστήματος RAG δεν πρέπει να περιορίζεται μόνο ε μία μορφή αξιολόγησης, αλλά να εξετάζει και το είδος των αποτυχιών που προκύπτουν. Ιδιαίτερη σημασία έχει η διάκριση μεταξύ μιας συνειδητής άρνησης απάντησης, όταν δεν υπάρχουν επαρκείς πληροφορίες, και μιας λανθασμένης απάντησης που παρουσιάζεται ως έγκυρη. Η διάκριση αυτή είναι σημαντική, καθώς οι δύο περιπτώσεις έχουν διαφορετικό κόστος και διαφορετικές συνέπειες για την αξιοπιστία του συστήματος.

Ανεξάρτητα από τα αποτελέσματα μιας αξιολόγησης, η αξιοπιστία της εξαρτάται από τον διαχωρισμό των δεδομένων. Όταν επαναλαμβάνονται τα ίδια τεστ και οι αξιολογήσεις πάνω στα ίδια δεδομένα συνεχώς και υπάρχει μεταβολή του συστήματος για να βγάλει καλύτερα αποτελέσματα τότε εμφανίζονται τεχνητά βελτιωμένα καθώς το σύστημα προσαρμόζεται στα συγκεκριμένα δεδομένα αξιολόγησης. Για τον λόγο αυτό, χρησιμοποιείται ένα σύνολο δεδομένων που έχει δημιουργηθεί μόνο

για την ανάπτυξη το οποίο συμβουλευέται ελεύθερα ο σχεδιαστής, ενώ παράλληλα διατηρείται ένα ξεχωριστό σύνολο ελέγχου, το οποίο παραμένει σφραγισμένο μέχρι την τελική αξιολόγηση, όταν η ανάπτυξη του συστήματος έχει ολοκληρωθεί και δεν πραγματοποιούνται άλλες αλλαγές. Η αξία του δεύτερου εξαρτάται από το κατά πόσο έχει παραμείνει ανεξάρτητο από τη διαδικασία ανάπτυξης, καθώς κάθε πρόσβαση σε αυτό μπορεί να μεταφέρει πληροφορία στη διαμόρφωση του συστήματος και να μειώσει την ανεξαρτησία της τελικής μέτρησης.

2.8 Σχετικές εργασίες

2.8.1 Συστήματα απάντησης ερωτημάτων σε ακαδημαϊκό περιβάλλον

Η αυτοματοποίηση της απάντησης σε διοικητικά ερωτήματα φοιτητών αποτελούσε αντικείμενο μελέτης πριν από την ευρεία χρήση των μεγάλων γλωσσικών μοντέλων. Οι πρώτες υλοποιήσεις ήταν συστήματα διαλόγου με κανόνες, στα οποία προκαθορισμένες προθέσεις αντιστοιχίζονταν σε έτοιμα πρότυπα απαντήσεων, τα οποία συντάσσονταν από τους αρμόδιους. Τα συστήματα μπορούσαν να επιτύχουν ικανοποιητικά αποτελέσματα εντός του πεδίου για το οποίο είχαν σχεδιαστεί, όμως παρουσίαζαν σημαντική υποβάθμιση στην ποιότητα όταν τα ερωτήματα απέκλιναν από τα προβλεπόμενα σενάρια. Παράλληλα, η συνεχής επέκταση και ενημέρωση του εύρους των απαντήσεων αύξανε σημαντικά το κόστος συντήρησης, όπως αναφέρθηκε και στην υποενότητα 2.2.2 [4].

Στη συνέχεια, η εξέλιξη των συστημάτων αυτών ήταν οι ταξινομητές πρόθεσης (intent classifiers), οι οποίοι εκπαιδεύονταν σε επισημειωμένα παραδείγματα διαφορετικών διατυπώσεων. Με αυτόν τον τρόπο, το σύστημα ανταποκρινόταν καλύτερα σε διαφορετικές εκφράσεις του ίδιου αιτήματος. Εντούτοις, διατηρήθηκε το βασικό μειονέκτημα της προηγούμενης αρχιτεκτονικής: το περιεχόμενο των απαντήσεων παρέμενε στατικά προκαθορισμένο (hardcoded) και δεν αντλούνταν δυναμικά από τα επίσημα έγγραφα. Ως αποτέλεσμα, η ενημέρωση της πηγής δεν συνεπαγόταν απαραίτητα αντίστοιχη ενημέρωση του περιεχομένου των απαντήσεων, οδηγώντας σε αποκλίσεις [5].

Οι πιο σύγχρονες προσεγγίσεις αξιοποιούν την ανάκτηση με ενισχυμένη παραγωγή (Retrieval-Augmented Generation – RAG), επιτρέποντας στο σύστημα να αντλεί πληροφορίες απευθείας από ιδρυματικά έγγραφα και να τις χρησιμοποιεί για τη διαμόρφωση της τελικής απάντησης. Δύο χαρακτηριστικά των υφιστάμενων προσεγγίσεων παρουσιάζουν ιδιαίτερο ενδιαφέρον στο πλαίσιο της παρούσας εργασίας. Πρώτον, οι περισσότερες υλοποιήσεις βασίζονται σε εμπορικά γλωσσικά μοντέλα, στα οποία η πρόσβαση πραγματοποιείται μέσω ειδικών διεπαφών προγραμματισμού εφαρμογών (API). Η επιλογή αυτή έχει ως αποτέλεσμα η επεξεργασία των ιδρυματικών δεδομένων να ανατίθεται σε εξωτερικές, τρίτες υπηρεσίες. Δεύτερον, η αξιολόγηση αυτών των συστημάτων γίνεται εξετάζοντας την ποιότητα των παραγόμενων απαντήσεων, χωρίς όμως να εξετάζεται πάντοτε ξεχωριστά η συμπεριφορά τους σε ερωτήματα που δεν καλύπτονται από το διαθέσιμο σώμα κειμένων [2].

2.8.2 Ανάκτηση με ενισχυμένη παραγωγή σε θεσμικά και νομικά σώματα κειμένων

Τα θεσμικά και κανονιστικά κείμενα αποτελούν μια ιδιαίτερη κατηγορία εγγράφων με αυξημένες απαιτήσεις ως προς την ανάκτηση και την παραγωγή απαντήσεων. Η συγκεκριμένη κατηγορία μελετήθηκε σε εφαρμογές που αφορούν τη διαχείριση ακαδημαϊκών και διοικητικών πληροφοριών, με ενθαρρυντικά αποτελέσματα από τις προσπάθειες που έγιναν για συστήματα γραμματειακής υποστήριξης [52].

Βασική ιδιαιτερότητα της συγκεκριμένης περίπτωσης αφορά την αυστηρότητα στη διατύπωση των απαντήσεων. Σε κανονιστικά και νομικά κείμενα, η ακριβής διατύπωση μιας διάταξης μπορεί να

επηρεάζει το νόημα και τις συνέπειες της πληροφορίας. Για τον λόγο αυτό, είναι σημαντικό το σύστημα να περιορίζει την ελεύθερη παράφραση του περιεχομένου από το γλωσσικό μοντέλο, καθώς οι μεταβολές στη διατύπωση των κανονιστικών κειμένων μπορεί να οδηγήσουν σε διαφορετικά συμπεράσματα.

Μια ακόμη σημαντική λεπτομέρεια επίσης είναι η προέλευση των παραπομπών καθώς και η ποιότητά τους. Σε τέτοια περιβάλλοντα είναι δύσκολο να ανακτηθούν όλες οι πληροφορίες και τα έγγραφα που απαιτούνται για την απάντηση μιας διαδικαστικής ερώτησης. Είναι πιθανό να απαιτούνται περισσότερες πηγές για να απαντηθεί ένα ερώτημα καθώς μια μόνο παραπομπή δεν είναι αρκετή για να συνταχθεί μια δομημένη απάντηση. Πέρα από την ανάκτηση πληροφοριών, είναι καθοριστικής σημασίας η δυνατότητα του συστήματος να αναγνωρίζει την ανεπάρκεια του διαθέσιμου υλικού και να αποφεύγει την παραγωγή απάντησης όταν το διαθέσιμο υλικό δεν επαρκεί [48].

Από τις σχετικές προσεγγίσεις προκύπτουν δύο βασικές παρατηρήσεις. Αρχικά, ο τεμαχισμός των εγγράφων είναι σημαντικό να λαμβάνει υπόψη τη δομή και τη σημασιολογική οργάνωση του κειμένου, αντί να βασίζεται αποκλειστικά σε διαχωρισμό σταθερού μήκους. Τα φυσικά όρια του κειμένου, όπως παράγραφοι, ενότητες και άρθρα, συμβάλλουν στη διατήρηση του απαραίτητου συμφραζομένου και στη δημιουργία περισσότερο συνεκτικών αποσπασμάτων για την ανάκτηση [2].

Κεφάλαιο 3ο: Τεχνολογίες και εργαλεία υλοποίησης

3.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται οι τεχνολογίες, τα εργαλεία και οι αλγόριθμοι που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Η δομή του κεφαλαίου ακολουθεί τη ροή των δεδομένων μέσα στο σύστημα, από τη συλλογή των εγγράφων έως την παραγωγή της τελικής απάντησης, ώστε να αποσαφηνιστεί ο ρόλος κάθε επιμέρους συστατικού στη συνολική διαδικασία επεξεργασίας. Η επιλογή των τεχνολογιών βασίστηκε στις απαιτήσεις του συστήματος, με γνώμονα τη βέλτιστη ισορροπία μεταξύ ποιότητας και ταχύτητας. Στις ενότητες που ακολουθούν παρουσιάζονται τα επιμέρους στοιχεία του έργου και αιτιολογείται η επιλογή τους σε σχέση με άλλες διαθέσιμες λύσεις.

3.2 Σχεδιαστικοί περιορισμοί

3.2.1 Τοπική εκτέλεση χωρίς υπηρεσία νέφους

Το σύστημα επεξεργάζεται ερωτήματα φοιτητών και έγγραφα του τμήματος, ορισμένα από τα οποία περιέχουν προσωπικά δεδομένα ή δεν προορίζονται για δημόσια διάθεση. Η αποστολή τους σε κάποια εξωτερική υπηρεσία θα σήμαινε ότι η πληροφορία εκτίθεται σε τρίτους παράγοντες οι οποίοι δεν ελέγχονται από το πανεπιστήμιο. Για τον λόγο αυτό τέθηκε εξαρχής ως προϋπόθεση ότι κανένα τμήμα της επεξεργασίας δεν θα εκτελείται εκτός του τοπικού μηχανήματος. Η συγκεκριμένη απαίτηση μείωσε το εύρος των διαθέσιμων επιλογών, σε μεγάλα γλωσσικά μοντέλα ανοιχτού κώδικα, που δουλεύουν τοπικά και απέκλεισε υπηρεσίες ενσωμάτωσης κειμένου όπως αυτές της OpenAI ή της Cohere. Πέρα από το ζήτημα της εμπιστευτικότητας, ένα ακόμη θετικό είναι ότι η τοπική λειτουργία απαλλάσσει το σύστημα από το κόστος χρήσης και την εξάρτηση από εξωτερικό πάροχο, ο οποίος ενδέχεται να μεταβάλει τιμολόγηση, διεπαφή ή διαθεσιμότητα των μοντέλων. Για ένα σύστημα που προορίζεται να λειτουργεί σε πανεπιστημιακό τμήμα και να συντηρείται με περιορισμένους πόρους, η ιδιότητα αυτή δεν μπορεί να παραλειφθεί [6].

3.2.2 Περιορισμός διαθέσιμης μνήμης

Το σύστημα αναπτύχθηκε και εκτελείται σε πλατφόρμα Apple Silicon με 16 GB ενοποιημένης μνήμης (unified memory). Η επιλογή αυτή δεν είναι ουδέτερη: καθορίζει σχεδόν κάθε απόφαση που ακολουθεί, και για τον λόγο αυτό παρουσιάζεται πρώτη. Η ενοποιημένη μνήμη σημαίνει ότι CPU και GPU μοιράζονται τον ίδιο φυσικό χώρο. Το αποτύπωμα των μόνιμα φορτωμένων μοντέλων —του γλωσσικού μοντέλου σε 4-bit κβαντισμό (~4,3 GB), του μοντέλου ενσωματώσεων BGE-M3 (~2,2 GB) και του cross-encoder επαναταξινομητή— προσεγγίζει τα 9–10 GB. Το περιθώριο που απομένει δεν επαρκεί για δεύτερο παραγωγικό μοντέλο. Αυτός ο απλός αριθμητικός περιορισμός εξηγεί τη σειρά των αποφάσεων που διαφορετικά θα φαίνονταν αυθαίρετες: την απόρριψη της κερδοσκοπικής αποκωδικοποίησης (βλ. Υποενότητα 5.6.4), την απόρριψη του κριτή NLI ως μόνιμα φορτωμένου μοντέλου (βλ. Υποενότητα 5.6.3) και τη φόρτωση του BGE-M3 ως καθολικού singleton που μοιράζονται ο δείκτης και ο ανακτητής.

Δύο επιπλέον συνέπειες που αξίζει να σημειωθούν είναι η ετερογένεια κατανεμητών μνήμης και ο αποκλεισμός κώδικα ειδικού για CUDA. Το MLX και το PyTorch διατηρούν ανεξάρτητους κατανεμητές μνήμης GPU. Η μνήμη που δεσμεύει το MLX προσμετράται από το PyTorch ως «άλλες κατανομές» έναντι του δικού του ορίου (watermark) στη συσκευή MPS. Υλοποιήθηκε ειδικό

υποσύστημα ελέγχου (`mrs_guard`) που επιβάλλει ανώτατο όριο μνήμης, το οποίο όμως εφαρμόζεται αποκλειστικά κατά τη μαζική δεικτοδότηση και ποτέ στη ζωντανή διαδρομή απάντησης — η επιβολή ορίου εκεί προκαλεί εξάντληση μνήμης κατά τη φόρτωση του γλωσσικού μοντέλου. Λόγω του αποκλεισμού υλοποιήσεων που απαιτούν CUDA, κάθε συστατικό πρέπει να διαθέτει συμβατή διαδρομή εκτέλεσης για Apple Silicon.

3.3 Γλώσσες προγραμματισμού

Η επιλογή των γλωσσών προγραμματισμού δεν καθορίστηκε άμεσα από τους βασικούς σχεδιαστικούς περιορισμούς του συστήματος, αλλά κυρίως από τις απαιτήσεις κάθε υποσυστήματος. Το έργο αναπτύχθηκε πάνω σε τρεις βασικούς προγραμματιστικούς πυλώνες, με τον καθένα να αναλαμβάνει έναν διακριτό ρόλο. Η Python διαχειρίζεται τον πυρήνα της λογικής: εκτελεί το σύνολο της επεξεργασίας δεδομένων και υλοποιεί τη διαδικασία ανάκτησης. Από την άλλη πλευρά, η TypeScript αξιοποιείται αποκλειστικά για την κατασκευή της διεπαφής χρήστη. Τέλος, το κομμάτι της επικοινωνίας με το σύστημα αποθήκευσης υλοποιείται μέσω SQL.

3.3.1 Python

Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου που δημιουργήθηκε από τον Guido van Rossum στα τέλη της δεκαετίας του 1980. Αυτό που την κάνει ξεχωριστή από άλλες γλώσσες γενικής χρήσης είναι η χρήση της στοίχισης ως συντακτικό στοιχείο. Ο σχεδιασμός της βασίζεται στη διατήρηση ενός ευανάγνωστου και καθαρού κώδικα. Κύριο χαρακτηριστικό της είναι ότι εκτελεί τον πηγαίο κώδικά της μέσω διερμηνευτή, γεγονός που διευκολύνει την ανάπτυξη, τη δοκιμή και την αποσφαλμάτωση εφαρμογών. Χαρακτηρίζεται από υψηλή ευελιξία, καθώς παρέχει τη δυνατότητα ανάπτυξης εφαρμογών ακολουθώντας διαφορετικά προγραμματιστικά παραδείγματα, όπως τον αντικειμενοστραφή, τον διαδικαστικό και τον συναρτησιακό προγραμματισμό. Παράλληλα, το σύστημα τύπων της είναι δυναμικό. Ωστόσο, παρέχεται η δυνατότητα προσθήκης προαιρετικών υποδείξεων (`type hints`). Αυτό πρακτικά σημαίνει ότι εξωτερικά εργαλεία μπορούν να προχωρήσουν σε στατικό έλεγχο για πιθανά σφάλματα, χωρίς όμως αυτή η διαδικασία να επιβαρύνει ή να επηρεάζει τον χρόνο εκτέλεσης [53].

Η κυριαρχία της Python στα συστήματα RAG δεν αποτελεί απλώς μια προγραμματιστική τάση. Είναι το αποτέλεσμα μιας πρακτικής συσσώρευσης εργαλείων στο ίδιο οικοσύστημα. Κάθε νέα βιβλιοθήκη χρειάζεται να αναπτυχθεί στο ήδη υπάρχον περιβάλλον. Για την αρχιτεκτονική του παρόντος συστήματος, η ωριμότητα αυτού του οικοσυστήματος αποτέλεσε καθοριστικό παράγοντα, καθώς για κάθε εργαλείο που απαιτούνταν παρείχε προσαρμοσμένες και τεκμηριωμένες λύσεις. Οποιαδήποτε απόπειρα υλοποίησης του πυρήνα σε διαφορετική γλώσσα θα οδηγούσε σε συγκεκριμένα αδιέξοδα: είτε θα έπρεπε να χρησιμοποιηθούν αντίστοιχες βιβλιοθήκες αμφίβολης σταθερότητας, είτε να στηθεί μια περίπλοκη γέφυρα επικοινωνίας με εξωτερικές διεργασίες Python, γεγονός που θα πρόσθετε τεράστια πολυπλοκότητα χωρίς κανένα απολύτως τεχνικό όφελος.

Η υλοποίηση βασίστηκε στην έκδοση 3.12, η οποία καλύπτει πρακτικά όλο τον κορμό του συστήματος. Σε αυτήν αναπτύχθηκε η διαδικασία συλλογής και προεπεξεργασίας των εγγράφων, ολόκληρος ο μηχανισμός ανάκτησης, καθώς και η επικοινωνία με το γλωσσικό μοντέλο. Παράλληλα, στην ίδια γλώσσα στήθηκε το API της εφαρμογής και τα σενάρια αξιολόγησης.

Επειδή ακριβώς ο κώδικας εξυπηρετούσε τόσα πολλά διαφορετικά υποσυστήματα και μεγάλωνε συνεχώς, η συστηματική χρήση υποδείξεων τύπου (type hints) κατέστη γρήγορα αναγκαία. Εδώ η σύγχρονη σύνταξη τύπων που προσφέρει η 3.12 μας διευκόλυνε σημαντικά. Η τακτική αυτή επέτρεψε να εντοπίζουμε ασυμφωνίες δεδομένων απευθείας κατά τη συγγραφή στον editor, αποτρέποντας έτσι απρόβλεπτες αστοχίες την ώρα της εκτέλεσης.

3.3.2 TypeScript

Η TypeScript αναπτύχθηκε από τη Microsoft και αποτελεί υπερσύνολο της JavaScript, στην οποία προσθέτει σύστημα στατικών τύπων. Οποιοσδήποτε κώδικας JavaScript αποτελεί ταυτόχρονα και έγκυρο κώδικα TypeScript. Για την αντίστροφη διαδικασία, ωστόσο, απαιτείται μεταγλώττιση (compilation). Κατά τη φάση αυτή, οι πληροφορίες των στατικών τύπων αφαιρούνται πλήρως, καταλήγοντας σε καθαρή JavaScript που μπορεί να εκτελεστεί απρόσκοπτα σε κάθε σύγχρονο browser. Σε μια δυναμικά τυποποιημένη γλώσσα όπως η JavaScript, ορισμένα σφάλματα εντοπίζονται μόνο κατά την εκτέλεση. Για παράδειγμα, η πρόσβαση σε ανύπαρκτη ιδιότητα ενός αντικειμένου μπορεί να επιστρέψει undefined, με το σφάλμα να εμφανίζεται κατά τη μεταγενέστερη χρήση της τιμής [54].

Στο σύστημα, η επικοινωνία της διεπαφής με τον εξυπηρετητή δεν ακολουθεί το σχήμα μίας αίτησης και μίας απάντησης. Η ροή της πληροφορίας από το γλωσσικό μοντέλο γίνεται σταδιακά. Για να υποστηριχθεί αυτό, ο εξυπηρετητής (server) δεν περιμένει την ολοκλήρωση, αλλά στέλνει συνεχόμενα μηνύματα διαφορετικών τύπων. Τα μηνύματα αυτά αντικατοπτρίζουν την τρέχουσα κατάσταση επεξεργασίας και μεταφέρουν τα επιμέρους τμήματα του κειμένου μέχρι την τελική ολοκλήρωση της απάντησης. Η περιγραφή των μηνυμάτων αυτών ως τύπων επιτρέπει στον μεταγλωττιστή να επαληθεύσει ότι η διεπαφή χειρίζεται κάθε περίπτωση και ότι μια μεταβολή στη μορφή τους θα εντοπιστεί άμεσα. Σε προγενέστερη έκδοση της διεπαφής, γραμμένη σε απλή JavaScript, τέτοιες αστοχίες εμφανίζονταν μόνο κατά τη διάρκεια της συνομιλίας, γεγονός που αποτέλεσε τον βασικό λόγο μετάβασης στην TypeScript.

3.3.3 SQL

Η SQL είναι δηλωτική γλώσσα ερωτημάτων για σχεσιακές βάσεις δεδομένων, με καταγωγή από τις εργασίες της IBM στη δεκαετία του 1970 και τυποποιημένη έκτοτε από τον ANSI. Ο δηλωτικός της χαρακτήρας συνίσταται στο ότι το ερώτημα περιγράφει το ζητούμενο αποτέλεσμα και όχι τη διαδικασία με την οποία θα προκύψει· τον τρόπο εκτέλεσης τον αποφασίζει ο βελτιστοποιητής του συστήματος διαχείρισης, με βάση τα διαθέσιμα ευρετήρια και τη στατιστική εικόνα των δεδομένων [55].

Στο σύστημα χρησιμοποιούνται δύο διαφορετικά συστήματα διαχείρισης, τα οποία περιγράφονται στην ενότητα 3.9: η PostgreSQL για την αποθήκευση των συνομιλιών και η SQLite για την καταγραφή της λειτουργίας. Στο πρώτο, τα ερωτήματα δεν γράφονται απευθείας αλλά παράγονται από ενδιάμεσο επίπεδο αντιστοίχισης αντικειμένων, ώστε η λογική της εφαρμογής να παραμένει ανεξάρτητη από τη συγκεκριμένη υλοποίηση της βάσης. Στο δεύτερο, όπου οι απαιτήσεις περιορίζονται στη δημιουργία ενός πίνακα και στην εισαγωγή εγγραφών, τα ερωτήματα είναι γραμμένα σε SQL χωρίς ενδιάμεσο επίπεδο, καθώς η προσθήκη του δεν θα προσέφερε κάτι.

3.3.4 Bash

Η γλώσσα Bash αξιοποιείται στο σύστημα σε πολύ περιορισμένη χρήση αποκλειστικά για την υλοποίηση ενός χρονοπρογραμματιστή συλλογής δεδομένων. Η ανάγκη για επίβλεψη της διαδικασίας

συλλογής των δεδομένων, ειδικότερα τις νυχτερινές ώρες, οδήγησε στην επιλογή αυτής της γλώσσας κελύφους. Είναι υπεύθυνη για την ομαλή εκτέλεση του κώδικα, ελέγχει την ετοιμότητα της βάσης διανυσμάτων και αποτρέπει τις ταυτόχρονες εκκινήσεις. Η λειτουργία ολοκληρώνεται επιστρέφοντας έναν τυπικό κωδικό εξόδου, τον οποίο ο εγγενής χρονοπρογραμματιστής του συστήματος μπορεί να ερμηνεύσει άμεσα και με ασφάλεια [56].

3.4 Βιβλιοθήκες πρόσκτησης δεδομένων

3.4.1 Requests

Η Requests είναι βιβλιοθήκη της Python για την αποστολή αιτημάτων HTTP. Παρέχει λειτουργίες όπως διαχείριση συνόδου με διατήρηση των cookies μεταξύ διαδοχικών αιτημάτων, αυτόματη διαχείριση ανακατευθύνσεων, καθώς και εύκολη ρύθμιση επικεφαλίδων (headers) και χρονικών ορίων (timeouts). Με αυτόν τον τρόπο απλοποιείται σημαντικά η επικοινωνία με διαδικτυακές υπηρεσίες, καθώς προσφέρει έτοιμες λειτουργίες που διαφορετικά θα απαιτούσαν πιο εκτενή υλοποίηση με την πρότυπη βιβλιοθήκη της Python [57].

Στην παρούσα υλοποίηση, η βιβλιοθήκη χρησιμοποιείται από τα πέντε προγράμματα ανίχνευσης (crawlers) τα οποία συλλέγουν δεδομένα από ιστοσελίδες των οποίων το περιεχόμενο δημιουργείται στον εξυπηρετητή και αποστέλλεται έτοιμο στον περιηγητή. Η λειτουργία της βασίζεται σε σύγχρονη εκτέλεση, όπου κάθε αίτημα ολοκληρώνεται πριν αποσταλεί το επόμενο. Η επιλογή αυτή έγινε σκόπιμα καθώς ο όγκος περιορίζεται μόνο σε μερικές εκατοντάδες σελίδες. Παρότι μια ασύγχρονη προσέγγιση θα μπορούσε να μειώσει τον συνολικό χρόνο εκτέλεσης, θα καθιστούσε τον εντοπισμό σφαλμάτων πολύ πιο σύνθετο σε περίπτωση αποτυχίας μιας μεμονωμένης λήψης. Για μια διαδικασία που σχεδιάστηκε να εκτελείται χωρίς καμία επίβλεψη, η απρόσκοπτη αποσφαλμάτωση (debugging) έχει σαφώς μεγαλύτερη βαρύτητα από την απόλυτη ταχύτητα.

3.4.2 Beautiful Soup

Μια ακόμη σημαντική βιβλιοθήκη που αναλύει έγγραφα HTML και XML και τα μετατρέπει σε δομή δέντρου είναι η Beautiful Soup. Με αυτόν τον τρόπο η αναζήτηση καθίσταται πολύ πιο εύκολη καθώς γίνεται με βάση ετικέτες, κλάσεις ή αναγνωριστικά. Το βασικό χαρακτηριστικό της είναι η ανοχή της σε παραβιάσεις της σύνταξης. Οι ιστοσελίδες που χρησιμοποιούνται ως πηγές στο σύστημα προέρχονται από διαφορετικά συστήματα διαχείρισης περιεχομένου και συχνά περιλαμβάνουν ασυνέπειες στη σύνταξή τους, όπως ετικέτες που δεν κλείνουν σωστά, λανθασμένη εμφώλευση στοιχείων ή ιδιότητες χωρίς εισαγωγικά. Σε αντίθεση με έναν αυστηρό αναλυτή XML, η Beautiful Soup μπορεί να επεξεργαστεί τέτοιες σελίδες χωρίς να διακόψει την εκτέλεση [58].

Κατά τη διάρκεια συλλογής δεδομένων, το μόνο που έχει σημασία είναι το κυρίως περιεχόμενο της σελίδας. Στοιχεία όπως τα μενού πλοήγησης, οι πλευρικές στήλες και τα υποσέλιδα επαναλαμβάνονται πανομοιότυπα σε ολόκληρο τον ιστότοπο. Αν αυτά κατέληγαν στη βάση διανυσμάτων, θα δημιουργούσαν κοινό (και άρα άχρηστο) περιεχόμενο σε δεκάδες τμήματα κειμένου, υποβαθμίζοντας κατακόρυφα την ακρίβεια της ανάκτησης. Έτσι, με τη χρήση της βιβλιοθήκης απομονώνεται μόνο το κύριο τμήμα, επιλέγοντας τον κατάλληλο κόμβο του δέντρου το οποίο διαφέρει ανά πηγή και ορίζεται χωριστά σε κάθε πρόγραμμα ανίχνευσης.

3.4.3 Markdownify

Αφού απομονωθεί το ωφέλιμο τμήμα, για τη μετατροπή του καθαρού HTML σε κείμενο Markdown ενσωματώθηκε η βιβλιοθήκη markdownify. Η λειτουργία της πρακτικά αντιστοιχίζει τις ετικέτες δομής στα σύμβολα της νέας σήμανσης (για παράδειγμα, αποδίδει τις επικεφαλίδες με τον χαρακτήρα της δίσωσης και τις λίστες με παύλες), ενώ απορρίπτει εντελώς κάθε μορφοποίηση που αφορά αποκλειστικά την εμφάνιση. Στην παρούσα υλοποίηση, ο βασικός σκοπός της μετατροπής σε Markdown δεν είναι η παρουσίαση του κειμένου στον χρήστη, αλλά η διατήρηση της δομικής πληροφορίας, όπως οι επικεφαλίδες και οι λίστες, ώστε αυτή να μπορεί να αξιοποιηθεί από το επόμενο στάδιο τεμαχισμού [59].

3.4.4 Playwright

Αν και η βιβλιοθήκη Requests καλύπτει το μεγαλύτερο μέρος των διαδικτυακών αναγκών, αδυνατεί εκ σχεδιασμού να διαχειριστεί δυναμικό περιεχόμενο (Javascript) ή πολύπλοκες ροές ταυτοποίησης. Για την κάλυψη αυτής της ανάγκης ενσωματώθηκε η Playwright. Πιο συγκεκριμένα, είναι ένα εργαλείο που μπορεί να αυτοματοποιεί την εκτέλεση ενός πραγματικού περιηγητή, επιτρέποντας την προγραμματιστική αλληλεπίδραση με τα στοιχεία της οθόνης. Επειδή η εκκίνηση ενός πλήρους περιηγητή και η απόδοση της σελίδας (rendering) έχει δυσανάλογα μεγάλο υπολογιστικό κόστος, εφαρμόστηκε μια υβριδική προσέγγιση: ο αυτοματοποιημένος περιηγητής χρησιμοποιείται αποκλειστικά και μόνο για τη διαδικασία της σύνδεσης στο σύστημα Aboard. Με την ολοκλήρωσή της, τα cookies εξάγονται και μεταφέρονται σε μια σύνοδο της Request η οποία αναλαμβάνει την ταχύτερη λήψη των υπόλοιπων σελίδων. Έτσι μειώνεται το κόστος και ο χρόνος ολοκλήρωσης καθώς χρησιμοποιείται μόνο μια φορά στην αρχή της συλλογής των δεδομένων [60].

3.4.5 Certifi και η επαλήθευση των πιστοποιητικών

Κατά τη διάρκεια συλλογής των δεδομένων προέκυψε ένα ουσιαστικό πρόβλημα που αφορούσε την κρυπτογραφημένη επικοινωνία (HTTPS) με μία συγκεκριμένη πηγή. Συγκεκριμένα, ο εξυπηρετητής δεν έστελνε τα ζητούμενα ενδιάμεσα πιστοποιητικά που απαιτούνται για να κλείσει η αλυσίδα εμπιστοσύνης. Ενώ οι σύγχρονοι περιηγητές (browsers) προσπερνούν τέτοια σφάλματα αντλώντας πιστοποιητικά από την τοπική τους μνήμη (cache), το αυστηρό περιβάλλον της Python ορθά απέρριπτε τη σύνδεση. Μια προφανής λύση θα ήταν να απενεργοποιούνταν ο έλεγχος μέσω της παραμέτρου `verify=False`, πράγμα που απορρίφθηκε εξ αρχής καθώς αφήνει το σύστημα ευάλωτο σε επιθέσεις Man-in-the-Middle. Για αυτή ακριβώς την περίπτωση δόθηκε η λύση μέσω της βιβλιοθήκης `certifi`, η οποία διαχειρίζεται τα έγκυρα ριζικά πιστοποιητικά της Python. Αυτή η τοπική προσθήκη ήταν αρκετή για να αποκατασταθεί η αλυσίδα των πιστοποιητικών. Η επαλήθευση συνέχισε να λειτουργεί κανονικά στο παρασκήνιο, διατηρώντας το επίπεδο ασφάλειας ανέπαφο [61].

3.5 Εξαγωγή και επεξεργασία κειμένου

3.5.1 Docling

Σε ένα ωρολόγιο πρόγραμμα η πληροφορία δεν βρίσκεται στις λέξεις αλλά στη θέση τους. Ενδείξεις όπως “Δευτέρα” ή “αίθουσα B2” αποκτούν νόημα επειδή είναι τοποθετημένες με τέτοιο τρόπο στον πίνακα που να αντιστοιχίζεται το νόημά τους. Όταν ένας εξαγωγέας μετατρέψει τον πίνακα ως μια συνεχή ακολουθία λέξεων, οι συνδέσεις αυτές χάνονται και το κείμενο που προκύπτει δεν απαντά σε καμία ερώτηση, ακόμη κι αν περιέχει όλες τις απαραίτητες λέξεις [62].

Ακριβώς αυτό το κενό ήρθε να καλύψει η βιβλιοθήκη Docling η οποία αναπτύχθηκε από την IBM Research με σκοπό την προηγμένη εξαγωγή του περιεχομένου από αρχεία pdf. Η διαφορά με άλλους συμβατικούς εξαγωγείς είναι ότι δεν διαβάσει μόνο τη ροή περιεχομένου του αρχείου αλλά εφαρμόζει και μοντέλα ανάλυσης διάταξης στη σελίδα καθώς και ταξινομεί τα επιμέρους στοιχεία. Ένα δεύτερο μοντέλο ανακατασκευάζει τη δομή πινάκων και αντιστοιχεί κάθε κελί σε αντίστοιχη γραμμή και στήλη. Η έξοδος είναι ενιαία δομημένη αναπαράσταση ανεξάρτητα από τη μορφή εισόδου. Ίδια λογική έχουν και οι κανονισμοί, η αρίθμηση των άρθρων και η ιεραρχία των ενοτήτων τους δεν είναι διακοσμητικά στοιχεία αλλά η δομή πάνω στην οποία στηρίζεται ο τεμαχισμός που περιγράφεται στην ενότητα 3.5.6.

Η ενσωμάτωση του Docling απαιτεί δύο ρυθμίσεις που δεν προκύπτουν μέσα από την τεκμηρίωση. Η πρώτη είναι η επιλογή της υπολογιστικής μονάδας στην οποία θα εκτελείται. Το Docling αναγνωρίζει αυτόματα με βάση το υπολογιστικό σύστημα που είναι την ταχύτερη διαθέσιμη συσκευή, η οποία στη συγκεκριμένη περίπτωση είναι η MPS, η μονάδα γραφικώς της Apple Silicon. Το μοντέλο ανάλυσης διάταξης όμως χρειάζεται διπλή ακρίβεια και δεν υποστηρίζεται από αυτή τη διεπαφή. Για τον λόγο αυτό χρειάστηκε να οριστεί η βασική μονάδα επεξεργασίας που θα γίνονται οι μετατροπές. Δεύτερον, η αποτυχία επεξεργασίας δεν διακόπτει τη ροή ούτε παράγει εμφανή εξαίρεση· αντίθετα, ο αγωγός υποβαθμίζεται στην επόμενη εφεδρική βαθμίδα και συνεχίζει κανονικά. Το αποτέλεσμα είναι η σταδιακή μείωση της ποιότητας της εξαγωγής, η οποία γίνεται αντιληπτή μόνο μετά από έλεγχο του παραγόμενου περιεχομένου.

3.5.2 mlx-llm

Ωστόσο, παρά την ισχυρή επεξεργασία του Docling που αναφέρθηκε υπάρχουν τύποι εγγράφων που ακόμη και αυτό αδυνατεί να εξάγει κάποιο ουσιαστικό αποτέλεσμα. Χαρακτηριστικά στα ωρολόγια προγράμματα ορισμένων εξαμήνων, εντοπίζονται σοβαρές δομικές ασυνέπειες και ελλείψεις ιεραρχίας. Σε τέτοιες περιπτώσεις οι διαχωριστικές γραμμές δεν εξυπηρετούν κάποιο σκοπό καθώς είναι απλά γραφικά στοιχεία χωρίς σημασιολογικό περιεχόμενο, ενώ τα συγχωνευμένα κελιά καταγράφονται ως μεμονωμένα τμήματα κειμένου σε αυθαίρετες συντεταγμένες [63].

Εφόσον δεν μπορεί να ανακτηθεί με κάποιο τρόπο μέσω των ορίων του πίνακα ή των συντεταγμένων του αποφασίστηκε ότι πρέπει να αναγνωρισθεί οπτικά. Η σελίδα αποδίδεται σε εικόνα μέσω της pdf2image, η οποία στηρίζεται στο poppler για τη ραστεροποίηση, ενώ η Pillow χρησιμεύει ως ενδιάμεση αναπαράσταση της εικόνας στη μνήμη. Η εικόνα δίνεται στη συνέχεια σε πολυτροπικό γλωσσικό μοντέλο, συγκεκριμένα στο Qwen2-VL-7B σε κβαντισμό τεσσάρων bit, το οποίο φορτώνεται μέσω της βιβλιοθήκης mlx-llm και καλείται να αποδώσει το περιεχόμενο του πίνακα σε μορφή Markdown. Το μοντέλο αντιλαμβάνεται τη συγχώνευση των κελιών με τον ίδιο τρόπο που την αντιλαμβάνεται ένας άνθρωπος, δηλαδή από τη χωρική διάταξη και όχι από τα μεταδεδομένα του αρχείου.

Μια ακόμη εναλλακτική για την περίπτωση αυτή θα ήταν κάποια υπηρεσία οπτικής αναγνώρισης ή ανάλυσης εγγράφων νέφους. Πέρα από το ότι δεν χρειάζεται να γίνει ιδιαίτερη αλλαγή στον κώδικα του συστήματος είναι σαφώς πιο ώριμη και αποτελεσματική λύση. Ωστόσο λόγω των περιορισμών που τέθηκαν από την αρχή απορρίφθηκε αμέσως και το τοπικό πολυτροπικό μοντέλο αποτέλεσε επομένως τη μοναδική διαθέσιμη διεξοδό για τα έγγραφα αυτά.

Λόγω των περιορισμένων πόρων σε υπολογιστική ισχύ και του υψηλού κόστους που απαιτεί μια τέτοια διεργασία δεν γίνεται να χρησιμοποιείται για κάθε περίπτωση αρχείου pdf, παρά μόνο για αρχεία που είναι δύσκολη η εξαγωγή τους και δεν γίνεται από τη στοίχιση. Δηλαδή, η επεξεργασία απαιτεί τη φόρτωση ενός δεύτερου μοντέλου επτά δισεκατομμυρίων παραμέτρων σε μηχανήμα με δεκαέξι

γιναμπαίτ μνήμης, ενώ ο χρόνος ανά σελίδα είναι τάξεις μεγέθους μεγαλύτερος από εκείνον της ανάγνωσης κειμένου.

3.5.3 Pdfplumber και PyMuPDF

Δύο ακόμη σημαντικές βιβλιοθήκες που χρησιμοποιούνται για την εξαγωγή κειμένου από αρχεία τύπου pdf είναι η Pdfplumber και PyMuPDF. Η εκτέλεσή τους είναι άμεση και απλή καθώς δεν παρεμβάλλεται κάποιο μοντέλο μηχανικής μάθησης σε κανένα στάδιο. Λειτουργούν διαβάζοντας απευθείας τη δομή του αρχείου και ειδικότερα τους τελεστές που τοποθετούν τους χαρακτήρες σε συγκεκριμένες συντεταγμένες [64, 65].

Η pdfplumber είναι χτισμένη πάνω στην pdfminer.six υλοποίηση αναφοράς για την ανάλυση αρχείων PDF στη γλώσσα Python. Κατά την εκτέλεση, εξάγει για κάθε χαρακτήρα τη θέση, τη γραμματοσειρά και το μέγεθός του. Πάνω στην πληροφορία αυτή στηρίζεται η μέθοδος εξαγωγής πινάκων η οποία εντοπίζει δομές αναζητώντας ευθυγραμμίσεις μεταξύ των σχεδιασμένων γραμμών. Αν και λειτουργεί ικανοποιητικά σε πίνακες με ορατό πλέγμα, αποτυγχάνει σε όσους στηρίζονται αποκλειστικά σε κενά διαστήματα. Παρά τους δομικούς της περιορισμούς, η συγκεκριμένη βιβλιοθήκη επιλέχθηκε σκόπιμα ως η τελική λύση ανάγκης (fallback) στον καταρράκτη εξαγωγής (cascade). Μπορεί η παραγόμενη δομή να είναι υποδεέστερη, όμως είναι ένα εργαλείο απόλυτα αξιόπιστο. Προγραμματιστικά, κρίνεται σαφώς προτιμότερο να ευρετηριαστεί ένα κείμενο χαμηλότερης ποιότητας, παρά να χαθεί εντελώς η πληροφορία του εγγράφου.

Από την άλλη πλευρά, η PyMuPDF αποτελεί τη διεπαφή της MuPDF για την Python. Επειδή η βιβλιοθήκη είναι γραμμένη σε C, η διαφορά της ταχύτητας με άλλες αντίστοιχου τύπου βιβλιοθήκες της Python είναι τάξεων μεγέθους. Εκτός από την εξαγωγή κειμένου έχει τη δυνατότητα να εκθέτει και τη γεωμετρία της σελίδας δηλαδή διαστάσεις, πλαίσια, οριοθέτησης και θέσεις των μπλοκ κειμένου. Στο σύστημα, χρησιμεύει σε περιπτώσεις που το ζητούμενο είναι ακατέργαστο κείμενο ή συντεταγμένες, όπως κατά την αρχική φόρτωση των εγγράφων και κατά την ανάλυση των ωρολογίων προγραμμαμάτων.

Η τελική απόφαση για το ποιο θα χρησιμοποιηθεί εξαρτάται πάντα από το είδος και το περιεχόμενο του εγγράφου. Όταν απαιτείται διατήρηση δομής χρησιμοποιείται ο εξαγωγέας που τη διατηρεί, ενώ όταν αρκεί το κείμενο προτιμάται ο ταχύτερος. Η πλήρης ανάλυση διάταξης θα κόστιζε δευτερόλεπτα ανά σελίδα για πληροφορία που η MuPDF επιστρέφει σε χιλιοστά του δευτερολέπτου.

3.5.4 Python-docx

Ένα μικρό μέρος του συνόλου των δεδομένων αποτελείται από αρχεία επεξεργαστή κειμένου. Η μορφή τους μοιάζει με ένα συμπιεσμένο φάκελο που περιέχει αρχεία XML κατά το πρότυπο Office Open XML, τα οποία η python-docx αποσυμπίεζει και εκθέτει ως αντικείμενα της γλώσσας, δηλαδή ως παραγράφους, στυλ και πίνακες. Η χρήση τους όμως στο σύστημα περιορίζεται στην ανάγνωση του κειμένου κατά παράγραφο, καθώς τα έγγραφα αυτής της κατηγορίας δεν περιέχουν σύνθετες δομές [66].

3.5.5 Pandas

Αφού γίνει η εξαγωγή των πινάκων από τα σύνθετα ακαδημαϊκά έγγραφα, μετά έχει σειρά η επεξεργασία των δεδομένων και στη συνέχεια η εισαγωγή τους σε κάποια σχεσιακή βάση. Ωστόσο, αυτό δεν μπορεί να γίνει γιατί τα δεδομένα που προκύπτουν δεν είναι σε μορφή άμεσα αξιοποιήσιμη. Πιο συγκεκριμένα, η ομάδα εργαστηρίου εμφανίζεται μία μόνο φορά, ως επικεφαλίδα πάνω από το μπλοκ των γραμμών στις οποίες αναφέρεται, με αποτέλεσμα οι γραμμές αυτές να μην είναι αυτοτελείς,

οι διδάσκοντες θεωρίας και εργαστηρίου καταγράφονται σε χωριστές στήλες, εκ των οποίων η δεύτερη είναι συχνά κενή καθώς και τα ονόματα, τέλος, εμφανίζονται άλλοτε ολόκληρα με κεφαλαία και άλλοτε με πεζά [67].

Η βιβλιοθήκη Pandas αξιοποιήθηκε για την κανονικοποίηση των πινάκων η οποία εισάγει τη δομή του πλαισίου δεδομένων και επιτρέπει την εκτέλεση πράξεων σε ολόκληρες στήλες αντί για μεμονωμένες εγγραφές. Η επιλογή αυτή βελτιστοποίησε την επεξεργασία καθώς οι απαιτούμενοι μετασχηματισμοί εφαρμόζονται σε ολόκληρες στήλες και όχι σε μεμονωμένες εγγραφές. Διαφορετικά με χρήση άλλης μεθόδου όπως αυτή της Python απαιτείται να υλοποιηθεί με περισσότερο κώδικα και χειροκίνητη διαχείριση της κατάστασης μεταξύ των εγγραφών.

Η χρήση της pandas περιορίζεται αποκλειστικά στο στάδιο της προεπεξεργασίας. Αφού τα δεδομένα έρθουν σε μια ικανοποιητική μορφή μετατρέπονται σε μορφή JSON και το DataFrame αποδεσμεύεται. Κατά συνέπεια, η βιβλιοθήκη δεν χρησιμοποιείται σε κανένα στάδιο της διαδρομής ενός ερωτήματος και δεν επηρεάζει τον χρόνο απόκρισης του συστήματος.

3.5.6 Τεμαχισμός κειμένου

Το κείμενο που εξάγεται από ένα έγγραφο δεν μπορεί να χρησιμοποιηθεί ως έχει, για δύο βασικούς λόγους. Ο πρώτος αφορά την ενσωμάτωση, το μοντέλο μπορεί να παράγει ένα διάλυμα ανά είσοδο, επομένως ένα αρχείο με πολλά υποθέματα θα αναπαριστούσε τον μέσο όρο όλων των θεμάτων του και δεν θα εμφάνιζε ισχυρή ομοιότητα με κανένα συγκεκριμένο ερώτημα. Ο δεύτερος έχει να κάνει με τον περιορισμό του γλωσσικού μοντέλου. Το παράθυρο των συμφραζομένων (context window) έχει συγκεκριμένη χωρητικότητα και πρέπει να χωρέσει ταυτόχρονα το ανακτημένο κείμενο, την ερώτηση του χρήστη και τις αρχικές οδηγίες του συστήματος. Το κρίσιμο μέγεθος δεν είναι ωστόσο το μήκος των τμημάτων αλλά το σημείο στο οποίο γίνεται η τομή: ένα τμήμα που ξεκινά στη μέση μιας πρότασης και τερματίζει στη μέση της επόμενης δεν απαντά σε τίποτα, ανεξάρτητα από την έκτασή του. Το ουσιαστικό πρόβλημα, ωστόσο, δεν είναι το μήκος των τμημάτων, αλλά το σημείο στο οποίο «κόβεται» το κείμενο. Αν ένα τμήμα ξεκινάει ή τελειώνει απότομα στη μέση μιας πρότασης, χάνει εντελώς το νόημά του. Ένα τέτοιο κομμάτι είναι πρακτικά άχρηστο για το μοντέλο, ανεξάρτητα από το πόσο μεγάλο είναι [2].

Για τον τεμαχισμό χρησιμοποιούνται δύο υλοποιήσεις από τη βιβλιοθήκη LangChain. Επιπροσθέτως, ο Recursive Character Text Splitter λειτουργεί με βάση την ιεραρχία χαρακτηριστικών. Επιχειρεί πρώτα να διαίρει το κείμενο στα όρια παραγράφων και εφόσον τα τμήματα παραμένουν εκτενή, τα χωρίζει σε προτάσεις και μετά σε λέξεις, χρησιμοποιώντας πάντοτε το ισχυρότερο διαθέσιμο όριο. Επιπλέον, μεταξύ διαδοχικών τμημάτων διατηρείται μικρή επικάλυψη (overlap), ώστε πληροφορίες που βρίσκονται στα όρια δύο τμημάτων να μη χάνονται. Ο Markdown Header Text Splitter ακολουθεί διαφορετική προσέγγιση χωρίζοντας το κείμενο με βάση τις επικεφαλίδες και διατηρεί την ιεραρχία τους ως μεταδεδομένα, ώστε κάθε τμήμα να φέρει την πληροφορία της ενότητας στην οποία ανήκει.

Η στρατηγική τεμαχισμού που θα εφαρμοστεί εξαρτάται από τον τύπο του εγγράφου. Μια σύντομη ανακοίνωση παραμένει ακέραιη, καθώς οποιαδήποτε διαίρεση στο περιεχόμενό της θα κατέστρεφε την πληροφορία που μεταφέρει. Αντίθετα για έγγραφα άλλου τύπου όπως κανονισμούς και προγράμματα σπουδών χρησιμοποιείται ο Markdown Header Text Splitter και όταν κάποια ενότητα παραμένει ιδιαίτερα εκτενής εφαρμόζεται στη συνέχεια ο Recursive Character Text Splitter.

Οι δύο τεμαχιστές δεν υλοποιήθηκαν εκ νέου, καθώς η λογική των διαχωριστικών είναι δοκιμασμένη και δεν περιέχει κάποιο ζήτημα προς επίλυση. Η γενικότερη σχέση του συστήματος με τη βιβλιοθήκη LangChain, η οποία χρησιμοποιείται ως συλλογή επιμέρους εργαλείων και όχι ως πλαίσιο ανάπτυξης, εξηγείται στην ενότητα 3.6.3.

3.6 Μοντέλα και περιβάλλοντα εκτέλεσης

3.6.1 Το πλαίσιο MLX

Το MLX αποτελεί ένα εξειδικευμένο πλαίσιο υπολογισμών, αναπτυγμένο από την Apple, το οποίο σχεδιάστηκε με βασικό σκοπό την πλήρη αξιοποίηση των συστημάτων Apple Silicon. Αυτό που το κάνει ξεχωριστό είναι η ικανότητα του να εκμεταλλεύεται την ενοποιημένη μνήμη (Unified Memory) όπου η CPU και η GPU μοιράζονται τον ίδιο χώρο. Κατά συνέπεια, οι δομές των δεδομένων δεν απαιτούν ρητή αντιγραφή και μεταφορά μεταξύ διαφορετικών μονάδων, ενώ οι υπολογιστικές πράξεις μπορούν να εκτελούνται στη GPU μέσω του Metal. Παράλληλα, η αποτίμηση των εκφράσεων είναι οκνηρή, επιτρέποντας την κατασκευή και βελτιστοποίηση του γράφου υπολογισμών πριν από την τελική εκτέλεση [68].

Στο σύστημα, το MLX χρησιμοποιείται κυρίως για την αποκωδικοποίηση και την παραγωγή απαντήσεων από το μεγάλο γλωσσικό μοντέλο μέσω της βιβλιοθήκης `mlx-lm`. Η επιλογή του MLX προηγήθηκε της επιλογής του ίδιου του μοντέλου, καθώς υπαγόρευσε εξ αρχής τη μορφή και την κβάντιση (quantization) των διαθέσιμων βάρων.

Πίνακας 3.1: Πλαίσια εκτέλεσης που εξετάστηκαν και λόγοι επιλογής

Πλαίσιο	Έκβαση	Λόγος
MLX	Επιλέχθηκε	Ενοποιημένη μνήμη, κρυφή μνήμη KV προσβάσιμη από τον κώδικα
Llama.cpp / Ollama	Απορρίφθηκε	Η κατάσταση της αποκωδικοποίησης βρίσκεται πίσω από όριο διεργασίας
Pytorch με Backend MPS	Απορρίφθηκε για το LLM	Υψηλότερη κατανάλωση μνήμης, ελλιπής υποστήριξη κβαντισμένων πράξεων
CUDA	Μη εφαρμόσιμο	Απαιτεί υλικό NVIDIA (Π2)

Οι λόγοι που έκριναν την απόφαση ήταν δύο. Ο πρώτος λόγος αφορά τον περιορισμό της διαθέσιμης μνήμης, καθώς το σύστημα εκτελείται σε μηχανήμα με 16 GB RAM. Πέρα από το γλωσσικό μοντέλο έπρεπε να συνυπάρχουν και άλλα μέρη της εφαρμογής που θα απαιτούσαν επιπλέον χώρο στη μνήμη και για αυτό χρειάστηκε να αξιοποιηθεί μια πιο προσιτή λύση για τη διαχείριση του χώρου μνήμης και του υλικού.

Ένα ακόμη πλεονέκτημα που προτρέπει στην χρήση του `mlx-lm` είναι ότι παρέχει πρόσβαση στην KV Cache (Key-Value Cache) του γλωσσικού μοντέλου. Αυτό μπορεί να βελτιστοποιήσει τη λειτουργία του μοντέλου καθώς επιτρέπει την επαναχρησιμοποίηση του κοινού τμήματος της προτροπής (system prompt) μεταξύ διαδοχικών ερωτημάτων, αποφεύγοντας τον επανυπολογισμό του ίδιου περιεχομένου σε κάθε αίτημα. Με τον τρόπο αυτό μειώνεται ο χρόνος παραγωγής των απαντήσεων χωρίς να επηρεάζεται η ποιότητα των αποτελεσμάτων. Κατά τη διαδικασία σχεδιασμού εξετάστηκαν και άλλες εναλλακτικές, όπως το `Llama.cpp` μέσω του `Ollama` και το `PyTorch` με υποστήριξη `Metal` (MPS). Παρόλο που οι λύσεις αυτές μπορούν επίσης να εκτελέσουν μεγάλα γλωσσικά μοντέλα δεν παρείχαν τον ίδιο βαθμό ελέγχου πάνω στην επαναχρησιμοποίηση της KV Cache ή εμφάνιζαν μεγαλύτερες

απαιτήσεις σε μνήμη για το συγκεκριμένο υλικό. Για τον λόγο αυτό το MLX αποτέλεσε την καταλληλότερη επιλογή για την υλοποίηση του συστήματος.

3.6.2 Το γλωσσικό μοντέλο Qwen3-8B

Το Qwen3-8B είναι ένα μεγάλο γλωσσικό μοντέλο που σχεδιάστηκε και αναπτύχθηκε από την Alibaba Cloud και ανήκει στην οικογένεια Qwen3. Πρόκειται για ένα πυκνό μοντέλο 8 δισεκατομμυρίων παραμέτρων, εκπαιδευμένο σε πολύγλωσσο σώμα και προσαρμοσμένο στην τήρηση οδηγιών (instruction tuning). Στο σύστημα χρησιμοποιείται η έκδοση με κβαντισμό 4 bit, όπως διατίθεται για το MLX, και εκτελείται τοπικά χωρίς καμία επικοινωνία με εξωτερική υπηρεσία αφού διατίθεται με ανοικτά βάρη [69].

Ο ρόλος του μοντέλου στο σύστημα είναι περιορισμένος και χρησιμοποιείται αποκλειστικά στο στάδιο παραγωγής της απάντησης. Ως είσοδο λαμβάνει την ερώτηση του χρήστη καθώς και τα αποσπάσματα που επέλεξε το υποσύστημα ανάκτησης. Με βάση τις οδηγίες του συστήματος που του έχουν δοθεί καθοδηγείται ώστε να παράγει μια τελική απάντηση, και να δηλώνει αδυναμία απάντησης όταν οι διαθέσιμες πληροφορίες δεν επαρκούν. Αντίθετα, λειτουργίες όπως η αναγνώριση της πρόθεσης του χρήστη, η ανάκτηση των σχετικών εγγράφων και η επανακατάταξη των αποτελεσμάτων υλοποιούνται από ανεξάρτητα υποσυστήματα, ώστε η μη ντετερμινιστική συμπεριφορά να περιορίζεται αποκλειστικά στο τελικό στάδιο παραγωγής.

Η επιλογή του συγκεκριμένου μοντέλου έγινε έπειτα από αρκετές δοκιμές και καθορισμό των απαιτήσεων του συστήματος. Αρχικά εξετάστηκε το Llama-Krikri-8b-Instruct το οποίο είναι ένα ελληνόγλωσσο μοντέλο που αναπτύχθηκε από το Ινστιτούτο Επεξεργασίας του Λόγου και παρουσιάζει ιδιαίτερα καλές επιδόσεις σε σημεία αναφοράς της ελληνικής γλώσσας. Σε συστήματα RAG δεν έχει μόνο σημασία η ποιότητα της γλώσσας αλλά και η ικανότητά του μοντέλου να ακολουθεί πιστά τις οδηγίες της προτροπής. Εφόσον οι πληροφορίες ανακτώνται από την βάση και παρέχονται στο μοντέλο μέσω του context window, το ζητούμενο είναι η παραγωγή απαντήσεων που βασίζονται αποκλειστικά στο παρεχόμενο υλικό, χωρίς την προσθήκη γνώσεων που δεν τεκμηριώνονται από τα ανακτημένα αποσπάσματα. Στη συγκεκριμένη περίπτωση το Qwen3 διαπιστώθηκε ότι έχει πιο σταθερή συμπεριφορά και μικρότερη τάση παραγωγής μη τεκμηριωμένων ισχυρισμών, γεγονός που οδήγησε στην αντικατάσταση του προηγούμενου μοντέλου.

Στο σύστημα χρησιμοποιείται η κβαντισμένη έκδοση των 4 bit μέσω της βιβλιοθήκης MLX-LM. Ο κβαντισμός αποτελεί τεχνική συμπίεσης των βαρών του μοντέλου, κατά την οποία κάθε παράμετρος αποθηκεύεται με μικρότερη αριθμητική ακρίβεια, μειώνοντας σημαντικά τις απαιτήσεις σε μνήμη και επιτρέποντας την εκτέλεση μεγάλων μοντέλων σε υπολογιστικά συστήματα με περιορισμένους πόρους. Παράλληλα δοκιμάστηκε και η έκδοση των 6 bit με την προσδοκία ότι θα βελτίωνε την ποιότητα των παραγόμενων απαντήσεων εις βάρος ενός μικρού μέρους της απόδοσης. Ωστόσο, τα αποτελέσματα έδειξαν το αντίθετο. Ο χρόνος απόκρισης αυξήθηκε αισθητά, καθώς η έκδοση αυτή απαιτούσε μεγαλύτερη υπολογιστική ισχύ και μνήμη με αποτέλεσμα την καθυστέρηση της απάντησης. Για τον λόγο αυτό διατηρήθηκε η έκδοση των 4 bit, η οποία παρουσίασε καλύτερη συνολική συμπεριφορά τόσο ως προς την απόδοση όσο και ως προς την αξιοπιστία των αποτελεσμάτων.

3.6.3 PyTorch και NumPy

Η PyTorch χρησιμοποιείται κυρίως ως υποδομή άλλων βιβλιοθηκών, ενώ η NumPy αξιοποιείται τόσο έμμεσα από το οικοσύστημα όσο και άμεσα σε επιμέρους στάδια επεξεργασίας διανυσμάτων. Η

PyTorch είναι ένα από τα πιο σημαντικά πλαίσια ανάπτυξης εφαρμογών καθώς παρέχει μηχανισμούς για την αποδοτική εκτέλεση νευρικών δικτύων, υποστηρίζοντας επιτάχυνση μέσω του Metal Performance Shaders (MPS) σε συστήματα Apple Silicon. Αντίστοιχα η NumPy παρέχει την δυνατότητα επεξεργασίας και διαχείρισης αριθμητικών πινάκων οι οποίοι είναι απαραίτητοι για μεγάλο μέρος του οικοσυστήματος της Python [70, 71].

Η PyTorch δεν είναι άμεσα χρησιμοποιήσιμη από το σύστημα για την εκτέλεση του μεγάλου γλωσσικού μοντέλου, όμως αξιοποιείται έμμεσα από τις βιβλιοθήκες Sentence-Transformers και Transformers για την εκτέλεση του μοντέλου ενσωματώσεων και του επανακατατάκτη. Η NumPy χρησιμοποιείται σε επιμέρους στάδια προεπεξεργασίας των διανυσμάτων, όπως η κανονικοποίηση και οι μετατροπές μορφής πριν από την αποθήκευσή τους στη βάση διανυσμάτων.

Κατά τη δημιουργία των ενσωματώσεων επεξεργάζεται μεγάλος αριθμός εγγράφων, γεγονός που αυξάνει σημαντικά τις απαιτήσεις σε μνήμη. Για την αποφυγή εξάντλησης της διαθέσιμης μνήμης εφαρμόζεται όριο χρήσης 70% στον backend MPS της PyTorch. Η ρύθμιση αυτή χρησιμοποιείται μόνο στη διαδικασία δημιουργίας των ενσωματώσεων και όχι κατά την εξυπηρέτηση ερωτημάτων, όπου ο ενσωματωτής επεξεργάζεται μία μόνο πρόταση κάθε φορά και η μεγαλύτερη ποσότητα μνήμης πρέπει να παραμένει διαθέσιμη για το γλωσσικό μοντέλο.

3.6.4 Η βιβλιοθήκη sentence-transformers και το μοντέλο ενσωμάτωσης BGE-M3

Η σημασιολογική αναπαράσταση των κειμένων αποτελεί βασικό στάδιο της διαδικασίας ανάκτησης, καθώς επιτρέπει τη σύγκριση της σημασίας δύο κειμένων αντί της απλής αντιστοίχισης λέξεων. Αυτή τη διαδικασία την αναλαμβάνει η βιβλιοθήκη Sentence-Transformers, η οποία παρέχει έναν απλό τρόπο φόρτωσης μοντέλων ενσωμάτωσης και δημιουργίας embeddings, αναλαμβάνοντας εσωτερικά όλες τις απαραίτητες διαδικασίες, όπως η ομαδοποίηση των δεδομένων εισόδου, η εξαγωγή της τελικής αναπαράστασης και η κανονικοποίηση των παραγόμενων διανυσμάτων. Πιο αναλυτικά, μετατρέπει ένα τμήμα κειμένου σε ένα αριθμητικό διάνυσμα σταθερών διαστάσεων, έτσι ώστε κείμενα με παρόμοιο νόημα να αντιστοιχίζονται σε διανύσματα που βρίσκονται κοντά μεταξύ τους στον διανυσματικό χώρο [22, 23].

Για τη σημασιολογική αναπαράσταση κειμένου χρησιμοποιείται το μοντέλο BAAI/bge-m3, ένα πολύγλωσσο μοντέλο ενσωμάτωσης με διανύσματα 1024 διαστάσεων και μέγιστο μήκος εισόδου 8192 διακριτικών. Το μοντέλο αυτό αξιοποιείται σε όλα τα στάδια όπου απαιτείται σημασιολογική αναπαράσταση κειμένου: κατά την ευρετηρίαση των εγγράφων για τη δημιουργία των embeddings, κατά τη μετατροπή του ερωτήματος του χρήστη σε διάνυσμα, στη σύγκριση με τα πρότυπα διανύσματα του δρομολογητή πρόθεσης και στον έλεγχο ομοιότητας της προσωρινής μνήμης απαντήσεων.

Η επιλογή του BGE-M3 βασίστηκε κυρίως σε δύο χαρακτηριστικά. Το πρώτο είναι η ουσιαστική υποστήριξη πολλών γλωσσών, συμπεριλαμβανομένων των ελληνικών, γεγονός ιδιαίτερα σημαντικό για το εξεταζόμενο σύστημα. Το δεύτερο αφορά τον τρόπο χρήσης του μοντέλου. Πολλά μοντέλα απαιτούν διαφορετική μορφή εισόδου για τα ερωτήματα και για τα αποσπάσματα κειμένου, χρησιμοποιώντας ειδικά προθέματα, όπως query: και passage. Αντίθετα, το BGE-M3 δεν απαιτεί τέτοιο διαχωρισμό, επιτρέποντας τη χρήση του ίδιου μοντέλου τόσο κατά την ευρετηρίαση όσο και κατά την αναζήτηση. Με τον τρόπο αυτό απλοποιείται η αρχιτεκτονική του συστήματος και αποφεύγονται σφάλματα που θα μπορούσαν να προκύψουν από διαφορετική επεξεργασία των ίδιων δεδομένων στα δύο στάδια.

Συμπληρωματικά, χρησιμοποιείται το μοντέλο BAAI/bge-reranker-v2-m3, το οποίο έχει ως βασικό ρόλο την επανακατάταξη των αποτελεσμάτων που προκύπτουν από το στάδιο της ανάκτησης. Σε αντίθεση με το μοντέλο ενσωμάτωσης, δεν παράγει διανύσματα ούτε υπολογίζει ομοιότητα μεταξύ

τους. Αντίθετα, δέχεται την ερώτηση και το υπονήφιο απόσπασμα ως μία ενιαία ακολουθία εισόδου και υπολογίζει απευθείας μία βαθμολογία συνάφειας για το συγκεκριμένο ζεύγος. Με τον τρόπο αυτό μπορεί να εκτιμήσει αν ένα απόσπασμα απαντά πραγματικά στην ερώτηση και όχι απλώς αν πραγματεύεται παρόμοιο θέμα.

Για την αξιοποίηση του μοντέλου επιλέχθηκε η βιβλιοθήκη Sentence-Transformers αντί της FlagEmbedding, η οποία προτείνεται από τους δημιουργούς του BGE-M3. Η επιλογή δεν έχει μεγάλη σημασία καθώς και οι βιβλιοθήκες βγάζουν τα ίδια αποτελέσματα και χρησιμοποιούν τα ίδια βάρη του μοντέλου. Παράλληλα, η Sentence-Transformers παρουσίασε καλύτερη συμβατότητα με τις υπόλοιπες βιβλιοθήκες του έργου, καθώς απέφυγε συγκρούσεις εκδόσεων με το MLX-LM και το οικοσύστημα Transformers, διευκολύνοντας τη συνύπαρξη όλων των επιμέρους υποσυστημάτων στην ίδια εφαρμογή.

3.7 Ανάκτηση και αποθήκευση διανυσμάτων

3.7.1 Το σύστημα Qdrant

Η διαχείριση των διανυσματικών αναπαραστάσεων στο σύστημα υλοποιείται μέσω του Qdrant, ενός συστήματος βάσης δεδομένων ανοικτού κώδικα, γραμμένου στη γλώσσα Rust. Οργανώνει τα διανύσματα σε συλλογές και για καθεμία ορίζει διάσταση διανύσματος και μετρική απόστασης και μέσω του αλγορίθμου HNSW ευρετηριάζει τα διανύσματα για καλύτερη οργάνωση. Κάθε εγγραφή εκτός από το διάνυσμα συνοδεύεται από ωφέλιμο φορτίο (payload) σε μορφή JSON, το οποίο χρησιμεύει στην αναζήτηση των εγγράφων καθώς τα φίλτρα εφαρμόζονται κατά τη διάρκεια της διάσχισης του γράφου και όχι μετά την ολοκλήρωσή της. Η επικοινωνία με την υπηρεσία γίνεται μέσω REST ή gRPC, με την υποστήριξη επίσημων βιβλιοθηκών σε Python [40, 41].

Παράλληλα, το Qdrant μπορεί να υποστηρίξει δύο τρόπους λειτουργίας. Ο ένας είναι η ενσωματωμένη τοπική λειτουργία που λειτουργεί εσωτερικά στη διεργασία της εφαρμογής αποθηκεύοντας τα δεδομένα σε τοπικό φάκελο, γεγονός ιδανικό για δοκιμές και για το στάδιο της υλοποίησης. Ο δεύτερος τρόπος είναι η λειτουργία εξυπηρετητή (server mode), κατά την οποία λειτουργεί ως αυτόνομη διεργασία έτοιμη για περιβάλλον παραγωγής. Για το έργο αυτό αξιοποιήθηκαν και οι δύο τρόποι, η ενσωματωμένη λειτουργία για το στάδιο της ανάπτυξης και η λειτουργία εξυπηρετητή αναλαμβάνει το τελικό περιβάλλον, εξασφαλίζοντας ότι ο μηχανισμός ευρετηρίασης και η διαδικτυακή εφαρμογή μπορούν να προσπελάσουν ταυτόχρονα τα δεδομένα.

Κάθε εγγραφή στη συλλογή αποτελείται από διάνυσμα 1024 διαστάσεων, το ίδιο το κείμενο του τμήματος (chunk) καθώς και ένα σύνολο μεταδεδομένων όπως είναι ο τύπος του εγγράφου, η πηγή προέλευσης, το ακαδημαϊκό έτος. Τα δεδομένα είναι χρήσιμα κατά τη διαδικασία την αναζήτησης καθώς μπορούν να εφαρμόζονται τα αντίστοιχα φίλτρα και επιστρέφονται μόνο τα έγγραφα που πληρούν τα επιθυμητά κριτήρια.

Η επιλογή του Qdrant έναντι άλλων αντίστοιχων βάσεων βασίστηκε στη δυνατότητα ολοκληρωμένης υποστήριξης μεταδεδομένων και στην πλήρως τοπική λειτουργία του. Άλλη εναλλακτική λύση όπως το Pinecone, απορρίφθηκε γιατί απαιτούσε τη χρήση υπηρεσιών νέφους για την αποθήκευση των δεδομένων, γεγονός που δεν επιτρέπεται στην αρχιτεκτονική αυτή. Ακόμη, το FAIS είναι και αυτό αποδοτικό στην αναζήτηση διανυσμάτων, δεν παρέχει εγγενή αποθήκευση και φιλτράρισμα

μεταδεδομένων, απαιτώντας την ανάπτυξη πρόσθετου μηχανισμού για τη διαχείρισή τους. Αντίστοιχα, η χρήση του pgvector θα συνέδεε τη λειτουργία της διανυσματικής αναζήτησης με τη σχεσιακή βάση δεδομένων της εφαρμογής, αυξάνοντας την αλληλεξάρτηση δύο ανεξάρτητων υποσυστημάτων. Για τους λόγους αυτούς, το Qdrant αποτέλεσε την καταλληλότερη επιλογή για τις απαιτήσεις του εξεταζόμενου συστήματος.

3.7.2 LangChain

Το LangChain αποτελεί μία από τις πλέον διαδεδομένες βιβλιοθήκες ανάπτυξης εφαρμογών που βασίζονται σε μεγάλα γλωσσικά μοντέλα (Large Language Models – LLMs). Παρέχει έτοιμες υλοποιήσεις για τις απαιτήσεις τέτοιων συστημάτων όπως για τη φόρτωση και επεξεργασία εγγράφων, τον τεμαχισμό κειμένου, την ανάκτηση πληροφορίας, τη διαχείριση προτροπών και τη σύνθεση πολύπλοκων ροών εκτέλεσης. Στόχος της είναι να απλοποιεί την ανάπτυξη εφαρμογών ανάκτησης με ενισχυμένη παραγωγή, προσφέροντας έτοιμες υλοποιήσεις για λειτουργίες που εμφανίζονται συχνά σε τέτοιου είδους συστήματα [72].

Στο συγκεκριμένο σύστημα η χρήση του LangChain είναι περιορισμένη και αξιοποιούνται μόνο επιμέρους στοιχεία της βιβλιοθήκης. Πιο συγκεκριμένα, χρησιμοποιούνται δύο τεμαχιστές κειμένου για το μέρος του Chunking: ο RecursiveCharacterTextSplitter και ο MarkdownHeaderTextSplitter καθώς και ο λεξικός ανακτητής BM25Retriever, ο οποίος χρησιμοποιείται στο λεξικό σκέλος της υβριδικής ανάκτησης. Ο ανακτητής επιτρέπει την προσαρμογή του τρόπου τεμαχισμού των λέξεων. Η ρύθμιση αυτή αποδείχθηκε απολύτως αναγκαία για τη διαχείριση της ελληνικής γλώσσας και αναλύεται στην επόμενη υποενότητα.

Παρά τις διευκολύνσεις και τις έτοιμες λειτουργίες που προσφέρει η βιβλιοθήκη, κρίθηκε αναγκαίο να μην χρησιμοποιηθούν οι περισσότερες από αυτές και αντί αυτού να συνταχθεί ο απαιτούμενος κώδικας. Αυτό είχε ως σκοπό κάθε στάδιο της ανάκτησης και της παραγωγής απαντήσεων να μπορεί να ελεγχθεί και να προσαρμοστεί στις απαιτήσεις της εφαρμογής.

Χαρακτηριστικό παράδειγμα αποτελεί η υλοποίηση της σύντηξης των αποτελεσμάτων με τη μέθοδο Reciprocal Rank Fusion (RRF). Παρότι το LangChain παρέχει έτοιμη υλοποίηση, προτιμήθηκε να αναπτυχθεί εκ νέου, καθώς η υπάρχουσα δεν εκθέτει σημεία ελέγχου που ήταν απαραίτητα στην εφαρμογή, όπως ο τρόπος αναγνώρισης ίδιων αποτελεσμάτων, η διαχείριση ισοβαθμιών και η δυνατότητα εισαγωγής αποτελεσμάτων από προηγούμενα στάδια του αγωγού. Οι επιλογές αυτές έχουν σημαντικό αντίκτυπο καθώς επηρεάζουν τα αποσπάσματα που προωθούνται στον επανακατατάκτη και τελικά στο γλωσσικό μοντέλο.

Με τον τρόπο αυτό η βιβλιοθήκη συμμετέχει στην αρχιτεκτονική του συστήματος αποκλειστικά εκεί που απαιτείται ως συλλογή ανεξάρτητων εργαλείων. Η επιλογή συμβαδίζει με τον περιορισμό Π5, σύμφωνα με τον οποίο έτοιμες υλοποιήσεις αξιοποιούνται μόνο όταν δεν περιορίζουν τον έλεγχο του αλγορίθμου ή την προσαρμογή του στις ανάγκες του συστήματος.

3.7.3 Η βιβλιοθήκη snowballstemmer και η ελληνική μορφολογία στο αραιό σκέλος

Κατά τη διάρκεια της υλοποίησης του συστήματος παρατηρήθηκε μια σχεδιαστική αστοχία που είχε να κάνει με την ανάκτηση της πληροφορίας. Το στάδιο της ανάκτησης το αναλαμβάνει ο BM25, ο οποίος δίνει μια βαθμολογία με βάση τη συνεμφάνιση ακριβώς των ίδιων λεκτικών μονάδων σε ερώτημα και έγγραφο. Σε περιπτώσεις όπου μια λέξη έχει πολλές κλίσεις όπως για παράδειγμα «Άρθρο», «Άρθρου», «Άρθρα» και «Άρθρων» αντιμετωπίζονται ως τέσσερις ανεξάρτητοι όροι. Αυτό δημιουργεί το εξής πρόβλημα: αν ένας φοιτητής κάνει ένα ερώτημα που περιέχει τη λέξη «άρθρο» δεν παρουσιάζει καμία λεξική επικάλυψη με διάταξη κανονισμού που γράφει «Άρθρου 12» και ο ανακτητής επιστρέφει μηδενική βαθμολογία για ένα απόσπασμα που είναι στην πραγματικότητα το ζητούμενο. Το φαινόμενο είναι συστηματικό στα ελληνικά, όπου η κλίση των ουσιαστικών, των επιθέτων και των ρημάτων παράγει πλήθος μορφών ανά λήμμα, και επιβαρύνει δυσανάλογα ακριβώς το είδος του περιεχομένου που κυριαρχεί στο σώμα: τη διοικητική και νομική διατύπωση [13, 73].

Η Snowball είναι γλώσσα περιγραφής αλγορίθμων στελέχωσης (stemming) που ανέπτυξε ο Martin Porter ως γενίκευση του ομώνυμου αλγορίθμου του, μαζί με ένα σύνολο υλοποιήσεων για περισσότερες από είκοσι γλώσσες. Η snowballstemmer είναι η αντίστοιχη υλοποίηση για Python. Ο στελεχωτής αφαιρεί καταλήξεις με βάση ρητούς μορφολογικούς κανόνες και όχι με λεξικό ή στατιστικό μοντέλο· δεν εκτελεί λημματοποίηση και δεν εγγυάται ότι το αποτέλεσμα είναι υπαρκτή λέξη, αλλά εγγυάται ότι διαφορετικές κλιτικές μορφές του ίδιου λήμματος καταλήγουν στην ίδια συμβολοσειρά. Για τον σκοπό της λεξικής ανάκτησης η ιδιότητα αυτή είναι επαρκής: δεν ενδιαφέρει η γλωσσολογική ορθότητα του στελέχους, αλλά η σύμπτωσή του μεταξύ ευρετηρίου και ερωτήματος.

Η βιβλιοθήκη SnowballStemmer αξιοποιείται μέσω του ελληνικού στελεχωτή (Greek Stemmer) για την κανονικοποίηση των λέξεων πριν από το στάδιο της λεξικής αναζήτησης. Στο πλαίσιο αυτό το κείμενο υφίσταται μια βασική προεπεξεργασία πριν εφαρμοστεί η διαδικασία στελέχωσης, όπως η ενοποίηση της αναπαράστασης των χαρακτήρων, η μετατροπή σε πεζά και ο τεμαχισμός σε λεκτικές μονάδες. Βασικός σκοπός της διαδικασίας είναι διαφορετικές κλιτικές μορφές της ίδιας λέξης να αντιστοιχίζονται σε κοινό στέλεχος, αυξάνοντας τη λεξική αντιστοίχιση μεταξύ ερωτήματος και εγγράφων.

Η κανονικοποίηση αυτή προβλέπεται να εφαρμόζεται στον λεξικό ανακτητή BM25 σε δύο σημεία, κατά την ευρετηρίαση των εγγράφων και κατά την επεξεργασία των ερωτημάτων, ώστε να διασφαλίζεται η συνέπεια μεταξύ των δύο σταδίων. Η λειτουργία αυτή είναι υλοποιημένη αλλά παραμένει ανενεργή και οι μετρήσεις πραγματοποιήθηκαν από τον προεπιλεγμένο διαχωρισμό λέξεων της βιβλιοθήκης. Ο λόγος αναφέρεται στην Ενότητα 5.6, καθώς η λειτουργία αυτή δεν απέδωσε ουσιαστικό όφελος στην τελική κατάταξη διότι η ανάκληση βρισκόταν ήδη στο ανώτατο όριο και ο επανακατατάκτης απορρόφησε τη βελτίωση του λεξικού σκέλους. Ανεξαρτήτως της κατάστασης της σημαίας, η στελέχωση αφορά αποκλειστικά το λεξικό σκέλος της υβριδικής ανάκτησης και όχι το σημασιολογικό μοντέλο, το οποίο αναγνωρίζει τις μορφολογικές παραλλαγές χωρίς να απαιτείται αντίστοιχη γλωσσική επεξεργασία.

Τέλος, επειδή τα έγγραφα του συστήματος είναι διοικητικά και κανονιστικά κείμενα, περιλαμβάνουν λειτουργικές λέξεις με σημασιολογική αξία. Για τον λόγο αυτό, η χρήση λίστας λέξεων αποκλεισμού (stop words) διατηρήθηκε απενεργοποιημένη.

3.8 Τεχνολογίες διαδικτυακής εφαρμογής

3.8.1 FastAPI και Unicorn

Το FastAPI είναι σύγχρονο πλαίσιο ανάπτυξης διαδικτυακών υπηρεσιών για την Python χτισμένο πάνω στην προδιαγραφή ASGI (Asynchronous Server Gateway Interface). Βασικό χαρακτηριστικό του είναι ότι αξιοποιεί τις υποδείξεις τύπων (type hints) της γλώσσας για αυτόματη επικύρωση των δεδομένων εισόδου και την παραγωγή τεκμηρίωσης σύμφωνα με το πρότυπο OpenAPI. Η Unicorn αναλαμβάνει την εκτέλεση της εφαρμογής και λειτουργεί ως διακομιστής ASGI και διαχειρίζεται την επικοινωνία της εφαρμογής με το δίκτυο [74].

Ένα κύριο χαρακτηριστικό του FastAPI που βελτιώνει τη λειτουργία του RAG συστήματος είναι η προοδευτική αποστολή της παραγόμενης απάντησης προς τον χρήστη (token streaming). Σε αντίθεση με την απλή λογική άμεσης απόκρισης, όπου ολόκληρη η απάντηση δημιουργείται πριν αποσταλεί, το πρότυπο ASGI υποστηρίζει τμηματική αποστολή δεδομένων, επιτρέποντας στη διαδικτυακή διεπαφή να εμφανίζει την απάντηση καθώς αυτή παράγεται από το γλωσσικό μοντέλο.

Ωστόσο, ο σημαντικότερος λόγος επιλογής του είναι η υποστήριξη του μηχανισμού StreamingResponse, ο οποίος μπορεί να καταναλώσει απευθείας μια απλή γεννήτρια (generator) της Python. Αυτό καθιστά τη διαδικτυακή υλοποίηση ισάξια με αυτή της γραμμής εντολών χωρίς να απαιτείται η ξεχωριστή ασύγχρονη εκδοχή του κώδικα.

Παράλληλα, στο ίδιο επίπεδο υλοποιούνται βασικοί μηχανισμοί ασφάλειας. Ειδικότερα, η πολιτική ελέγχου διασταυρούμενης προέλευσης (Cross-Origin Resource Sharing – CORS) εμπεριέχεται αποκλειστικά στη διαδικτυακή εφαρμογή ώστε να αποτρέπεται η πρόσβαση από μη εξουσιοδοτημένες ιστοσελίδες. Λόγω των περιορισμών του συγκεκριμένου συστήματος εφαρμόζεται μηχανισμός περιορισμού ρυθμού αιτημάτων (rate limiting) με τεχνική ολισθαίνοντος παραθύρου (sliding window), καθώς κάθε αίτημα δεσμεύει τους διαθέσιμους υπολογιστικούς πόρους του γλωσσικού μοντέλου για αρκετά δευτερόλεπτα. Τέλος, η αναγνώριση κάθε προγράμματος περιήγησης πραγματοποιείται μέσω ανώνυμου αναγνωριστικού, χωρίς δημιουργία λογαριασμών ή αποθήκευση προσωπικών δεδομένων, επιλογή που συνάδει με τον περιορισμό Π1 και τη φιλοσοφία τοπικής λειτουργίας του συστήματος.

3.8.2 Server-Sent Events

Στα σύγχρονα συστήματα τύπου AI chatbot υποστηρίζεται η προοδευτική αποστολή απάντησης (token streaming) κατά την οποία οι λέξεις εμφανίζονται σταδιακά ενώ παράγεται η απάντηση. Για την εφαρμογή αυτής της λειτουργίας στο σύστημα κρίθηκε απαραίτητη η εφαρμογή του προτύπου Server-Sent Events (SSE). Το SSE επιτρέπει τη δημιουργία μιας μονόδρομης ροής δεδομένων πάνω από μια απλή, ανοικτή σύνδεση HTTP, όπου ο διακομιστής αποστέλλει διαδοχικά τμήματα κειμένου [75].

Μία άλλη εναλλακτική είναι τα WebSockets, τα οποία προσφέρουν αμφίδρομο κανάλι. Ωστόσο, το συγκεκριμένο σύστημα δεν απαιτεί τη συνεχόμενη επικοινωνία πομπού και δέκτη γιατί η επικοινωνία της εφαρμογής είναι εξ ορισμού μιας κατεύθυνσης καθώς ο φοιτητής υποβάλλει ένα ερώτημα και αναμένει την ολοκλήρωση της απάντησης. Ένα αμφίδρομο κανάλι θα απαιτούσε χειραγία αναβάθμισης πρωτοκόλλου, κατάσταση σύνδεσης προς διαχείριση και ξεχωριστή διαδρομή σφαλμάτων, χωρίς να εξυπηρετεί καμία απαίτηση. Για τον λόγο αυτό δεν επιλέχθηκαν τα WebSockets, καθώς θα εισήγαγαν

περιττή πολυπλοκότητα χωρίς να καλύπτουν κάποια επιπλέον λειτουργική απαίτηση του συστήματος. Αντίθετα, το SSE καλύπτει πλήρως τις ανάγκες της εφαρμογής, παραμένοντας συμβατό με την υπάρχουσα υποδομή HTTP και με τους μηχανισμούς επικύρωσης και περιορισμού ρυθμού που εφαρμόζονται στο επίπεδο του FastAPI.

Η επιλογή του SSE συνοδεύεται από μια σημαντική τεχνική ιδιαιτερότητα στο επίπεδο της διεπαφής που περιορίζει το σύστημα έτσι όπως είναι σχεδιασμένο. Ο εγγενής μηχανισμός των προγραμμάτων περιήγησης για την κατανάλωση SSE –το EventSource API– υποστηρίζει αποκλειστικά αιτήματα τύπου GET. Λόγω της αρχιτεκτονικής στο σύστημα, υποβάλλεται ερώτημα με μέθοδο POST και σώμα JSON, τόσο εξαιτίας του μεγέθους του όσο και επειδή συνοδεύεται από παραμέτρους συνομιλίας. Η συνέπεια είναι ότι το EventSource δεν μπορεί να χρησιμοποιηθεί και η ανάγνωση της ροής υλοποιείται στην πλευρά του πελάτη με χειρόγραφο αναλυτή, ζήτημα στο οποίο επανέρχεται η ενότητα 4.6.1.

Τέλος, αν εφαρμοστεί σε περιβάλλον παραγωγής απαιτείται ιδιαίτερη προσοχή κατά την εγκατάσταση του συστήματος καθώς η τεχνική του SSE είναι ιδιαίτερα ευαίσθητη στην προσωρινή αποθήκευση δεδομένων (buffering) από ενδιάμεσους διακομιστές (reverse proxies). Εάν ένας ενδιάμεσος κόμβος έχει ενεργοποιημένο το buffering, κινδυνεύει να ακυρώσει το εφέ της σταδιακής ροής, παραδίδοντας τελικά ολόκληρη την απάντηση μονομιάς στον χρήστη. Ως εκ τούτου, η απενεργοποίηση τέτοιων μηχανισμών αποτελεί αυστηρή προϋπόθεση για τη σωστή εγκατάσταση της εφαρμογής.

3.8.3 Pydantic και διαχείριση παραμέτρων

Η Pydantic είναι βιβλιοθήκη επικύρωσης και διαχείρισης δεδομένων για την Python, η οποία αξιοποιεί τις υποδείξεις τύπων (type hints) της γλώσσας. Μια κλάση δηλώνει τα πεδία της με τους τύπους τους και η επικύρωση εκτελείται κατά τη δημιουργία του αντικειμένου, μετατρέποντας τα δεδομένα στην αναμενόμενη μορφή ή απορρίπτοντάς τα με σαφές μήνυμα. Παράλληλα, μέσω του συνοδευτικού πακέτου pydantic-settings, ο ίδιος ακριβώς μηχανισμός επικύρωσης επεκτείνεται και στη διαχείριση των ρυθμίσεων της εφαρμογής, οι οποίες αντλούνται από τις μεταβλητές περιβάλλοντος (αρχείο .env) [76].

Ο βασικός ρόλος της βιβλιοθήκης είναι η επικύρωση των εισερχόμενων αιτημάτων του API, λειτουργία την οποία το FastAPI αξιοποιεί εξ ορισμού για την επικύρωση των αιτημάτων. Ο δεύτερος ρόλος είναι η διαχείριση και η οργάνωση των παραμέτρων ολόκληρου του αγωγού (pipeline). Το γεγονός ότι η βιβλιοθήκη αξιοποιείται σε δύο περιπτώσεις ήταν συνειδητή σχεδιαστική απόφαση καθώς το σύστημα διαθέτει ένα μόνο σημείο όπου ορίζεται τι θεωρείται έγκυρη τιμή, αντί για δύο ανεξάρτητους μηχανισμούς με πιθανότητα απόκλισης. Επιπλέον, η εξωτερική αποθήκευση των παραμέτρων επιτρέπει την τροποποίηση της συμπεριφοράς της εφαρμογής χωρίς αλλαγές στον πηγαίο κώδικα, διευκολύνοντας τη δοκιμή διαφορετικών διαμορφώσεων και την αναπαραγωγή των πειραματικών αποτελεσμάτων.

3.8.4 Next.js και React

Η React αποτελεί μια σύγχρονη βιβλιοθήκη για τη δημιουργία διεπαφών χρήστη, η οποία βασίζεται σε ένα δηλωτικό μοντέλο (declarative model). Είναι προσιτή για τον προγραμματιστή καθώς απαιτεί μόνο

την περιγραφή μιας διεπαφής για μια δεδομένη κατάσταση δεδομένων και η βιβλιοθήκη αναλαμβάνει να υπολογίσει και να εφαρμόσει τις απολύτως απαραίτητες μεταβολές στο δέντρο του εγγράφου (DOM). Ο μηχανισμός αυτός, γνωστός ως συμφιλίωση (reconciliation), αποτελεί το θεμέλιο της αρχιτεκτονικής της. Ακόμη ένα πλαίσιο που συνεργάζεται με την React είναι το Next.js. Παρέχει πολλές λειτουργίες, καθιστώντας το πιο ολοκληρωμένο, όπως για παράδειγμα τη δρομολόγηση, τον διακομιστή ανάπτυξης και την ενοποιημένη διαδικασία μεταγλώττισης [77, 78].

Η επιλογή της React στην παρούσα εφαρμογή σχετίζεται με τον τρόπο με τον οποίο παρουσιάζεται η παραγόμενη απάντηση. Κατά τη ροή της απάντησης μέσω της διαδικασίας token streaming, η διεπαφή δέχεται δεκάδες ενημερώσεις ανά δευτερόλεπτο και με κάθε ενημέρωση μεταβάλλεται το τελευταίο τμήμα του κειμένου που συνεχώς αυξάνεται. Στο πλαίσιο αυτό, η χειροκίνητη διαχείριση του DOM υπό αυτές τις συνθήκες θα απαιτούσε την ανάπτυξη ειδικής λογικής για τον εντοπισμό και την ενημέρωση των μεταβαλλόμενων τμημάτων της διεπαφής. Ως εκ τούτου, το framework της React έχει μια λειτουργία που αξιοποιείται για αυτόν τον σκοπό και είναι ο υπολογισμός της ελάχιστης διαφοράς (diffing).

Αξίζει να σημειωθεί ότι πριν την επιλογή της React, είχαν δοκιμαστεί και άλλα πλαίσια εφαρμογών, όμως απορρίφθηκαν λόγω δομικών περιορισμών. Η πρώτη εκδοχή υλοποιήθηκε μέσω του Gradio, ενός δημοφιλούς εργαλείου ταχείας κατασκευής διεπαφών για μοντέλα μηχανικής μάθησης. Ωστόσο, λόγω της αυστηρότητας και των περιορισμών που είχε καθώς δεν επέτρεπε την ευέλικτη μορφοποίηση και την ορθή παρουσίαση των αναφορών και της τεκμηρίωσης, δεν υιοθετήθηκε. Η επόμενη επιλογή ήταν η JavaScript λόγω της απλότητάς της. Αποφεύχθηκε η χρήση όμως, λόγω της αυξημένης ευπάθειας σε σφάλματα συγχρονισμού: δεδομένου ότι η ροή SSE φέρει πολλαπλούς τύπους συμβάντων, κάθε μεταβολή στη δομή των δεδομένων από την πλευρά του διακομιστή προκαλούσε αθόρυβα σφάλματα κατά την εκτέλεση (runtime errors).

Παράλληλα, η υλοποίηση διατηρήθηκε περιορισμένη ως προς τις εξωτερικές εξαρτήσεις, σύμφωνα με τον περιορισμό Π5. Χρησιμοποιούνται μόνο οι απαραίτητες εξαρτήσεις για το Frontend και δεν βασίζεται σε ξεχωριστή βιβλιοθήκη για την απόδοση Markdown ή σε έτοιμο πελάτη SSE. Οι συγκεκριμένες λειτουργίες υλοποιήθηκαν τοπικά, καθώς η απαιτούμενη λειτουργικότητα είναι περιορισμένη και δεν δικαιολογεί την προσθήκη επιπλέον πακέτων. Για την αποφυγή επιθέσεων έγχυσης σεναρίων (Cross-Site Scripting – XSS) προηγείται κατάλληλη διαφυγή των ειδικών χαρακτήρων HTML πριν από την εισαγωγή στοιχείων μορφοποίησης ώστε να αποτελεί μέρος της ίδιας της διαδικασίας απόδοσης. Με τον τρόπο αυτό περιορίζεται η επιφάνεια εξαρτήσεων της εφαρμογής, χωρίς να επηρεάζεται η απαιτούμενη λειτουργικότητα της διαδικτυακής διεπαφής.

3.8.5 HTML και CSS

Η δομή και η μορφοποίηση της διαδικτυακής διεπαφής υλοποιούνται αποκλειστικά με τις βασικές τεχνολογίες του ιστού, δηλαδή HTML5 και Vanilla CSS. Δεν χρησιμοποιήθηκαν πρόσθετα πλαίσια μορφοποίησης όπως το Tailwind CSS ή το Bootstrap καθώς το οπτικό ύφος της εφαρμογής είναι αυστηρά στοχευμένο, καλύπτεται από ένα περιορισμένο σύνολο κανόνων και, συνεπώς, δεν δικαιολογεί την προσθήκη μιας εξωτερικής εξάρτησης. Η προσέγγιση αυτή εξυπηρετεί τον Περιορισμό Π5 που αναφέρεται στην ελαχιστοποίηση των εξωτερικών πακέτων [79].

Σημαντικό στοιχείο της εφαρμογής είναι η προσαρμοστικότητα της διεπαφής (responsive design), ώστε να μπορεί να χρησιμοποιηθεί από διαφορετικού τύπου συσκευές. Η διάταξη των στοιχείων προσαρμόζεται ανάλογα με το διαθέσιμο πλάτος της οθόνης, ενώ οι κανόνες μορφοποίησης οργανώνονται έτσι ώστε η αλλαγή μεγέθους να μην επηρεάζει τη βασική λειτουργικότητα της εφαρμογής. Παράλληλα, υπάρχει η επιλογή σκούρου ή ανοιχτού θέματος για τις προτιμήσεις του χρήστη, η οποία υλοποιείται μέσω μεταβλητών CSS, επιτρέποντας την αλλαγή των βασικών χαρακτηριστικών εμφάνισης από ένα κεντρικό σημείο χωρίς την ανάγκη διατήρησης ξεχωριστών συνόλων κανόνων.

Τέλος, κατά την ανάπτυξη της διεπαφής δόθηκε προσοχή στην τήρηση βασικών προδιαγραφών προσβασιμότητας (accessibility). Αυτό επετεύχθη μέσω της αυστηρής σημασιολογικής δόμησης του εγγράφου (semantic HTML) για τη σωστή ιεράρχηση της πληροφορίας, καθώς και μέσω του ελέγχου για επαρκή χρωματική αντίθεση (contrast ratio) μεταξύ του κειμένου και του φόντου. Έτσι, διασφαλίζεται ότι η παραγόμενη πληροφορία, η οποία συχνά αφορά πυκνά ακαδημαϊκά και νομικά κείμενα, παραμένει ευανάγνωστη και προσβάσιμη για όλους τους χρήστες.

3.9 Συστήματα αποθήκευσης της εφαρμογής

3.9.1 PostgreSQL, SQLAlchemy και psycopg

Η PostgreSQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων ανοικτού κώδικα, το οποίο υποστηρίζει συναλλαγές σύμφωνα με τις ιδιότητες ACID και μηχανισμούς ελέγχου ταυτοχρονισμού πολλαπλών εκδόσεων (Multi-Version Concurrency Control – MVCC). Παρέχει τη δυνατότητα της ασφαλούς διαχείρισης των δεδομένων ακόμη και όταν πραγματοποιούνται ταυτόχρονα πολλαπλές λειτουργίες ανάγνωσης και εγγραφής [80, 81].

Ένα σημαντικό χαρακτηριστικό της εφαρμογής είναι το ιστορικό συνομιλιών. Αυτό υλοποιήθηκε με τη βοήθεια της PostgreSQL, καθώς τα δεδομένα αυτά παρουσιάζουν δομημένη και σχεσιακή μορφή και κάθε συνομιλία συνδέεται με ένα ή περισσότερα μηνύματα. Παράλληλα, μέσω της βιβλιοθήκης αξιοποιείται η αναζήτηση και η ανάκτηση των μηνυμάτων που ανήκουν σε μια συνομιλία, καθώς και η διαχείριση της διαγραφής τους όταν ο χρήστης αφαιρεί το αντίστοιχο ιστορικό. Η χρήση σχεσιακής βάσης επιτρέπει οι σχέσεις μεταξύ των δεδομένων και οι κανόνες ακεραιότητάς τους να ελέγχονται από το ίδιο το σύστημα διαχείρισης βάσης δεδομένων.

Για τη συνεργασία της Python με την PostgreSQL αξιοποιείται η βιβλιοθήκη SQLAlchemy, η οποία παρέχει μηχανισμό αντικειμενοσχεσιακής αντιστοίχισης (Object-Relational Mapping – ORM). Με τη χρήση της οι οντότητες μπορούν να αναπαρασταθούν ως κλάσεις και οι εγγραφές ως αντικείμενα, επιτρέποντας τη διαχείριση των δεδομένων χωρίς την ανάγκη άμεσης σύνταξης SQL για κάθε λειτουργία, διατηρώντας παράλληλα μια ενιαία δομή μεταξύ του μοντέλου δεδομένων και του κώδικα της εφαρμογής.

Η επικοινωνία της SQLAlchemy με τον διακομιστή PostgreSQL πραγματοποιείται μέσω του psycopg, το οποίο λειτουργεί ως οδηγός σύνδεσης (database driver). Επομένως, η κάθε τεχνολογία έχει διακριτό σκοπό: η PostgreSQL παρέχει το σύστημα αποθήκευσης και διαχείρισης των δεδομένων, η SQLAlchemy αναλαμβάνει την αντιστοίχιση και τη διαχείρισή τους από την πλευρά της Python και το psycopg υλοποιεί την επικοινωνία με τον διακομιστή της βάσης. Η SQLAlchemy εξυπηρετεί επίσης

την τυποποίηση και τη συνέπεια του κώδικα. Τα μοντέλα της βάσης ορίζονται με δηλωτικό τρόπο και στα πεδία τους καθορίζονται συγκεκριμένοι τύποι, με αποτέλεσμα να είναι πιο εύκολος ο εντοπισμός σφαλμάτων κατά το στάδιο της υλοποίησης μειώνοντας την πιθανότητα σφαλμάτων κατά την εκτέλεση.

3.9.2 SQLite σε λειτουργία WAL

Η SQLite είναι ένα ενσωματωμένο σχεσιακό σύστημα βάσης δεδομένων, το οποίο διαφέρει από την PostgreSQL ως προς τον τρόπο λειτουργίας του. Δεν απαιτείται η εκτέλεση ενός εξωτερικού προγράμματος ή διεργασίας για να λειτουργήσει και αποθηκεύει τα δεδομένα σε ένα αρχείο. Επιπλέον, υποστηρίζεται από την πρότυπη βιβλιοθήκη της Python, με αποτέλεσμα να μην απαιτείται η προσθήκη ξεχωριστής εξωτερικής εξάρτησης για τη βασική λειτουργία της [82].

Στη συγκεκριμένη υλοποίηση η SQLite δεν χρησιμοποιείται για την αποθήκευση του ιστορικού των χρηστών, αλλά για την καταγραφή στοιχείων τηλεμετρίας. Για κάθε ερώτημα αποθηκεύονται σημαντικές πληροφορίες που αφορούν τις επιδόσεις της εφαρμογής και την ποιότητα των απαντήσεων. Τα δεδομένα αυτά χρησιμοποιούνται για την αξιολόγηση και την παρακολούθηση της συμπεριφοράς του συστήματος και όχι ως μέρος της κύριας λειτουργικότητας που παρέχεται στον χρήστη. Η καταγραφή αυτών των στοιχείων πραγματοποιείται με ενεργοποιημένη τη λειτουργία Write-Ahead Logging (WAL), όπου καταγράφονται αρχικά σε ξεχωριστό αρχείο και ενσωματώνονται στη συνέχεια στην κύρια βάση. Αυτό επιτρέπει την ταυτόχρονη πρόσβαση, καθώς οι λειτουργίες ανάγνωσης μπορούν να πραγματοποιούνται παράλληλα με την καταγραφή νέων δεδομένων. Επομένως, με τον διαχωρισμό των σχεσιακών βάσεων, τα στοιχεία τους διαχωρίζονται και καθίστανται ανεξάρτητα και πιο ελαφριά, χωρίς να αυξάνεται η πολυπλοκότητα.

3.10 Εργαλεία ανάπτυξης και υποδομής

3.10.1 Docker και Docker Compose

Το Docker αποτελεί μια πλατφόρμα εικονικοποίησης σε επίπεδο λειτουργικού συστήματος, η οποία επιτρέπει την εκτέλεση εφαρμογών σε απομονωμένα περιβάλλοντα, γνωστά ως containers. Πιο συγκεκριμένα, τα containers δεν απαιτούν την εκτέλεση ενός πλήρους λειτουργικού συστήματος σε αντίθεση με τις παραδοσιακές εικονικές μηχανές αλλά αξιοποιούν τον πυρήνα του συστήματος υποδοχής, περιορίζοντας έτσι το υπολογιστικό και αποθηκευτικό κόστος. Το Docker Compose συμπληρώνει τη λειτουργικότητα αυτή επιτρέποντας τον δηλωτικό καθορισμό και τη διαχείριση μίας ή περισσότερων υπηρεσιών μέσω ενός αρχείου διαμόρφωσης [83].

Στη συγκεκριμένη υλοποίηση, το Docker περιορίστηκε μόνο στην εκτέλεση της διανυσματικής βάσης δεδομένων. Η επιλογή αυτή υπαγορεύτηκε από περιορισμούς που σχετίζονται με το περιβάλλον εκτέλεσης. Ειδικότερα, σε συστήματα Apple Silicon, όπως το περιβάλλον στο οποίο αναπτύχθηκε η παρούσα εφαρμογή, το Docker εκτελείται μέσω μιας ελαφριάς εικονικής μηχανής Linux, η οποία δεν υποστηρίζει διέλευση υλικού προς τη μονάδα γραφικών (GPU passthrough), όπως συμβαίνει σε περιβάλλοντα CUDA. Κατά συνέπεια, η μεταφορά μοντέλων μηχανικής μάθησης σε περιβάλλον container θα περιόριζε την εκτέλεσή τους σε χρήση της κεντρικής μονάδας επεξεργασίας (CPU), με σημαντική επιβάρυνση του χρόνου εκτέλεσης. Ως εκ τούτου, υιοθετήθηκε μια υβριδική τοπολογία όπου η υποδομή αποθήκευσης εκτελείται σε container, ενώ η εκτέλεση των μοντέλων (inference) πραγματοποιείται εγγενώς στο λειτουργικό σύστημα macOS.

3.10.2 launchd

Για τον χρονοπρογραμματισμό και την αυτοματοποίηση της συλλογής δεδομένων, επιλέχθηκε το launchd, το οποίο είναι το εγγενές πρόγραμμα διαχείρισης υπηρεσιών του macOS. Προσφέρει πολλές διευκολύνσεις και λειτουργίες όπως η δηλωτική περιγραφή των εργασιών, η βελτιωμένη διαχείριση εκτέλεσης καθώς και η ενσωματωμένη καταγραφή σφαλμάτων και η δυνατότητα αυτόματης επανεκκίνησης σε περίπτωση διακοπής. Αξίζει να σημειωθεί ότι κατά τη διάρκεια της υλοποίησης προέκυψε ένας τεχνικός περιορισμός σχετικά με τα δικαιώματα του λειτουργικού συστήματος. Ο μηχανισμός προστασίας προσωπικών δεδομένων του macOS (TCC – Transparency, Consent, and Control) απαγορεύει σε προγραμματισμένες εργασίες του παρασκηνίου την πρόσβαση σε προστατευμένους καταλόγους του χρήστη (όπως ο κατάλογος ~/Documents). Το γεγονός αυτό κατέστησε αναγκαία τη μεταφορά του έργου σε μη προστατευμένη τοποθεσία του δίσκου [84].

3.10.3 Pytest

Το pytest αποτελεί πλαίσιο δοκιμών για εφαρμογές Python και παρέχει έναν απλό μηχανισμό ορισμού και εκτέλεσης αυτοματοποιημένων δοκιμών. Οι δοκιμές μπορούν να διατυπωθούν ως συναρτήσεις και να επαληθευτεί η προβλεπόμενη συμπεριφορά τους με τη χρήση της εντολής assert. Παράλληλα, παρέχεται η δυνατότητα χρήσης στερεωμάτων (fixtures) για την προετοιμασία των απαιτούμενων δεδομένων και του περιβάλλοντος εκτέλεσης [85].

Η χρήση των αυτοματοποιημένων δοκιμών κρίνεται αναγκαία για τη συγκεκριμένη εφαρμογή, καθώς απαιτείται η διαρκής επαλήθευση των δεδομένων σε πολλά σημεία καθώς και η διασφάλιση της ορθότητας των αποτελεσμάτων. Λόγω του ντετερμινιστικού χαρακτήρα του συστήματος είναι αναγκαία αυτή η τακτική καθώς εξυπηρετεί λειτουργίες των οποίων το αναμενόμενο αποτέλεσμα μπορεί να καθοριστεί με σαφήνεια, όπως η επεξεργασία γεγονότων, η ανάλυση και αντιστοίχιση ονομάτων, οι υπολογισμοί πιστωτικών μονάδων και η προεπεξεργασία ελληνικού κειμένου πριν από την εφαρμογή του BM25.

Σε αντίθεση με τις δοκιμές μονάδας του συστήματος, η παραγωγή απαντήσεων από το μοντέλο αξιολογείται διαφορετικά. Η παραγωγή φυσικής γλώσσας δεν είναι πλήρως ντετερμινιστική ως προς τη διατύπωση και, επομένως, η απαίτηση για ακριβή ταύτιση ενός παραγόμενου κειμένου με μία προκαθορισμένη απάντηση δεν αποτελεί κατάλληλο κριτήριο ελέγχου. Η ποιότητα των παραγόμενων απαντήσεων επαληθεύεται μέσω κατάλληλων μετρικών και ενός προκαθορισμένου συνόλου δεδομένων ελέγχου, όπως παρουσιάζεται στο αντίστοιχο κεφάλαιο αξιολόγησης.

3.10.4 Git

Το Git αποτελεί κατανεμημένο σύστημα ελέγχου εκδόσεων, στο οποίο κάθε τοπικό αντίγραφο ενός αποθετηρίου περιλαμβάνει το ιστορικό των μεταβολών του έργου. Η χρήση του επιτρέπει την παρακολούθηση των αλλαγών στον πηγαίο κώδικα, τη διατήρηση προηγούμενων εκδόσεων και την ασφαλή ανάπτυξη διαφορετικών εκδοχών της εφαρμογής. Στο συγκεκριμένο σύστημα αξιοποιήθηκε για τη διαχείριση του πηγαίου κώδικα καθώς και για την καταγραφή της εξέλιξης του συστήματος κατά τη διάρκεια της ανάπτυξής του. Με αυτόν τον τρόπο διευκολύνει την ασφαλή δοκιμή εναλλακτικών προσεγγίσεων, χωρίς να επηρεάζεται άμεσα η σταθερή έκδοση του συστήματος, καθώς μία αλλαγή μπορεί να αξιολογηθεί διατηρώντας παράλληλα τις προηγούμενες λειτουργικές εκδόσεις. Παράλληλα,

παρέχεται η δυνατότητα επαναφοράς προηγούμενων εκδόσεων όταν απαιτείται η επανεξέταση μιας διαφορετικής λύσης [86].

3.10.5 Loguru

Η Loguru [87] αποτελεί βιβλιοθήκη καταγραφής συμβάντων (logging) για εφαρμογές Python, η οποία απλοποιεί την παρακολούθηση της εκτέλεσης και τον εντοπισμό πιθανών σφαλμάτων. Παρέχει μηχανισμούς για την καταγραφή μηνυμάτων και εξαιρέσεων, καθώς και για την αυτόματη διαχείριση των αρχείων καταγραφής [87].

Στην παρούσα εφαρμογή, η Loguru χρησιμοποιείται κατά τη συλλογή και επεξεργασία των εγγράφων, επιτρέποντας την παρακολούθηση της εκτέλεσης των επιμέρους διαδικασιών και τη διευκόλυνση του εντοπισμού προβλημάτων. Η χρήση της περιορίζεται στο συγκεκριμένο στάδιο και δεν αφορά την καταγραφή της αλληλεπίδρασης του χρήστη με το σύστημα, για την οποία χρησιμοποιείται ξεχωριστός μηχανισμός τηλεμετρίας.

Κεφάλαιο 4ο: Σχεδίαση και υλοποίηση του συστήματος

4.1 Λειτουργικές απαιτήσεις

Ο σχεδιασμός του συστήματος καθορίστηκε από τέσσερις λειτουργικές απαιτήσεις, οι οποίες διατυπώθηκαν πριν από την επιλογή τεχνολογιών και παρέμειναν σταθερές καθ' όλη τη διάρκεια της ανάπτυξης.

A1. Τεκμηριωμένη απάντηση σε ερωτήματα φυσικής γλώσσας. Το σύστημα οφείλει να απαντάει σε ερωτήματα διατυπωμένα στα ελληνικά και να βασίζει τις απαντήσεις του αποκλειστικά σε υλικό που αντλεί από τα έγγραφα του τμήματος, όπως ανακοινώσεις, κανονισμοί, ωρολόγια προγράμματα και όχι από τις γνώσεις που έχει το γλωσσικό μοντέλο στα βάθη του. Για τον σκοπό αυτό υιοθετήθηκε η αρχιτεκτονική ανάκτησης με ενισχυμένη παραγωγή αντί για την απευθείας χρήση του γλωσσικού μοντέλου για τους λόγους που αναπτύχθηκαν στην υποενότητα 2.7.1.

A2. Ενοποίηση ετερογενών πηγών. Το υλικό που χρειάζεται για να συνταχθεί μια απάντηση προς τον χρήστη εντοπίζεται κατακερματισμένο σε πολλαπλές ιστοσελίδες και αρχεία, καθώς και σε διαφορετικές μορφές. Το σύστημα καλείται να τις ενοποιεί σε ένα ενιαίο σημείο αναζήτησης, ώστε ο χρήστης να μην χρειάζεται να αναζητεί μόνος του την πηγή της απάντησης, καθώς αυτή του παρέχεται ενσωματωμένη στην απάντηση.

A3. Πλήρως τοπική εκτέλεση. Δεν επιτρέπεται καμία κλήση σε εξωτερική υπηρεσία κατά τη λειτουργία του συστήματος, παρά μόνο η περιοδική συλλογή δεδομένων από τις πηγές του τμήματος. Η απαίτηση δεν είναι τεχνική προτίμηση, αλλά συνέπεια της προστασίας των δεδομένων των φοιτητών, κατά τα αναφερόμενα στην υποενότητα 2.2.3, και εξετάζεται ως σχεδιαστικός περιορισμός στην ενότητα 3.2.

A4. Αυτόματη ενημέρωση χωρίς χειροκίνητη συντήρηση. Το τμήμα δημοσιεύει συνεχώς νέο περιεχόμενο και τροποποιεί υπάρχουσα πληροφορία, γι' αυτό το σύστημα δεσμεύεται να ανιχνεύει και να ενσωματώνει τις μεταβολές αυτόματα. Παράλληλα, είναι σημαντικό να μην αποθηκεύονται

διπλότυπα έγγραφα, καθώς και να μη χάνεται σημαντική πληροφορία κατά το στάδιο συλλογής των δεδομένων.

Οι τέσσερις αυτές απαιτήσεις είναι αλληλένδετες μεταξύ τους. Ειδικότερα, η A3 περιορίζει τις επιλογές για την ικανοποίηση της A1, επειδή αποκλείει τη χρήση των ισχυρότερων (εμπορικών) μοντέλων και επιβάλλει τη λειτουργία του συστήματος με αυστηρά όρια μνήμης. Επιπρόσθετα, η A4 συμπληρώνει την A2 με μια χρονική διάσταση: ο στόχος δεν είναι μόνο η αρχική συγκέντρωση των πηγών, αλλά και η συνεχής ενημέρωση του συστήματος, ώστε να συμβαδίζει με τις αλλαγές των εγγράφων σε βάθος χρόνου.

4.1.1 Συνολική αρχιτεκτονική

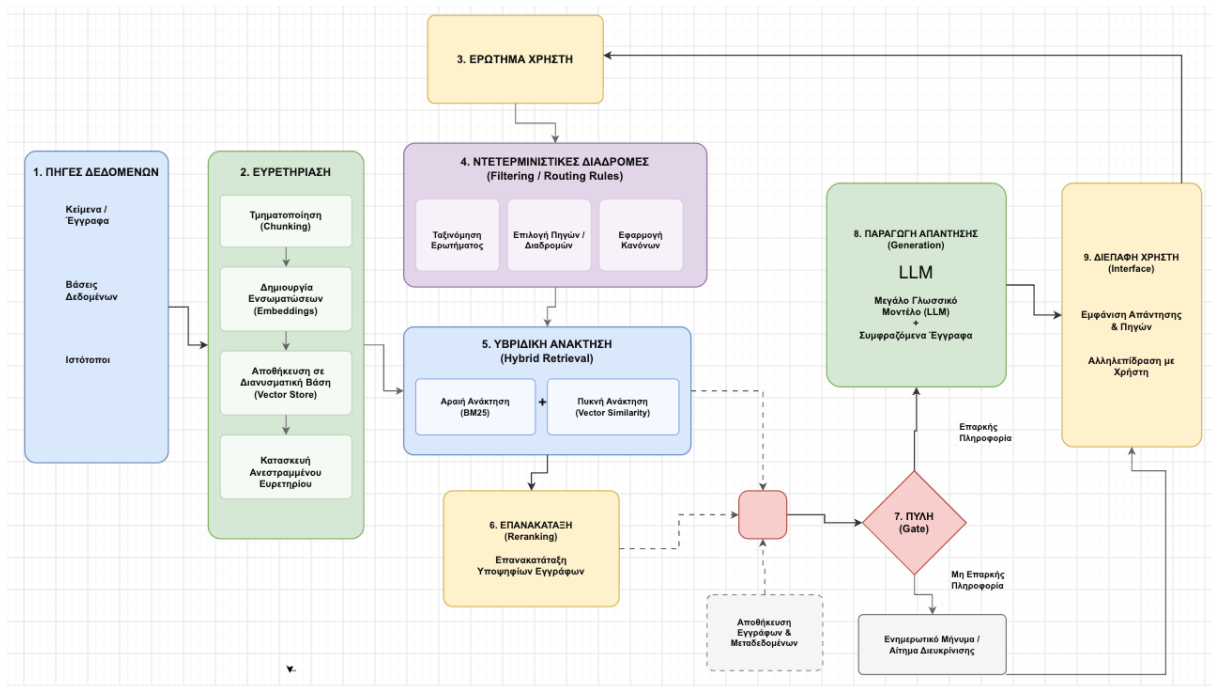
Το παρόν σύστημα αποτελείται από δύο βασικούς αγωγούς επεξεργασίας, τον αγωγό ευρετηρίασης και τον αγωγό απάντησης, οι οποίοι έχουν ως κοινό σημείο αποθήκευσης τη βάση διανυσμάτων. Ο χρήστης έχει πρόσβαση στην εφαρμογή μέσω της διαδικτυακής διεπαφής, η οποία εκθέτει τη λειτουργικότητα του αγωγού απάντησης.

Ο αγωγός ευρετηρίασης (indexing pipeline) εκτελείται περιοδικά και ανεξάρτητα από το υπόλοιπο σύστημα. Οι συλλέκτες αντλούν το περιεχόμενο από τις διαθέσιμες πηγές, ενώ στη συνέχεια ειδικοί αναλυτές εξάγουν το κείμενο ανεξαρτήτως μορφοτύπου. Τα έγγραφα εμπλουτίζονται με μεταδεδομένα, τεμαχίζονται σε αποσπάσματα ανάλογα με τον τύπο τους, μετατρέπονται σε διανυσματικές αναπαραστάσεις (embeddings) και, τέλος, εισάγονται στη βάση. Η διαδικασία εκτελείται σειριακά σε διακριτά στάδια. Σε περίπτωση αποτυχίας οποιουδήποτε μέρους, η εκτέλεση τερματίζεται άμεσα προκειμένου να αποτραπεί η δημιουργία ενός ημιτελούς συνόλου δεδομένων. Η εκτέλεση της ακολουθίας αυτής πραγματοποιείται συνήθως κατά τις νυχτερινές ώρες μέσω του αντίστοιχου χρονοπρογραμματιστή.

Από την άλλη πλευρά, ο αγωγός απάντησης εκτελείται για κάθε ερώτημα και οργανώνεται σε δύο σκέλη με διαφορετική λογική το καθένα. Το πρώτο είναι το ντετερμινιστικό σκέλος του αγωγού, το οποίο είναι υπεύθυνο για την απάντηση σε τυποποιημένα ερωτήματα, αξιοποιώντας μια προκαθορισμένη ακολουθία ελεγκτών. Κάθε ελεγκτής επιστρέφει μία απάντηση εφόσον ταυτοποιήσει το ερώτημα με απόλυτη βεβαιότητα, διαφορετικά παραχωρεί τη σειρά του στον επόμενο. Όταν κανένας ελεγκτής δεν αναλάβει να απαντήσει στο ερώτημα, ενεργοποιείται το σημασιολογικό σκέλος (RAG). Το ερώτημα κανονικοποιείται και ταξινομείται ως προς την πρόθεσή του και στη συνέχεια ανακτώνται τα υποψήφια αποσπάσματα μέσω υβριδικής αναζήτησης. Τα τμήματα αυτά επανακατατάσσονται (re-ranking) μέσω ενός διασταυρούμενου κωδικοποιητή (cross-encoder) και τα πιο συναφή με το ερώτημα δρομολογούνται στο γλωσσικό μοντέλο για την τελική σύνθεση της απάντησης.

Ένα αναπόσπαστο μέρος του συστήματος είναι η διαδικτυακή εφαρμογή. Η διεπαφή προγραμματισμού (API) εκθέτει τη ροή απάντησης ως μια συνεχή ροή συμβάντων (server-sent events) προς τη διεπαφή του χρήστη (UI). Αυτό επιτρέπει τη σταδιακή εμφάνιση της απάντησης, δίνοντας την αίσθηση στον χρήστη ότι παράγεται χωρίς καθυστέρηση.

Η συνολική αρχιτεκτονική παρουσιάζεται στο Σχήμα 4.1.



Σχήμα 4.1: Συνολική αρχιτεκτονική του συστήματος

Στα αριστερά παρουσιάζεται ο αγωγός ευρετηρίασης με τα επιμέρους στάδιά του, στα δεξιά ο αγωγός απάντησης με τα δύο σκέλη επεξεργασίας και στο κέντρο η βάση δεδομένων διανυσμάτων ως κοινή υποδομή αποθήκευσης. Τα στοιχεία που εκτελούνται εκτός σύνδεσης αποτυπώνονται με διακεκομμένο περίγραμμα.

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ότι τα δύο σκέλη του αγωγού απάντησης έχουν διαφορετικές αρμοδιότητες και εφαρμόζονται για διαφορετικούς σκοπούς. Το ντετερμινιστικό σκέλος καλύπτει ένα περιορισμένο αλλά συχνό σύνολο ερωτημάτων, όπως στοιχεία επικοινωνίας, ιδιότητα διδάσκοντα, ώρες μαθημάτων και πιστωτικές μονάδες, για τα οποία η απάντηση μπορεί να εξαχθεί πιο άμεσα χωρίς να υπάρχει η καθυστέρηση του γλωσσικού μοντέλου. Για τέτοιου είδους περιπτώσεις, η προσφυγή σε γλωσσικό μοντέλο θα προσέθετε υπολογιστικό κόστος και πιθανότητα σφάλματος χωρίς αντίστοιχο όφελος. Για τις υπόλοιπες ερωτήσεις αξιοποιείται το σκέλος ανάκτησης και παραγωγής, καθώς για την απάντησή τους απαιτείται επεξεργασία και αξιοποίηση του διαθέσιμου κειμένου.

4.1.2 Αρχές σχεδίασης

Τέσσερις βασικές αρχές διέπουν την υλοποίηση του συστήματος και αποτελούν τη βάση για αρκετές από τις σχεδιαστικές επιλογές που παρουσιάζονται στις επόμενες ενότητες.

Μία από τις αρχές αυτές είναι ο ντετερμινισμός. Σε περιπτώσεις όπου μια απάντηση μπορεί να παραχθεί με ασφάλεια μέσω αναζήτησης σε δομημένα δεδομένα, επιλέγεται ντετερμινιστική προσέγγιση. Το γλωσσικό μοντέλο αξιοποιείται μόνο όταν απαιτείται η κατανόηση του νοηματικού πλαισίου της ερώτησης, καθώς και η σύνταξη μιας σύνθετης απάντησης. Η αρχή αυτή απορρέει από τη διαπίστωση που έγινε στην υποενότητα 2.7.3: ένα μοντέλο που δεν καλείται να απαντήσει δεν μπορεί να παράγει παραισθήσεις ή μη θεμελιωμένες απαντήσεις στο συγκεκριμένο ερώτημα.

Παράλληλα, σε περίπτωση που χρειαστεί να γίνει κάποια τεχνική αλλαγή στην εφαρμογή, υπάρχει πάντα η δυνατότητα αναστροφής της συγκεκριμένης αλλαγής. Κάθε νέα αλλαγή ή προσθήκη ελέγχεται μέσω ανεξάρτητων παραμέτρων διαμόρφωσης, διασφαλίζοντας ότι, εάν η αλλαγή δεν επιφέρει τα επιθυμητά αποτελέσματα, η προηγούμενη συμπεριφορά μπορεί να επανέλθει άμεσα και με ασφάλεια. Για να υιοθετηθεί μια νέα συμπεριφορά στο σύστημα, πραγματοποιείται μια διαδικασία ελέγχου και αξιολόγησης για να επαληθευτεί η λειτουργία της, αξιοποιώντας τις παραμέτρους που αναφέρθηκαν. Με βάση αυτή τη λογική, το αρχείο διαμόρφωσης λειτουργεί ως ένας πλήρης κατάλογος των αναστρέψιμων αποφάσεων του συστήματος.

Σημαντικό στοιχείο είναι επίσης η ύπαρξη ενός κοινού μηχανισμού για την αξιολόγηση και την παραγωγή. Ο μηχανισμός ανάκτησης που χρησιμοποιείται κατά τη διαδικασία της αξιολόγησης (evaluation) είναι ακριβώς ο ίδιος με εκείνον της παραγωγικής λειτουργίας (production). Αυτό κρίθηκε απαραίτητο διότι οποιαδήποτε διαφορά στον κώδικα θα προκαλούσε αποκλίσεις μεταξύ των δύο περιβαλλόντων και τη διαμόρφωση διαφορετικού συμπεράσματος.

Επιπροσθέτως, οι μετασχηματισμοί του ερωτήματος εφαρμόζονται μόνο όταν κρίνεται απαραίτητο, καθώς μπορεί να αλλοιώσουν το περιεχόμενό του χωρίς ουσιαστικό λόγο. Πιο συγκεκριμένα, το ερώτημα του χρήστη επεξεργάζεται και αναδιατυπώνεται με σκοπό την καλύτερη κατανόησή του. Εφαρμόζεται, ωστόσο, μόνο όταν το βασικό στάδιο επεξεργασίας έχει ήδη δώσει ένδειξη ότι δεν μπορεί να οδηγήσει σε ικανοποιητικό αποτέλεσμα. Με τον τρόπο αυτό μειώνεται ο κίνδυνος ενός αχρείαστου μετασχηματισμού του ερωτήματος, το οποίο θα μπορούσε να απαντηθεί σωστά χωρίς πρόσθετη επεξεργασία.

4.1.3 Οργάνωση του κώδικα

Ο πηγαίος κώδικας οργανώνεται σε τέσσερις καταλόγους πρώτου επιπέδου που περιέχουν εκτελέσιμο κώδικα και περιλαμβάνει 68 αρχεία Python, χωρίς να προσμετρώνται τα σενάρια αξιολόγησης και οι μοναδιαίοι έλεγχοι.

Ο κατάλογος src/ περιέχει την κύρια λογική του συστήματος, οργανωμένη σε επτά υποπακέτα που αντιστοιχούν στα βασικά στάδια των δύο αγωγών επεξεργασίας. Σε αυτά περιλαμβάνονται η συλλογή και ανάλυση εγγράφων, η επεξεργασία δομημένου περιεχομένου, η δημιουργία διανυσματικών αναπαραστάσεων, η διαχείριση της βάσης δεδομένων διανυσμάτων, η λειτουργία του γλωσσικού μοντέλου, η λογική του συστήματος διαλόγου και βοηθητικές λειτουργίες.

Ο κατάλογος webapp/ περιλαμβάνει τη διεπαφή προγραμματισμού εφαρμογών, τη σύνδεση του αγωγού απάντησης με τη ροή συμβάντων προς τη διεπαφή χρήστη, καθώς και τη διαχείριση της πρόσβασης στη βάση δεδομένων συνομιλιών. Ο κατάλογος evaluation/ περιέχει τον μηχανισμό αξιολόγησης και τα σενάρια που χρησιμοποιούνται για τις συγκριτικές δοκιμές. Ο κατάλογος tests/ περιλαμβάνει τους μοναδιαίους ελέγχους των επιμέρους λειτουργιών, ενώ ο scripts/ περιέχει βοηθητικά σενάρια που χρησιμοποιούνται για λειτουργίες διαχείρισης και εκτέλεσης.

Στο ανώτατο επίπεδο του έργου βρίσκονται οι βασικοί ενορχηστρωτές των δύο αγωγών, καθώς και το αρχείο διαμόρφωσης που καθορίζει τις παραμέτρους λειτουργίας του συστήματος.

4.2 Το σώμα και ο αγωγός ευρετηρίασης

Το γνωσιακό υπόβαθρο του συστήματος δεν βασίστηκε σε κάποια έτοιμη συλλογή δεδομένων, αλλά δημιουργήθηκε με τη σύνθεση πέντε ετερογενών πηγών. Οι πηγές αυτές παρουσιάζουν μεγάλη ποικιλομορφία ως προς τη μορφή των δεδομένων τους, τις μεθόδους προσπέλασής τους, αλλά και τον κύκλο ζωής των πληροφοριών που περιέχουν. Για παράδειγμα, οι ανακοινώσεις και το ωρολόγιο πρόγραμμα μαθημάτων αλλάζουν συχνά μέσα στο εξάμηνο, αλλά κείμενα όπως το κεντρικό νομοθετικό πλαίσιο ή οι πάγιοι κανονισμοί λειτουργίας παραμένουν αμετάβλητα για πολλά χρόνια. Η ανάγκη οι ασύμβατες πληροφορίες μεταξύ τους να συνυπάρχουν σε ένα κεντρικό σημείο αναζήτησης (απαίτηση A2), σε συνδυασμό με την αυτόματη ενημέρωσή τους χωρίς την ανθρώπινη παρέμβαση (Απαίτηση A4), αποτέλεσαν τους καθοριστικούς παράγοντες για τον σχεδιασμό της αρχιτεκτονικής του αγωγού ευρετηρίασης (indexing pipeline), επισκιάζοντας οποιαδήποτε άλλη επιμέρους τεχνολογική επιλογή.

4.2.1 Πηγές και συλλογή εγγράφων

Στον παρακάτω πίνακα συνοψίζονται οι βασικές κατηγορίες των πηγών που τροφοδοτούν το σύστημα, μαζί με τις μεθόδους ανάκτησης και τους αντίστοιχους τύπους εγγράφων:

Πίνακας 4.1: Κατηγορίες πηγών, μέθοδοι ανάκτησης και τύποι εγγράφων

Κατηγορία Πηγής	Μέθοδος Ανάκτησης / Συλλογής	Τύποι Εγγράφων
Πλατφόρμα Ανακοινώσεων (Δυναμικό περιεχόμενο, επισυναπτόμενα PDF)	Αρχική ταυτοποίηση μέσω αυτοματοποιημένου φυλλομετρητή και εν συνεχεία τυπικά HTTP αιτήματα.	announcement, announcement_attachment
Στατικός Ιστότοπος Τμήματος (Προσωπικό, μαθήματα, εργαστήρια)	Πλοήγηση μέσω HTTP αιτημάτων και συντακτική ανάλυση (parsing) του HTML κώδικα.	curriculum, staff, lab, guide, infrastructure, contact, general
Πρόγραμμα Διδασκαλίας/Εξετάσεων (PDF)	Εξαγωγή δεδομένων μορφής πίνακα και μετατροπή τους σε δομή JSON.	schedule
Στατικά Έγγραφα PDF (Κανονισμοί, πρακτική άσκηση, οδηγοί)	Απευθείας τοποθέτηση στον κατάλογο εισαγωγής και επεξεργασία κατά τη φάση ευρετηρίασης.	regulation, guide, curriculum, official_document
Κυβερνητικές/Υπουργικές Πύλες (Νομοθεσία, ΕΣΠΑ, φοιτητική μέριμνα)	HTTP αιτήματα συνδυασμένα με μηχανισμό ελέγχου κατακερματισμών (hashes) για εντοπισμό αλλαγών.	regulation, law, iky, espa, infrastructure

Από τις παραπάνω πηγές, η πλατφόρμα των ανακοινώσεων αποτελεί την πιο σύνθετη περίπτωση, καθώς απαιτεί αυθεντικοποίηση και εκτέλεση κώδικα JavaScript. Προκειμένου να αποφευχθεί η υψηλή κατανάλωση πόρων που απαιτεί η συνεχής χρήση ενός αυτοματοποιημένου φυλλομετρητή (headless browser), επιλέχθηκε μια υβριδική προσέγγιση. Ο φυλλομετρητής χρησιμοποιείται αποκλειστικά για το σύνθετο στάδιο της διαδικασίας αυθεντικοποίησης. Αφού ολοκληρωθεί αυτό το βήμα, τα cookies της συνεδρίας μεταβιβάζονται σε έναν ελαφρύ HTTP client, ο οποίος και εκτελεί τη λήψη των σελίδων. Με

αυτόν τον τρόπο μειώνεται το υπολογιστικό κόστος, καθώς ο browser αξιοποιείται μόνο κατά το στάδιο της σύνδεσης.

Ο επόμενος στόχος είναι ο ιστότοπος του τμήματος. Για την ανάκτηση των πληροφοριών του υλοποιήθηκε ένας εξειδικευμένος μηχανισμός ανίχνευσης (crawler) που ακολουθεί στρατηγική αναζήτησης κατά πλάτος (breadth-first) με αυστηρά ορισμένο βάθος και σε προκαθορισμένες, επιτρεπτές διαδρομές. Παράλληλα, διαδρομές που εμπεριέχουν ανακοινώσεις του τμήματος παραλείπονται σκόπιμα, καθώς ανακτώνται ήδη από τη σελίδα ανακοινώσεων του τμήματος και η επανάληψή τους θα οδηγούσε σε διπλοεγγραφές. Μετά την ανάκτηση, το περιεχόμενο, το οποίο βρίσκεται σε μορφή HTML, μετατρέπεται σε κατάλληλη μορφή Markdown που είναι ιδανική για την κατανόηση από το μοντέλο, διατηρώντας τη δομή των επικεφαλίδων. Αυτό πραγματοποιείται σκόπιμα, καθώς η ιεραρχική σχέση μεταξύ των τίτλων αξιοποιείται από τα επόμενα στάδια επεξεργασίας κατά τον τεμαχισμό του περιεχομένου, όπως αναλύεται στην Ενότητα 4.2.4.

Για τα έγγραφα που προέρχονται από υπουργικές και κυβερνητικές πηγές, η αντιμετώπισή τους είναι διαφορετική, καθώς παρουσιάζουν χαμηλότερη συχνότητα αλλαγών. Πιο συγκεκριμένα, μπορεί να ανανεώνονται πιο σπάνια, όμως οι τροποποιήσεις που πραγματοποιούνται σε αυτές μπορεί να έχουν σημαντικό περιεχόμενο. Για κάθε πηγή ξεχωριστά διατηρείται ευρετήριο κατακερματισμών SHA-256, με αποτέλεσμα η συλλογή να παραλείπει τα δεδομένα που δεν έχουν μεταβληθεί από τον προηγούμενο έλεγχο. Ωστόσο, η μέθοδος αυτή δεν μπορεί να βασίζεται πάντοτε στα ακατέργαστα bytes της απόκρισης, καθώς κάθε αίτημα που γίνεται περιλαμβάνει διαφορετικό δυναμικό διακριτικό, με αποτέλεσμα τα bytes της ίδιας σελίδας να διαφέρουν σε κάθε ανάκτηση. Για αυτόν ακριβώς τον λόγο, ο κατακερματισμός του HTML υπολογίζεται στο εξαγόμενο κείμενο, το οποίο παραμένει σταθερό όταν δεν έχει αλλάξει πραγματικά το περιεχόμενο, ενώ στα PDF χρησιμοποιούνται τα αρχικά bytes του αρχείου.

Ανάλογη μεταχείριση χρειάστηκε και για τον διακομιστή που χρησιμοποιείται για την πρακτική άσκηση, λόγω του προβλήματος που προέκυψε με την αλυσίδα πιστοποίησης TLS. Ο συγκεκριμένος διακομιστής αποστέλλει μόνο το πιστοποιητικό του φύλλου χωρίς το απαιτούμενο ενδιάμεσο πιστοποιητικό. Η σύνδεση γίνεται αποδεκτή από τον περιηγητή, όμως συνεχίζει να απορρίπτεται από την Python. Η απενεργοποίηση της επαλήθευσης θα αποτελούσε μια εύκολη λύση, ωστόσο θα καθιστούσε ευάλωτο όλο τον συλλέκτη. Για τον λόγο αυτό, το πρόβλημα αντιμετωπίστηκε στοχευμένα. Το ενδιάμεσο πιστοποιητικό που έλειπε ενσωματώθηκε τοπικά στη δέσμη αξιοπιστίας (trust bundle) της εφαρμογής, διατηρώντας τους αυστηρούς ελέγχους ασφαλείας ενεργούς για όλες τις συνδέσεις.

Η συνολική διαδικασία ευρετηρίασης οργανώνεται μέσω ενός ενορχηστρωτή, ο οποίος αποτελείται από τέσσερα διαδοχικά στάδια. Κάθε στάδιο αποτελεί ανεξάρτητη διεργασία και εκτελείται σειριακά, καθώς η αποτυχία οποιουδήποτε βήματος τερματίζει ολόκληρη την εκτέλεση. Η επιλογή αυτή αποσκοπεί στην αποφυγή δημιουργίας ενός μερικώς ενημερωμένου σώματος. Ένα corpus που περιέχει μόνο μέρος των νέων δεδομένων είναι δυσκολότερο να εντοπιστεί και να αξιολογηθεί σε σχέση με ένα πλήρως παλαιότερο αλλά συνεπές corpus. Η ενημέρωση του συνόλου των ανακοινώσεων πραγματοποιείται καθημερινά, καθώς προστίθενται συνεχώς νέες ανακοινώσεις, ενώ οι στατικές σελίδες, όπως ο ιστότοπος του τμήματος και το πρόγραμμα μαθημάτων, ανανεώνονται χειροκίνητα, καθώς η συχνή ανανέωσή τους δεν είναι αναγκαία, διότι οι μεταβολές τους είναι περιορισμένες μέσα στο έτος.

Τέλος, σημαντικό μέρος της διαδικασίας είναι η αποθήκευση των αρχείων σε αντίστοιχους καταλόγους. Κάθε τύπος αρχείου αποθηκεύεται σε κατάλληλους καταλόγους, όπου ο καθένας δηλώνει το στάδιο της διαδικασίας στο οποίο βρίσκεται το αρχείο. Η μετακίνηση αυτή λειτουργεί ως ένδειξη ότι το συγκεκριμένο αρχείο έχει ήδη υποστεί επεξεργασία και έχει ολοκληρώσει τον κύκλο ευρετηρίασής του. Με αυτόν τον τρόπο το σύστημα αντιλαμβάνεται εύκολα ποια αρχεία εκκρεμούν και ποια έχουν ήδη ολοκληρωθεί, χωρίς να απαιτείται η διατήρηση μιας επιπλέον λίστας ή βάσης δεδομένων που θα καταγράφει ξεχωριστά την κατάσταση κάθε αρχείου. Η απλή επιλογή αυτή απλοποιεί τη διαχείριση του αγωγού και μειώνει την πιθανότητα ασυμφωνίας μεταξύ της κατάστασης των αρχείων και των δεδομένων που έχουν εισαχθεί στη διανυσματική βάση.

4.2.2 Εξαγωγή κειμένου και διαδοχή αναλυτών

Η διαδικασία εξαγωγής κειμένου από αρχεία PDF υλοποιείται μέσω του αρχείου `pdfextractor.py` και σε περίπτωση αποτυχίας ακολουθεί προκαθορισμένη σειρά εκτέλεσης. Στη διαδρομή ευρετηρίασης εφαρμόζεται πρώτα το `Docling` ως κύρια επιλογή και, εφόσον αυτό αποτύχει, αξιοποιείται το `pdfplumber` ως εφεδρική λύση. Το `mlx_vision_parser.py` ενεργοποιείται χωριστά, σε έγγραφα των οποίων η ονομασία υποδηλώνει ωρολόγιο πρόγραμμα, όπου τα συγχωνευμένα κελιά των πινάκων δεν είναι αναγνωρίσιμα από τους συμβατικούς αναλυτές

Πίνακας 4.2: Βαθμίδες εξαγωγής κειμένου από αρχεία PDF

Βαθμίδα	Μηχανισμός Εξαγωγής	Κριτήριο Ενεργοποίησης
1	Docling , με ανάλυση της διάταξης του εγγράφου μέσω μοντέλου	Προεπιλεγμένη μέθοδος λειτουργίας.
2	mlx_vision_parser.py , με απόδοση της σελίδας σε εικόνα και επεξεργασία από τοπικό πολυτροπικό μοντέλο	Όταν επαληθεύεται η συνάρτηση ελέγχου <code>_is_complex_pdf()</code> .
3	pdfplumber — Συμβατική μέθοδος εξαγωγής κειμένου με χρήση της συνάρτησης <code>extract_tables()</code> .	Εφεδρική λύση (fallback μηχανισμός) σε περίπτωση αποτυχίας των προηγούμενων.

Το `Docling` επιλέχθηκε κυρίως γιατί αναγνωρίζει τη διάταξη ενός PDF και όχι μόνο το περιεχόμενό του ως απλή ακολουθία χαρακτήρων. Παράλληλα, σε ιδιαίτερες περιπτώσεις όπου τα κελιά πινάκων είναι συγχωνευμένα και δεν διακρίνονται καθαρά, αξιοποιείται το `mlx_vision_parser.py`, το οποίο, με τη χρήση της `pdf2image`, εξάγει τη σελίδα του PDF σε μορφή εικόνας και στη συνέχεια ένα τοπικό πολυτροπικό μοντέλο τη μετατρέπει σε μορφή Markdown. Η επιλογή ενός τοπικού μοντέλου για τη διαδικασία της αναγνώρισης εγγράφου αποτελεί μέρος της απαίτησης A3 για την τοπική επεξεργασία των δεδομένων.

Ως εφεδρική λύση των παραπάνω μηχανισμών ορίστηκε το `pdfplumber`. Με τη χρήση της μεθόδου `extract_tables()`, εξάγεται το περιεχόμενο των πινάκων ενός PDF σε περίπτωση αποτυχίας του `Docling` και του `mlx_vision_parser.py`. Ωστόσο, απαιτείται ιδιαίτερη προσοχή στη συμπεριφορά του μηχανισμού εφεδρείας, καθώς η μετάβαση σε κατώτερη βαθμίδα δεν ήταν άμεσα εμφανής. Ως αποτέλεσμα, οι μεταβολές στις μετρήσεις αξιολόγησης μπορούσαν να εμφανιστούν χωρίς να είναι εξαρχής προφανής η αιτία τους.

Είναι σημαντικό να αναφερθεί επίσης ότι, στην αρχή της υλοποίησης, προέκυπταν προβλήματα ασυμβατότητας των μηχανισμών αυτών. Αυτό οφειλόταν στο γεγονός ότι δεν είχε καθοριστεί μια συγκεκριμένη αρχιτεκτονική λειτουργίας και μερικές φορές επιλεγόταν αυτόματα η GPU από το Docling. Για αυτό δημιουργήθηκε η `_get_docling_converter()`, η οποία ορίζει συγκεκριμένα τη συσκευή και τον τρόπο λειτουργίας για να μην υπάρχουν απρόβλεπτες επιλογές.

Πριν από την εξαγωγή εφαρμόζεται επίσης η `_looks_like_roster()`, η οποία εξετάζει τη δομή του PDF και απορρίπτει έγγραφα όταν το ποσοστό γραμμών που αντιστοιχούν σε πίνακες ή σειρές αναγνωριστικών υπερβαίνει ένα προκαθορισμένο όριο. Αυτό επιλέχθηκε σκόπιμα, επειδή υπήρξαν περιπτώσεις όπου κατά την εξαγωγή δημιουργήθηκαν πολλά διαφορετικά τμήματα κειμένου που ήταν παρόμοια μεταξύ τους, με αποτέλεσμα να δημιουργηθούν πολλά παρόμοια chunks, γεγονός που επηρέασε αρνητικά την ανάκτηση και οδήγησε σε μετρήσιμη μείωση της recall.

4.2.3 Μεταδεδομένα και χαρακτηρισμός

Κάθε αρχείο που συλλέγεται στο σώμα κειμένων συνοδεύεται από βασικά μεταδεδομένα που δηλώνουν στοιχεία της ταυτότητάς του, όπως `TITLE`, `DOC_TYPE`, `URL`, `IS_HISTORICAL`. Τα στοιχεία αυτά δηλώνονται κατά τη συλλογή, καθώς ο collector διαθέτει ήδη πληροφορίες για την προέλευση και το πλαίσιο του εγγράφου. Στη συνέχεια, διαβάζονται από την `parse_txt_file()`, διαχωρίζοντας τα αποθηκευμένα μεταδεδομένα από το κύριο περιεχόμενο του εγγράφου.

Παράλληλα, τα μεταδεδομένα μετατρέπονται σε μια κοινή μορφή μέσω της `normalize_payload()` του `src/ingestion/schema.py` και αποθηκεύονται στη βάση διανυσμάτων Qdrant. Αυτό συμβαίνει επειδή υπάρχουν πολλοί διαφορετικοί τύποι αρχείων και για αυτό χρειάζεται να κανονικοποιηθούν σε μια ομοιόμορφη δομή, ώστε να διατηρείται ένα ενιαίο σχήμα μεταδεδομένων ανεξάρτητα από την προέλευση του περιεχομένου.

Στο `schema.py` ορίζεται το `DOC_TYPES`, το οποίο είναι ένα σύνολο καθιερωμένων τύπων εγγράφων που χρησιμοποιούνται για τον χαρακτηρισμό του περιεχομένου. Το σύνολο αυτό έχει κυρίως τεκμηριωτικό χαρακτήρα και δεν χρησιμοποιείται από τον αγωγό για την επικύρωση ή την απόρριψη εγγράφων.

Ορισμένα έγγραφα που δεν έχουν επαρκείς πληροφορίες για το περιεχόμενό τους λαμβάνουν την τιμή `None` στα αντίστοιχα πεδία. Αυτό ευνοεί την εφαρμογή φίλτρων, όπου τα πεδία με τιμή `None` δεν ικανοποιούν συνθήκες ισότητας με κάποια συγκεκριμένη τιμή και, επομένως, μπορούν να αποκλειστούν όταν το αντίστοιχο κριτήριο απαιτεί συγκεκριμένη τιμή. Επιπλέον, αξιοποιείται το πεδίο `IS_HISTORICAL`, το οποίο συντελεί στην αναγνώριση της χρονολογίας του εγγράφου και αν έχει λήξει ή όχι. Αυτό παίζει ρόλο στο στάδιο της δημιουργίας της απάντησης, καθώς τα έγγραφα που έχουν χαρακτηριστεί ως ιστορικά λαμβάνουν χαμηλότερη βαθμολογία κατά την επανακατάταξη, χωρίς να αποκλείονται από τα αποτελέσματα. Ιδιαίτερα σημαντικό για προθεσμίες και ημερομηνίες είναι το αρχείο `deadline_extractor.py`, το οποίο εντοπίζει τις αντίστοιχες ημερομηνίες λήξης σε ανακοινώσεις που περιλαμβάνουν προθεσμία. Οι ημερομηνίες που εξάγονται αποθηκεύονται στο πεδίο `valid_to` και αξιοποιούνται σε επόμενο στάδιο της διαδικασίας ανάκτησης. Πιο συγκεκριμένα, σε ερωτήματα που αφορούν προθεσμίες, οι ανακοινώσεις των οποίων η ημερομηνία λήξης έχει παρέλθει απομακρύνονται πριν από την εφαρμογή του reranking, μειώνοντας έτσι την πιθανότητα να δοθεί λάθος ημερομηνία.

4.2.3.1 Τεμαχισμός ανά τύπο εγγράφου

Το επόμενο στάδιο μετά τη συλλογή των εγγράφων είναι ο τεμαχισμός. Σε αυτό το επίπεδο, τα έγγραφα του σώματος μετατρέπονται σε μικρότερες μονάδες, οι οποίες χρησιμοποιούνται στη συνέχεια για την ανάκτηση πληροφοριών. Επειδή συλλέγεται πληροφορία από διαφορετικές πηγές με ετερογενή δομή, ο τεμαχισμός του περιεχομένου έχει προσαρμοστεί ανάλογα με τον τύπο εγγράφου. Για παράδειγμα, μια ανακοίνωση αποτελεί συνήθως ένα σύντομο και αυτοτελές κείμενο, ενώ ένας κανονισμός ή ένας νόμος περιλαμβάνει ιεραρχικά οργανωμένες ενότητες και άρθρα.

Η επιλογή της κατάλληλης στρατηγικής τεμαχισμού πραγματοποιείται από τη `chunks_from_txt()`, η οποία λαμβάνει υπόψη τόσο το `DOC_TYPE` του εγγράφου όσο και την πηγή από την οποία προέρχεται. Με τον τρόπο αυτό, η πηγή μπορεί να χρησιμοποιηθεί για την επιλογή της κατάλληλης μεθόδου τεμαχισμού σε περίπτωση που τα μεταδεδομένα έχουν ελλιπή πεδία. Οι βασικές διαδρομές περιλαμβάνουν ειδική επεξεργασία για ανακοινώσεις, τεμαχισμό δομημένων εγγράφων Markdown και έναν γενικό αναδρομικό τεμαχισμό για τις υπόλοιπες περιπτώσεις.

Πίνακας 4.3: Στρατηγική τεμαχισμού και παράμετροι ανά τύπο εγγράφου

Τύπος Εγγράφου	Στρατηγική Τεμαχισμού	Παράμετροι (Κατώφλια / Μέγεθος)
announcement	Διατήρηση ως ενιαίο τμήμα (ολόκληρο). Διαφορετικά, αναδρομικός τεμαχισμός (recursive).	Κατώφλι: 2000 χαρακτήρες
curriculum, regulation, pages	MarkdownHeaderTextSplitter και έπειτα αναδρομικός τεμαχισμός.	1500/200
law	Διαχωρισμός με κανονική έκφραση (regex) σε Άρθρο <code>\d+</code> και έπειτα αναδρομικός.	900/100
schedule	Αυστηρά ένα τμήμα ανά ζεύγος (κωδικός μαθήματος, ομάδα/group).	—
Υπουργικές σελίδες	Αναδρομικός τεμαχισμός προσαρμοσμένος ανά πηγή.	700/80 ή 600/60
Απλό κείμενο	Αναδρομικός τεμαχισμός (γενική περίπτωση).	1200/200

Για τις ανακοινώσεις επιλέχθηκε να διατηρείται όλο το κείμενο, όταν δεν υπερβαίνει το προκαθορισμένο όριο μεγέθους. Η επιλογή αυτή επιτρέπει στο σύστημα να λαμβάνει ολόκληρο το περιεχόμενο μιας ανακοίνωσης, καθώς ο τεμαχισμός του θα δημιουργούσε νοηματικό κενό και πληροφορίες όπως ημερομηνίες ή προθεσμίες θα απομονώνονταν από το κύριο μέρος της. Ακόμη και σε περίπτωση που ένα έγγραφο χωριστεί σε επιμέρους τμήματα, διατηρεί τις επικεφαλίδες και τα κύρια στοιχεία του περιεχομένου, ώστε να διατηρείται σχετικό με την ενότητα στην οποία ανήκει και να μην αποσυνδέεται από το αρχικό θεματικό του πλαίσιο.

Ιδιαίτερη περίπτωση αποτελεί το ωρολόγιο πρόγραμμα, όπου κάθε τμήμα αντιστοιχεί σε συγκεκριμένο ζεύγος `course_code` και `group`. Έτσι, οι πληροφορίες διαφορετικών ομάδων δεν καταλήγουν στο ίδιο τμήμα, μειώνοντας την πιθανότητα σύγχυσης μεταξύ των αντίστοιχων ωρών και στοιχείων του προγράμματος.

Παράλληλα, ανεξάρτητα από τη μέθοδο τεμαχισμού που χρησιμοποιείται, εφαρμόζεται επίσης ανώτατο όριο μεγέθους στα παραγόμενα τμήματα. Όταν ένα τμήμα υπερβαίνει το συγκεκριμένο όριο, τεμαχίζεται εκ νέου. Αυτό οφείλεται στο γεγονός ότι ο `reranker` διαθέτει περιορισμένο μέγεθος εισόδου που μπορεί να υποστηρίξει και για αυτό μεγάλα τμήματα κειμένου τεμαχίζονται σε μικρότερα.

Εν κατακλείδι, έχουν υιοθετηθεί συγκεκριμένοι μηχανισμοί για την αντιμετώπιση συγκεκριμένων περιπτώσεων που προέκυψαν κατά την αξιολόγηση του συστήματος. Για ορισμένα έγγραφα-καταλόγους, η `_split_bullet_entries()` διαχωρίζει το περιεχόμενο έτσι ώστε κάθε εγγραφή να αποτελεί ξεχωριστό τμήμα, διευκολύνοντας την ανάκτηση συγκεκριμένων πληροφοριών. Επιπλέον, η `looks_like_injected_spam()` χρησιμοποιείται για τον εντοπισμό και την απόρριψη ανεπιθύμητου περιεχομένου που εμφανίζεται σε συγκεκριμένες εξωτερικές σελίδες. Η απόρριψη πραγματοποιείται πριν από την απόδοση του `chunk_index`, ώστε να διατηρείται η συνεχής αρίθμηση των τμημάτων που απαιτείται για τη διεύρυνση της ανάκτησης με γειτονικά τμήματα.

4.2.4 Ενσωμάτωση, αποθήκευση και επανευρετηρίαση

Αμέσως μετά το στάδιο του τεμαχισμού, τα παραγόμενα τμήματα μετατρέπονται σε διανυσματικές αναπαραστάσεις μέσω του `src/embeddings/embedder.py`. Η διαδικασία αυτή πραγματοποιείται από την `get_embedder()`, η οποία παρέχει ένα κοινό στιγμιότυπο του μοντέλου ενσωματώσεων. Τα διανύσματα που παράγονται από το μοντέλο είναι 1.024 διαστάσεων και κανονικοποιούνται ως προς τη L2 νόρμα, ώστε το εσωτερικό γινόμενο μεταξύ δύο διανυσμάτων να αντιστοιχεί στη συνημιτονική ομοιότητα. Επειδή απαιτείται συχνά η διαδικασία φόρτωσης πολλαπλών αντιγράφων, για αυτό το μοντέλο φορτώνεται μια φορά στη μνήμη και χρησιμοποιείται για διαφορετικούς σκοπούς, περιορίζοντας έτσι τη σπατάλη υπολογιστικών πόρων.

Επειδή η μνήμη είναι περιορισμένη και η επεξεργασία διανυσμάτων απαιτεί αρκετή μνήμη, χρησιμοποιείται η `apply_mps_cap()` του `src/utls/mps_guard.py`, η οποία είναι υπεύθυνη για τη διαχείριση της μνήμης. Η χρησιμότητά της έγκειται στο ότι επιβάλλει συγκεκριμένο όριο στη χρήση της διαθέσιμης μνήμης από το MPS και εφαρμόζεται στις διεργασίες μαζικής ενσωμάτωσης. Με τον τρόπο αυτό εξασφαλίζεται η ομαλή λειτουργία της διεργασίας κατά την επεξεργασία μεγάλου αριθμού τμημάτων, τα οποία θα μπορούσαν να δημιουργήσουν προβλήματα στην εκτέλεση. Το συγκεκριμένο όριο δεν εφαρμόζεται στη ζωντανή διαδικασία απάντησης, καθώς εκεί η διαθέσιμη μνήμη χρησιμοποιείται ήδη και από τα μοντέλα της ανάκτησης και της επανακατάταξης.

Τα διανύσματα κατά το στάδιο της εισαγωγής τους στη βάση αποκτούν ένα μοναδικό αναγνωριστικό, το οποίο παράγεται με `uuid.uuid5()` από τα `source_name`, `file_stem` και `chunk_index`. Επομένως, η επανευρετηρίαση του ίδιου περιεχομένου παράγει το ίδιο αναγνωριστικό. Ωστόσο, επειδή ένας επανατεμαχισμός μπορεί να αλλάξει τον αριθμό των παραγόμενων τμημάτων, πριν από κάθε νέα εισαγωγή χρησιμοποιείται η `_delete_existing_chunks()`, η οποία διαγράφει τα προηγούμενα τμήματα του ίδιου αρχείου μέσω συνθήκης `FieldCondition` στο `source_file`. Με αυτόν τον τρόπο, αποφεύγεται να εισαχθούν παλαιά τμήματα που δεν αντιστοιχούν πλέον στη νέα έκδοση του εγγράφου.

Η αποθήκευση των διανυσμάτων σε συνδυασμό με τα μεταδεδομένα τους πραγματοποιείται στη συλλογή `ihu_rag` του Qdrant. Η σύνδεση με τη βάση πραγματοποιείται αποκλειστικά μέσω της `get_qdrant_client()` του `src/utils/qdrant_conn.py`, ώστε η δημιουργία και η διαχείριση του client να γίνεται με ενιαίο τρόπο. Παράλληλα, για τον καλύτερο έλεγχο της βάσης, έχουν οριστεί μερικές παράμετροι όπως το `QDRANT_INDEXING_THRESHOLD` και το `QDRANT_HNSW_EF`, ώστε η συμπεριφορά της διανυσματικής βάσης να παραμείνει ελεγχόμενη καθώς αυξάνεται το μέγεθος του σώματος κειμένων.

Τέλος, η αρχιτεκτονική σχεδίαση της βάσης προϋποθέτει μια αυστηρά καθορισμένη σειρά ολοκλήρωσης για τη διαδικασία εισαγωγής, προκειμένου να αποφεύγονται ασυνεπείς ενδιάμεσες καταστάσεις. Το αρχείο τοποθετείται στον κατάλογο `_done/` μόνο μετά την επιτυχή εισαγωγή του, ενώ στη συνέχεια ενημερώνεται το `ministry_manifest.json`. Ο δείκτης έκδοσης του σώματος ενημερώνεται τελευταίος, μετά την ολοκλήρωση των προηγούμενων βημάτων. Με αυτόν τον τρόπο, σε περίπτωση διακοπής της διαδικασίας, υποστηρίζεται η επανεκκίνηση της εκτέλεσης από το σημείο στο οποίο διεκόπη, χωρίς απώλεια της προηγούμενης προόδου.

4.3 Οι ντετερμινιστικές διαδρομές

Πριν από κάθε ανάκτηση και πριν από κάθε κλήση του γλωσσικού μοντέλου, το ερώτημα περνάει από τρία επίπεδα που μπορούν να το απαντήσουν πλήρως χωρίς καμία από τις δύο λειτουργίες. Στα ερωτήματα που έχουν απλή διατύπωση και ζητούν μία άμεση πληροφορία, όπως ένα τηλέφωνο, μια ιδιότητα διδάσκοντα ή οι πιστωτικές μονάδες ενός μαθήματος, το γλωσσικό μοντέλο προσθέτει καθυστέρηση και πιθανότητα παραίσθησης χωρίς να αφαιρεί τίποτα. Για αυτό ο έλεγχος από ντετερμινιστικούς μηχανισμούς αποτρέπει το ενδεχόμενο παραίσθησης, αφού η απάντηση προέρχεται απευθείας από το σώμα του κειμένου.

4.3.1 Έτοιμες απαντήσεις και αναδιατύπωση με ιστορικό

Το πρώτο επίπεδο επεξεργασίας, το οποίο υλοποιείται μέσω του `src/chatbot/meta_intent.py`, διαχειρίζεται ερωτήματα καθοδήγησης και ταυτότητας του συστήματος. Όταν κάποιο ερώτημα αντιστοιχίζεται με έναν από τους ορισμένους κανόνες, τότε επιστρέφεται μια προκαθορισμένη απάντηση. Προηγουμένως, το ερώτημα κανονικοποιείται με την αφαίρεση τόνων και σημείων στίξης. Ωστόσο, για να επιστραφεί η έτοιμη απάντηση, απαιτείται απόλυτη ταύτιση του ερωτήματος του χρήστη με τους προκαθορισμένους κανόνες του συστήματος. Αυτό συμβαίνει γιατί, αν υπήρχε μερική αντιστοίχιση, θα οδηγούσε σε λανθασμένη απάντηση, όταν ένα μεγαλύτερο ερώτημα περιέχει μια φράση που ταιριάζει σε κάποιον γενικό κανόνα. Με αυτόν τον τρόπο τα συγκεκριμένα ερωτήματα απαντώνται χωρίς να απαιτείται κλήση γλωσσικού μοντέλου.

Επίσης, είναι σημαντική και η περίπτωση ερωτημάτων που εξαρτώνται από το ιστορικό της συνομιλίας. Για αυτές τις περιπτώσεις αξιοποιείται η `_rewrite_query_with_history()`, η οποία μετατρέπει τέτοια ερωτήματα σε αυτοτελή μορφή. Αρχικά πραγματοποιείται ντετερμινιστικός έλεγχος, ο οποίος αναγνωρίζει συντακτικά σφάλματα που χρειάζονται διορθώσεις. Όταν εντοπιστεί τέτοια περίπτωση, ο νέος όρος συγκρίνεται με τις λεκτικές μονάδες του προηγούμενου ερωτήματος μέσω της απόστασης Levenshtein, με μέγιστη αποδεκτή απόσταση $d \leq \max(2, \text{len}/3)$. Εφόσον αναγνωριστεί τέτοια περίπτωση, αντικαθίσταται ο αντίστοιχος όρος, ενώ σε αντίθετη περίπτωση εφαρμόζεται εναλλακτικός κανόνας που χρησιμοποιεί την τελευταία κατάλληλη λεκτική μονάδα του προηγούμενου ερωτήματος, εφόσον αυτή έχει μήκος τουλάχιστον τεσσάρων χαρακτήρων. Επιπλέον, για να αποφευχθούν

λανθασμένες ερμηνείες, τίθεται ο συντακτικός περιορισμός ότι το ρήμα της διόρθωσης πρέπει να βρίσκεται αυστηρά στην αρχή του ερωτήματος.

Στη συνέχεια, το ερώτημα διέρχεται από τη συνάρτηση `_needs_rewrite()`, η οποία λειτουργεί ως πύλη ελέγχου και κρίνει εάν απαιτείται περαιτέρω αναδιατύπωση. Ενεργοποιείται στοχευμένα όταν εντοπίζονται δεικτικές αντωνυμίες, ερωτηματικά ρήματα χωρίς εμφανές υποκείμενο ή αδύνατοι τύποι αντωνυμιών, αλλά αυστηρά και μόνο εάν το ερώτημα δεν υπερβαίνει τις τρεις λέξεις σε μήκος. Εφόσον δεν υπάρχει σχετικό μοτίβο ή το ερώτημα μπορεί να σταθεί αυτοτελώς, δεν πραγματοποιείται κάποια αλλαγή και αποφεύγεται η περιττή κλήση του γλωσσικού μοντέλου. Μόνο όταν απαιτείται αξιοποίηση του ιστορικού και οι προηγούμενοι ντετερμινιστικοί έλεγχοι δεν επαρκούν, προχωρά η διαδικασία στο μοντέλο.

Το τελευταίο στάδιο, εφόσον εξαντληθούν οι προηγούμενοι μηχανισμοί, περιλαμβάνει την κλήση του γλωσσικού μοντέλου (LLM), το οποίο τροφοδοτείται με τις δύο τελευταίες σειρές (turns) της συνομιλίας. Οι παράμετροι σε αυτή την κλήση έχουν συγκεκριμένες τιμές, καθώς χρειάζεται να επιτευχθεί ο μέγιστος ντετερμινισμός, δηλαδή το όριο μήκους τίθεται στα 60 tokens (`max_tokens=60`) και η θερμοκρασία μηδενίζεται (`temperature=0`). Όταν τελειώσει η παραγωγή της αναδιατύπωσης, ακολουθεί έλεγχος ορθολογικότητας (sanity check), στον οποίο, εάν η έξοδος του μοντέλου είναι κενή ή το μήκος της υπερβαίνει το τριπλάσιο του αρχικού ερωτήματος, η αναδιατύπωση απορρίπτεται ως μη ασφαλής και το σύστημα προωθεί το αρχικό ερώτημα του χρήστη. Ακολουθώντας την ίδια λογική με την Ενότητα 4.3.9 (ορθογραφική επιδιόρθωση), το γλωσσικό μοντέλο περιορίζεται στον ρόλο της ύστατης λύσης. Αξιοποιείται αποκλειστικά στις περιπτώσεις όπου οι ελαφρύτεροι ντετερμινιστικοί μηχανισμοί δεν επαρκούν.

4.3.2 Το επίπεδο γεγονότων: δομή και σειρά χειριστών

Επιπρόσθετα, έχουν υλοποιηθεί και άλλοι μηχανισμοί ντετερμινιστικού επιπέδου που αφορούν ερωτήματα που μπορούν να απαντηθούν χωρίς ανάκτηση ή χρήση γλωσσικού μοντέλου. Το `src/chatbot/fact_layer.py` βασίζεται σε έναν διατεταγμένο πίνακα τεσσάρων χειριστών (`_HANDLERS`), οι οποίοι εξετάζονται διαδοχικά. Κάθε χειριστής έχει δική του σημαία ενεργοποίησης (flag) και, όταν είναι απενεργοποιημένος, παραλείπεται χωρίς να επηρεάζονται οι υπόλοιποι.

Το σύστημα έχει οργανωθεί με τέτοιο τρόπο ώστε να λειτουργεί ως μηχανισμός προτεραιότητας. Πρώτος στη σειρά είναι ο `try_out_of_scope_answer`, ο οποίος εντοπίζει ερωτήματα που αφορούν άλλο τμήμα ή εγκατάσταση και τα απορρίπτει πριν ενεργοποιηθεί κάποιος άλλος χειριστής. Ακολουθεί ο `try_staff_status_answer`, που διαχειρίζεται ερωτήματα σχετικά με την κατάσταση μελών του προσωπικού, όπου αναγνωρίζει αν ένα μέλος του προσωπικού είναι ενεργό ή έχει αποσυρθεί. Τρίτος είναι ο `try_staff_contact_answer`, ο οποίος χρησιμοποιείται για ερωτήματα σχετικά με στοιχεία επικοινωνίας, όπως γραφείο και τηλέφωνο. Τελευταίος έλεγχος στην ακολουθία είναι ο `try_structured_answer`, ο οποίος χειρίζεται δομημένα ερωτήματα, όπως πληροφορίες ωρολογίου και αντίστοιχες αναζητήσεις αριθμητικών ή χρονικών δεδομένων.

Για την ενορχήστρωση των παραπάνω χειριστών αξιοποιείται η `try_fact_answer()`, η οποία εξετάζει την εκτέλεσή τους με τη συγκεκριμένη σειρά και τερματίζει μόλις κάποιος επιστρέψει μη κενό αποτέλεσμα. Μέσω της χρήσης του ελέγχου `if not is_enabled()`: `continue` διασφαλίζεται ότι κάθε χειριστής ελέγχεται από τη δική του, ανεξάρτητη σημαία ενεργοποίησης (feature flag). Σε περίπτωση απενεργοποίησης ενός

χειριστή, συνεχίζεται η διαδικασία χωρίς να διακοπεί η αλυσίδα αξιολόγησης. Μόλις όμως επιστραφεί το πρώτο μη κενό αποτέλεσμα, η συνθήκη τερματίζεται, ανεξάρτητα από τους επόμενους χειριστές.

Ιδιαίτερη σημασία έχει το γεγονός ότι, όταν ένας χειριστής δεν μπορεί να προσδιορίσει με επαρκή βεβαιότητα την απάντηση, επιστρέφει κενό αποτέλεσμα και επιτρέπει στο ερώτημα να συνεχίσει προς τη σημασιολογική διαδρομή. Με αυτόν τον τρόπο αποφεύγονται οι αυθαίρετες απαντήσεις χωρίς τεκμηρίωση. Τέλος, τόσο η γραμμή εντολών όσο και η διαδικτυακή εφαρμογή χρησιμοποιούν την ίδια `try_fact_answer()` και, συνεπώς, την ίδια σειρά χειριστών. Με αυτή τη σχεδιαστική επιλογή, διευκολύνεται η προσθήκη περισσότερων χειριστών χωρίς τη χρονοβόρα αλλαγή κώδικα και η ρύθμιση περιορίζεται μόνο στην προσθήκη του αντίστοιχου χειριστή στον πίνακα.

4.3.3 Η πύλη εισόδου

Η πύλη πεδίου είναι ένα σχεδιαστικό μέτρο που ανήκει στην ακολουθία ντετερμινιστικών ελέγχων και ο σκοπός της είναι να απορρίπτει ερωτήματα που δεν έχουν σχέση με το τμήμα. Η επιλογή αυτή αποτρέπει τα ερωτήματα που αναφέρονται σε άλλα τμήματα (όπως το τμήμα «Μηχανολόγων»), τα οποία σε άλλη περίπτωση θα μπορούσαν να παρακάμψουν τους ελέγχους και να φθάσουν στο γλωσσικό μοντέλο παράγοντας μια λανθασμένη απάντηση, καθώς θα αντιστοιχίζονταν με το Τμήμα Μηχανικών Πληροφορικής. Η συγκεκριμένη αστοχία είναι δύσκολο να εντοπιστεί με απλούς ελέγχους ορθότητας, καθώς η απάντηση μπορεί να είναι μορφολογικά ορθή και να συνοδεύεται από πηγή, ενώ παράλληλα να αναφέρεται σε διαφορετικό υποκείμενο. Για αυτόν τον λόγο, ο έλεγχος εμβέλειας προηγείται όλων των ελέγχων ντετερμινιστικής απάντησης.

[Λ8 — `structured_path.py:777-796`] Παράθεμα 4.8 — Η πύλη πεδίου: σειρά ελέγχων και βέτο ιδιοκτησίας (`src/chatbot/structured_path.py`).

Η διαδικασία αυτή διακρίνεται σε τέσσερις διαδοχικούς ελέγχους και υλοποιείται στη `out_of_scope_reason()` του `src/chatbot/structured_path.py`.

- Αφαίρεση τόνων: Αποτελεί το πρώτο βήμα και αφορά την αφαίρεση των τονικών σημαδιών, ώστε όταν συγκρίνεται μια λέξη να μην αποτυγχάνει λόγω τονισμού. Ο συγκεκριμένος μηχανισμός δεν εφαρμόζει μεταγραφή από λατινικούς σε ελληνικούς χαρακτήρες.
- Επιβεβαίωση του τμήματος: Ο θετικός έλεγχος ερωτήματος που αναφέρεται στο Τμήμα Μηχανικών Πληροφορικής σταματάει αμέσως τη διαδικασία, καθώς είναι ένα ερώτημα που είναι σχετικό με το τμήμα. Το συγκεκριμένο βήμα προηγείται όλων των υπόλοιπων ελέγχων.
- Έλεγχος ξένων τμημάτων: Ακολουθεί έλεγχος για την αντιστοίχιση 21 εξωτερικών τμημάτων, τα οποία αναγνωρίζονται συνολικά μέσω 28 προτύπων, καθώς περιλαμβάνονται και επτά γενικές αναφορές χωρίς την πλήρη ονομασία.
- Έλεγχος πανεπιστημιούπολης: Ο τελευταίος έλεγχος αφορά αναφορές σε διαφορετική πανεπιστημιούπολη και εφαρμόζεται μόνο όταν στο ερώτημα συνυπάρχει ουσιαστικό που δηλώνει εγκατάσταση.

Ο μηχανισμός αυτός είναι καθαρά ντετερμινιστικού χαρακτήρα, καθώς δεν καλείται κανένα γλωσσικό μοντέλο και δεν προκαλείται καμία υπολογιστική καθυστέρηση. Οι γενικοί όροι («γυμνοί» τύποι) προστέθηκαν διότι οι χρήστες τείνουν να παραλείπουν την πλήρη ονομασία (π.χ. αναζητούν «Μηχανολόγων» αντί για «Τμήμα Μηχανολόγων»). Από αυτή τη λίστα εξαιρέθηκε σκόπιμα η λέξη «Μηχανικών», καθώς περιλαμβάνεται και στην ονομασία του υπό εξέταση Τμήματος, αποτρέποντας την εσφαλμένη απόρριψη έγκυρων ερωτημάτων. Τέλος, η απαίτηση παρουσίας ουσιαστικού

εγκατάστασης στον τέταρτο κανόνα αποτρέπει την άσκοπη απόρριψη ερωτημάτων που απλώς αναφέρουν μια άλλη πόλη σε διαφορετικό πλαίσιο (π.χ. «μπορώ να κάνω πρακτική στις Σέρρες;»). Τα συστημικά όρια που συνεπάγεται η συγκεκριμένη, αυστηρή διάταξη κανόνων καταγράφονται διεξοδικά στο Κεφάλαιο 5.

4.3.4 Η σημασιολογική διαδρομή

Όταν κανένας ντετερμινιστικός χειριστής δεν μπορεί να επιστρέψει απάντηση, το ερώτημα εισέρχεται στη σημασιολογική διαδρομή του συστήματος, η οποία υλοποιείται στη `IHURetriever.search()`. Η συγκεκριμένη μέθοδος αξιοποιείται τόσο κατά τη ζωντανή λειτουργία της εφαρμογής όσο και κατά τη διαδικασία αξιολόγησης, ώστε τα αποτελέσματα των μετρήσεων να αντιστοιχούν στην πραγματική λειτουργία του συστήματος.

Αρχικά, το κείμενο κανονικοποιείται προτού φτάσει στο στάδιο της δημιουργίας *embedding*. Είναι σημαντική αυτή η σειρά εκτέλεσης, καθώς μια μη κανονικοποιημένη είσοδος, όπως ένα ερώτημα που περιέχει λατινικούς χαρακτήρες, μπορεί να οδηγήσει σε διαφορετική διανυσματική αναπαράσταση και, κατ' επέκταση, σε λανθασμένη ταξινόμηση πρόθεσης. Επομένως, η διόρθωση μετά τη δημιουργία του *embedding* δεν θα μπορούσε να αποκαταστήσει το ήδη παραγόμενο διάνυσμα.

Επιπρόσθετα, τη διαχείριση της συνολικής εξυπηρέτησης του ερωτήματος αναλαμβάνει μια «καρφωμένη» δέσμη κατάστασης (*pinned state bundle*). Πιο συγκεκριμένα, αξιοποιείται η κλάση `CorpusBundle`, η οποία αποτελεί μια αμετάβλητη δομή (*immutable structure*). Η δομή αυτή εμπεριέχει το σώμα κειμένων (*corpus*) φορτωμένο στη μνήμη, το λεξιλογικό ευρετήριο, το διανυσματικό ευρετήριο γειτόνων (*neighbor index*), το λεξικό μεταγραφής, καθώς και τα δέντρα διόρθωσης που χρησιμοποιούνται για τη διόρθωση. Η δημιουργία και μεταβίβαση αυτού του σταθερού στιγμιότυπου διασφαλίζει τη δομική ακεραιότητα των δεδομένων κατά τη διέλευση του ερωτήματος.

4.3.5 Κανονικοποίηση του ερωτήματος

Η κανονικοποίηση του ερωτήματος οργανώνεται σε τέσσερις διαδοχικές βαθμίδες, οι οποίες ενεργοποιούνται μόνο όταν υπάρχουν ενδείξεις ανεπαρκούς ανάκτησης. Υπάρχουν ορισμένες παρεμβάσεις που γίνονται αυτόματα σε όλες τις ερωτήσεις ανεξάρτητα από την κατάστασή τους και άλλες που εφαρμόζονται μόνο όταν η αρχική διαδρομή δεν οδηγεί σε ικανοποιητικό αποτέλεσμα (4.3.9).

Βαθμίδα 0 - Μεταγραφή από λατινικούς χαρακτήρες

Αρχικά γίνεται επεξεργασία των δίψηφων γραμμάτων και στη συνέχεια των μεμονωμένων χαρακτήρων. Η μεταγραφή αυτή δεν είναι απλή αντιστοίχιση συμβόλων, καθώς ένα λατινικό σύμβολο μπορεί να αντιστοιχεί σε περισσότερους από έναν ελληνικούς χαρακτήρες. Για τον λόγο αυτό, η εσφαλμένη μετατροπή μπορεί να οδηγήσει σε ανύπαρκτες λέξεις. Αυτό το θέμα αντιμετωπίζεται χρησιμοποιώντας λεξικό που κατασκευάζεται από το ίδιο το σώμα κειμένων μέσω της `build_greeklish_lexicon()`, η οποία αντιστοιχίζει κάθε φωνηεντικό σκελετό με τη συχνότερη πραγματική λέξη που εμφανίζεται στο σώμα.

(βλ. Παράθεμα Z.1 στο Παράρτημα Z.)

Ωστόσο, η συγκεκριμένη ταύτιση (snapping) διέπεται από αυστηρούς περιορισμούς. Πιο συγκεκριμένα, ο μηχανισμός εφαρμόζεται σε σκελετούς μήκους τεσσάρων χαρακτήρων και άνω, γιατί οι λειτουργικές λέξεις και τα άρθρα μεταγράφονται επαρκώς μέσω της αφελούς μεθόδου και, σε διαφορετική περίπτωση, θα συγκρούονταν με συχνά εμφανιζόμενα σύντομα σύμβολα. Επιπλέον, υπάρχει ένα όριο που ισχύει μετά τη μετατροπή και είναι ότι η υποψήφια λέξη γίνεται αποδεκτή μόνο εάν το μήκος της διαφέρει από το αρχικό κατά δύο χαρακτήρες το πολύ. Παράλληλα, λεκτικές μονάδες που έχουν μορφή διεύθυνσης ιστού ή ονόματος κόμβου διατηρούνται αυτούσιες. Σε αντίθετη περίπτωση, διευθύνσεις όπως το `apps.ieu.gr` θα μεταγράφονταν, καταστρέφοντας το αναγνωριστικό πριν από τη διαδικασία ενσωμάτωσης (embedding). Επιπλέον, λατινικοί όροι που είναι γραμμένοι εξ ολοκλήρου με κεφαλαία παραμένουν αμετάβλητοι. Τέλος, οποιαδήποτε είσοδος αποτελείται αμιγώς από ελληνικούς χαρακτήρες διέρχεται από τη βαθμίδα αμετάβλητη σε επίπεδο byte.

Βαθμίδα 1 - Αποκατάσταση τόνων

Κατά το στάδιο της προεπεξεργασίας, πριν ο χρήστης στείλει κάποια ερώτηση, το σύστημα διατρέπει το σώμα κειμένων και κατασκευάζει ένα λεξικό ταχείας αναζήτησης (hash map), στο οποίο οι άτονες μορφές όλων των λέξεων λειτουργούν ως κλειδιά, έχοντας ως τιμές τις καταγεγραμμένες συχνότητες των τονισμένων παραλλαγών τους. Έτσι, όταν εκτελείται η ερώτηση, εάν η ατονική μορφή μιας λεκτικής μονάδας ταυτίζεται με ένα από τα συγκεκριμένα άτονα κλειδιά, η λέξη αναγνωρίζεται άμεσα ως ορθή και επιστρέφεται η συχνότερη τονισμένη της μορφή. Δεδομένου ότι η ταύτιση ορίζεται ως μηδενικής απόστασης (zero-distance match), η συγκεκριμένη βαθμίδα είναι δομικά αδύνατον να αντικαταστήσει μια λέξη με μια άλλη. Είναι σημαντικό να γίνει αυτό το βήμα και να βρεθεί η τονισμένη μορφή της λέξης, γιατί ο μεταγενέστερος έλεγχος λεξιλογίου είναι πολύ ευαίσθητος στους τόνους και θα μπορούσε να θεωρήσει κάθε άτονη λέξη ως εκτός λεξιλογίου (Out-Of-Vocabulary).

Βαθμίδα 2 — Διόρθωση επωνύμου με μετρικό δέντρο:

Κατά την εκκίνηση του συστήματος, κατασκευάζονται από το σώμα κειμένων δύο μετρικά δέντρα Burkhard-Keller (BK-Trees), μετρώντας την απόσταση Levenshtein: ένα εξειδικευμένο δέντρο ονομάτων αποτελούμενο από 811 λεκτικές μονάδες και ένα γενικό δέντρο με 19.644 λεκτικές μονάδες.

(βλ. Παράθεμα Z.1 στο Παράρτημα Z.)

Η διαδικασία μετατροπής κατανέμεται σε δύο διακριτές μεθόδους της ίδιας κλάσης. Στη συνάρτηση `snap_token()` υλοποιούνται οι εξής δικλείδες ασφαλείας: μια λεκτική μονάδα κάτω των τεσσάρων χαρακτήρων δεν επιδιορθώνεται, όπως και οι λεκτικές μονάδες που υπάρχουν ήδη στο λεξιλόγιο του σώματος. Από την άλλη πλευρά, στη μέθοδο `_best()` υπολογίζεται το κατώφλι αποδοχής, όπου η απόσταση κανονικοποιείται ως προς το μεγαλύτερο από τα δύο μήκη και συγκρίνεται με την παράμετρο `THRESHOLD_NAMES = 0.20`. Παράλληλα, η ακτίνα αναζήτησης προκύπτει από τη μαθηματική σχέση:

$$r = \max(1, \text{int}(\theta \cdot \text{len}) + 1) \quad (4.1)$$

Τυχόν ισοβαθμίες επιλύονται με την εφαρμογή τριών διαδοχικών κριτηρίων: (α) μεγαλύτερη συχνότητα εμφάνισης στο σώμα, (β) μικρότερη απόσταση Levenshtein, (γ) αλφαβητική σειρά. Τέλος, αν και το γενικό δέντρο υφίσταται λειτουργικά, δεν εφαρμόζεται: η διόρθωση περιορίζεται ρητά στα

επώνυμο, ενώ η αιτιολόγηση αυτής της απενεργοποίησης βασίζεται σε αυστηρά μετρικά δεδομένα και αναλύεται εκτενώς στην Ενότητα 5.χ.

4.3.6 Δρομολόγηση πρόθεσης και κατασκευή της δεξαμενής

Μετά από την ολοκλήρωση της επιδιόρθωσης, το ερώτημα κανονικοποιείται σε μορφή Unicode NFC και μετέπειτα ενσωματώνεται σε διανυσματική μορφή, καθώς ακολουθεί η ταξινόμηση της πρόθεσής του. Ο αρχιτεκτονικός συνδυασμός του συστήματος είναι φτιαγμένος με τέτοιο τρόπο ώστε το ίδιο διάνυσμα να επαναχρησιμοποιείται και από την ανάκτηση χωρίς να απαιτείται επιπλέον υπολογιστικό κόστος.

(βλ. Παράθεμα Γ.1 στο Παράρτημα Γ.)

Η μέθοδος `classify_intent()` αναλύει το ερώτημα και παράγει τρία ανεξάρτητα είδη πληροφορίας:

- **Ορθογώνια σήματα:** Υπολογίζονται πριν από την τελική απόφαση πρόθεσης και αποτελούν πληροφορίες που παρουσιάζονται ως φίλτρα. Για παράδειγμα, για την ερώτηση «Πότε έχω μάθημα Δευτέρα στο ακαδημαϊκό έτος 2025-2026;», το σύστημα θα βγάλει ότι η πρόθεση είναι `teaching_schedule`, ενώ το ορθογώνιο σήμα είναι `academic_year = 2025-2026`. Τα συγκεκριμένα σήματα δεν αλλάζουν την πρόθεση, όμως παράγουν αντίστοιχους περιορισμούς για την ανάκτηση.
- **Επίπεδο 1 — Υπερισχύον πρότυπο:** Αποτελείται από μια διατεταγμένη λίστα κανονικών εκφράσεων (regular expressions). Ο πρώτος κανόνας που παρουσιάζει ταύτιση επικρατεί και η συνάρτηση τερματίζει δίχως να προχωρήσει στο δεύτερο επίπεδο. Κάθε κανόνας μπορεί να φέρει τη δική του δικλείδα ασφαλείας (guard clause), η οποία τον ακυρώνει χωρίς να διακόψει τη σάρωση. Η διάκριση αυτή εξυπηρετεί ερωτήματα τα οποία αναφέρουν ένα θέμα (π.χ. ως πλαίσιο) χωρίς να το ζητούν άμεσα.
- **Επίπεδο 2 — Ομοιότητα προτύπων:** Εδώ υπολογίζεται η συνημιτονική ομοιότητα (cosine similarity) του ερωτήματος έναντι των μέσων διανυσμάτων (centroids) ανά πρόθεση. Το σύστημα επιστρέφει απλώς την υψηλότερη τιμή ομοιότητας, χωρίς να υπολογίζεται το κατώφλι στο παρόν στάδιο.

Συνολικά, αναγνωρίζονται δεκατρείς προθέσεις: `exam_schedule`, `teaching_schedule`, `regulation`, `announcement`, `staff`, `contact`, `scholarship`, `higher_ed_law`, `espa_internship`, `enrollment_transfer`, `housing`, `graduation_ceremony` και `general`. Από αυτές, μόνο μία χαρακτηρίζεται ως αποκλειστική: η `exam_schedule` (όπου `schedule_kind = exam`). Ο διαχωρισμός των διαφορετικών ομάδων φοιτητών δεν εξασφαλίζεται στο επίπεδο της ταξινόμησης, αλλά κατά τον τεμαχισμό των κειμένων (Ενότητα 4.2.4).

Ένα σημαντικό χαρακτηριστικό αυτού του συστήματος είναι ο τρόπος με τον οποίο φιλτράρει τα έγγραφα. Αντί να υπολογίζονται διαφορετικοί κανόνες για κάθε μηχανισμό αναζήτησης, η απόφαση του φιλτραρίσματος λαμβάνεται ακριβώς μία φορά και εφαρμόζεται ταυτόχρονα και στα δύο σκέλη:

- Στο διανυσματικό σκέλος (πυκνό) εφαρμόζεται ως φίλτρο απευθείας στη βάση δεδομένων.
- Στο λεξιλογικό (αραιό) σκέλος, επειδή δεν υποστηρίζεται το φιλτράρισμα, εφαρμόζεται εκ των υστέρων ως συνάρτηση απόρριψης (`drop_fn`).

Αυτή η ομοιότητα μεταξύ των δύο μηχανισμών είναι απαραίτητη, καθώς, αν δεν υπήρχε, η λεξιλογική αναζήτηση θα ανέσυρε ξανά τα ακατάλληλα έγγραφα που μόλις είχε απορρίψει η διανυσματική.

Βγάζοντας τον κανόνα από την ίδια γραμμή κώδικα, το σύστημα διασφαλίζει ότι οι δύο μηχανισμοί δεν θα διαφωνήσουν ποτέ.

Όσον αφορά τη διαχείριση των εγγράφων, το σύστημα ακολουθεί δύο λογικές, ανάλογα με την πρόθεση του χρήστη.

- **Αποκλειστική Πρόθεση:** Σε ερωτήσεις που αφορούν συγκεκριμένα ζητήματα, το σύστημα «κλειδώνει» την αναζήτηση και ψάχνει μόνο στα έγγραφα που αφορούν αυτό το θέμα, απορρίπτοντας οτιδήποτε άλλο.
- **Μη Αποκλειστική (Γενική) Πρόθεση:** Σε περιπτώσεις που το ερώτημα είναι πολύ γενικό, ο μηχανισμός «κρύβει» προληπτικά τα πολύ ειδικά έγγραφα (όπως τα προγράμματα εξετάσεων). Αυτό γίνεται για να μη γεμίσει η δεξαμενή των αποτελεσμάτων με άσχετες πληροφορίες που θα εκτοπίσουν τις σωστές απαντήσεις.

Σε αυτό το σημείο εφαρμόζεται ο μηχανισμός της ασύμμετρης δρομολόγησης (asymmetric routing), ο οποίος προσφέρει την απαραίτητη ευελιξία. Αν το σύστημα κρίνει ότι μια γενική ερώτηση μοιάζει αρκετά με ένα έγγραφο που έχει περιοριστεί (με βαθμολογία ομοιότητας άνω του 0.55), τότε καθιστά εκ νέου ορατά τα εν λόγω έγγραφα. Ωστόσο, δεν επιβάλλει την επιλογή τους, απλά τα καθιστά προσβάσιμα, επιτρέποντάς τους να ανταγωνιστούν ισότιμα με τα υπόλοιπα έγγραφα. Ο επανακατατάκτης (reranker) παίρνει την τελική απόφαση για το αν ταιριάζουν ή όχι, καθιστώντας το σύστημα αυστηρό στην πληροφορία που μπορεί να περάσει αλλά ταυτόχρονα ικανό να την ανακτήσει σε περίπτωση αβεβαιότητας.

Για την ανάκτηση αξιοποιούνται δύο σκέλη, δημιουργώντας μια αρχική δεξαμενή υποψηφίων μεγέθους

$$\text{candidate_k} = \max(40, 3 \cdot \text{top_k}) \quad (4.2)$$

Πίνακας 4.4: Τα δύο σκέλη της υβριδικής ανάκτησης και τα βάρη τους

Σκέλος	Μηχανισμός	Βάρος
Πυκνό (Dense)	Συνημιτονική αναζήτηση στη βάση, κατώφλι 0.15, φίλτρο στον διακομιστή	0.6
Αραιό (Sparse)	Λεξιλογικό ευρετήριο BM25 επί του σώματος σε μνήμη, φίλτρο εκ των υστέρων	0.4

Οι δύο μηχανισμοί λειτουργούν συμπληρωματικά, καθώς παρουσιάζουν διαφορετικές αδυναμίες. Κωδικοί μαθημάτων, συντομογραφίες και επώνυμα μπορούν να παρουσιάζουν πολύ ισχυρή λεκτική ταύτιση στο BM25, ενώ η αντίστοιχη διανυσματική αναπαράσταση μπορεί να βρίσκεται μακριά από το ερώτημα, καθώς μεμονωμένοι αριθμητικοί ή ονομαστικοί όροι δεν αποτελούν πάντοτε ισχυρές σημασιολογικές άγκυρες για το embedding. Αντίθετα, μια παράφραση που δεν μοιράζεται κοινές λέξεις με το σχετικό έγγραφο μπορεί να παραμείνει ουσιαστικά αόρατη στο αραιό σκέλος, αλλά να ανακτηθεί από τη σημασιολογική ομοιότητα του dense σκέλους.

(βλ. Παράθεμα Γ.2 στο Παράρτημα Γ.)

Αφού παραχθούν οι δύο λίστες υποψήφιων εγγράφων, χρειάζεται να συνενωθούν σε μια τελική αξιολογημένη δεξαμενή αποτελεσμάτων. Η τελική σύντηξη (fusion) υλοποιείται επί των θέσεων κατάταξης και όχι επί των βαθμολογιών. Ο λόγος που γίνεται αυτό είναι ότι το λεξιλογικό ευρετήριο επιστρέφει διάταξη δίχως να εκθέτει τις ακατέργαστες βαθμολογίες του (raw scores), καθιστώντας αδύνατη μια απευθείας αριθμητική συγχώνευση.

Η συνεισφορά κάθε εγγράφου στην τελική βαθμολογία υπολογίζεται μέσω του αλγορίθμου Reciprocal Rank Fusion (RRF), βάσει της σχέσης

$$\text{RRF}(d) = \sum_i \frac{w_i}{K + r_i(d) + 1}, \quad K = 60 \quad (4.3)$$

με τη σταθερά εξομάλυνσης να ορίζεται ως $K = 60$. Η συγκεκριμένη συνάρτηση προσδίδει φθίνουσα συνεισφορά ως προς τη θέση, ενώ η προσθήκη της μονάδας στον παρονομαστή αποτρέπει τη διαίρεση αποκλειστικά με το K για το έγγραφο της πρώτης θέσης. Το μοναδικό αναγνωριστικό τεμαχίου (`_qid`) λειτουργεί ως κλειδί αποδιπλασιασμού (deduplication key): ένα έγγραφο το οποίο είναι παρόν και στα δύο σκέλη αθροίζει τις ανεξάρτητες συνεισφορές του και ανέρχεται φυσικά στην τελική κατάταξη, χωρίς να απαιτείται ρητός κανόνας. Τέλος, η συνάρτηση είναι σχεδιασμένη ώστε να δέχεται N σταθμισμένες λίστες και όχι αυστηρά δύο, διασφαλίζοντας ότι μια ενδεχόμενη πολλαπλή διατύπωση του ερωτήματος (multi-query expansion) θα εξυπηρετείται ομαλά από την ίδια διαδρομή κώδικα.

4.3.7 Επανακατάταξη και πολιτική εγκυρότητας

Μετά το στάδιο της σύντηξης των αποτελεσμάτων, η διαδικασία περνάει στο στάδιο της επανακατάταξης. Ο επανακατατάκτης (BGEReranker) λειτουργεί ως διασταυρούμενος κωδικοποιητής (cross-encoder) και είναι ο μοναδικός μηχανισμός επεξεργασίας του ερωτήματος και του υποψήφιου κειμένου. Σε αντίθεση με τα προηγούμενα συστήματα ανάκτησης, τα οποία συγκρίνουν ανεξάρτητες διανυσματικές αναπαραστάσεις (bi-encoders), ο επανακατατάκτης αξιολογεί άμεσα κατά πόσο το συγκεκριμένο απόσπασμα απαντά στο υποβληθέν ερώτημα.

Σε αυτό το στάδιο βαθμολογούνται 40 υποψήφια τμήματα κειμένου, από τα οποία μόνο τα έξι θα επιλεγθούν για να δοθούν στο γλωσσικό μοντέλο. Πρέπει να σημειωθεί ότι οι μονάδες μέτρησης των τμημάτων κειμένου είναι χαρακτήρες και όχι λεκτικές μονάδες (tokens), καθώς επίσης και ότι το όριο κάθε τμήματος περικόπτεται στους 512 χαρακτήρες. Η συγκεκριμένη ασυμφωνία με το ονομαστικό παράθυρο πλαισίου (context window) του μοντέλου είναι εσκεμμένη και τεκμηριώνεται εκτενώς στην ενότητα 5.

Με βάση τον αρχιτεκτονικό σχεδιασμό του διασταυρούμενου κωδικοποιητή προκύπτει ένας θεμελιώδης περιορισμός που καθορίζει τον σχεδιασμό ολόκληρου του σταδίου. Το μοντέλο αγνοεί οποιαδήποτε βαθμολογία έχει υπολογιστεί πριν από αυτό το στάδιο, καθώς αυτή δεν έχει καμία απολύτως επίδραση στη βαθμολογία του επανακατατάκτη. Βάσει αυτού του περιορισμού, προκύπτουν δύο διακριτοί μηχανισμοί διαχείρισης της εγκυρότητας:

A. Πριν την επανακατάταξη — Αυστηρή Απόρριψη (Hard Filter):

Σε περίπτωση που στο σύνολο τμημάτων υπάρχει κάποιο τμήμα που αναφέρεται σε έγγραφο που φέρει ημερομηνία λήξης (`valid_to`) παλαιότερη της τρέχουσας, αφαιρείται οριστικά από τη δεξαμενή. Η

επιλογή αυτή κρίνεται αναγκαία, καθώς δεν θα πρέπει να υπάρχει στο σύνολο κειμένου ένα έγγραφο που έχει παρέλθει το χρονικό του περιθώριο και μπορεί να επηρεάσει τα τρέχοντα.

B. Μετά την επανακατάταξη — Βαθμολογικές Ποινές (Penalties):

Αφού ο επανακατατάκτης αποδώσει τις αρχικές του βαθμολογίες, το σύστημα εφαρμόζει έναν βρόχο προσαρμογών:

Πίνακας 4.5: Προσαρμογές της βαθμολογίας επανακατάταξης

Προσαρμογή	Τιμή	Συνθήκη Ενεργοποίησης
Ποινή ιστορικότητας	-0.5	Ιστορικό τμήμα το οποίο δεν εμπίπτει σε καθεστώς εξαίρεσης
Ποινή νομοθεσίας	-0.3	Τμήμα τύπου law σε ερώτημα διαφορετικής πρόθεσης
Ενίσχυση πρόθεσης	0.0	Ανενεργή στην τρέχουσα υλοποίηση

(βλ. Παράθεμα Γ.4 στο Παράρτημα Γ.)

Ο σκοπός των ποινών είναι να μειώσουν τη βαθμολογία ενός άσχετου τμήματος κειμένου χωρίς να το διαγράψουν από τη συλλογή, διασφαλίζοντας έτσι ότι μια απολύτως σχετική απάντηση δεν μπορεί να χαθεί κατά λάθος εξαιτίας των μεταδεδομένων.

Εξαίρεση στον κανόνα της ποινής ιστορικότητας αποτελούν οι ανακοινώσεις. Αυτό συμβαίνει γιατί οι ανακοινώσεις φέρουν εκ σχεδιασμού παλαιότερες χρονολογίες, καθώς μια περυσινή ανακοίνωση παραμένει απολύτως ορθή για το γεγονός που ανακοίνωσε. Ως εκ τούτου, εξαιρούνται από την ποινή ως προεπιλογή.

Η διαδικασία συνεχίζεται ταξινομώντας τη λίστα των τμημάτων κειμένου και στη συνέχεια επιλέγοντας τα έξι τμήματα με την υψηλότερη βαθμολογία.

4.3.8 Διαμόρφωση των συμφραζομένων

Αφού ολοκληρωθεί η επιλογή των τελικών τμημάτων, το κείμενο αναδιαμορφώνεται από τους μηχανισμούς που θα παρουσιαστούν παρακάτω. Η επεξεργασία αυτή δεν μεταβάλλει το περιεχόμενο των τμημάτων, παρά μόνο προσθέτει επιπλέον πληροφορία σχετικά με στοιχεία του κειμένου, πέρα από τα αρχικά αποτελέσματα.

- Διεύρυνση με γειτονικά τμήματα (`_expand_context()`): Ο πρώτος μηχανισμός είναι `_expand_context()`, μέσω του οποίου τα τμήματα κειμένου που προέκυψαν από το προηγούμενο στάδιο μπορούν να εμπλουτιστούν με άμεσα γειτονικά τους. Αυτό, όμως, πραγματοποιείται μόνο όταν προέρχονται από το ίδιο αρχείο κειμένου `source_file` και έχουν διαδοχικούς δείκτες `chunk_index - 1` και `chunk_index + 1`. Τα γειτονικά τμήματα ανακτώνται από το `_sibling_index` που διατηρείται στη μνήμη, χωρίς νέα πρόσβαση στον δίσκο.

Η διαδικασία διέπεται από τέσσερις κρίσιμους κανόνες:

- Αφαίρεση επικάλυψης: Επειδή τα γειτονικά τμήματα μοιράζονται 200 κοινούς χαρακτήρες κατασκευής, η συνάρτηση `_join_overlap` εντοπίζει το μεγαλύτερο κοινό επίθημα-πρόθημα στη ραφή και το αφαιρεί, εξασφαλίζοντας ομαλή ροή κειμένου.
- Διαχείριση προϋπολογισμού: Ο μέγιστος προϋπολογισμός των 14.000 χαρακτήρων κατανέμεται άπληστα (greedily) κατά σειρά κατάταξης. Εάν η προσθήκη των γειτονικών τμημάτων σε ένα έγγραφο υπερβεί το όριο, το έγγραφο απλώς διατηρεί το μεμονωμένο αρχικό του κείμενο, χωρίς ποτέ να απορρίπτεται.
- Απορρόφηση (absorption): Σε περίπτωση που δύο τμήματα έχουν κοινή περιοχή συμφραζομένων, το υψηλότερα καταταγμένο τμήμα διατηρεί το συγχωνευμένο κείμενο, ενώ το δεύτερο χαρακτηρίζεται ως απορροφημένο και παραλείπεται κατά την τελική δόμηση της προτροπής (prompt), αποτρέποντας διπλοτυπίες.
- Εξαιρέσεις: Τα τμήματα που αφορούν προγράμματα ωρολογίου εξαιρούνται ρητά, καθώς η δομή τους (ένα τμήμα ανά μάθημα και ομάδα) δεν ενέχει την έννοια της χωρικής γειτνίασης στο αρχικό έγγραφο.

2. Δυναμική συνάθροιση (`_inject_aggregation()`):

Ο μηχανισμός αυτός είναι ιδιαίτερα σημαντικός για το σύστημα. Το όριο των έξι τμημάτων που διαλέγονται από τον μηχανισμό πολλές φορές μπορεί να μην επαρκεί για να δημιουργηθεί μια απάντηση. Για αυτόν τον λόγο, έχει υιοθετηθεί ένας δυναμικός μηχανισμός συνάθροισης (dynamic aggregation mechanism), ο οποίος αναγνωρίζει την ανάγκη για περισσότερη πληροφορία. Όταν το ορθογώνιο σήμα συνάθροισης έχει ενεργοποιηθεί από το στάδιο της ταξινόμησης, οι πρώτες έξι θέσεις της λίστας παραμένουν απολύτως ταυτόσημες με την κανονική κατάταξη. Στη συνέχεια, τα υπόλοιπα επιλαχόντα τμήματα προσαρτώνται στο τέλος, μέχρι τη συμπλήρωση του ορίου `AGGREGATION_MAX_CHUNKS = 24`.

4.3.9 Κλιμάκωση σε ασθενή ανάκτηση

Όταν η υψηλότερη βαθμολογία των τμημάτων μετά την επανακατάταξη βρίσκεται σε χαμηλό επίπεδο, το σύστημα δεν απορρίπτει αμέσως τα αποτελέσματα. Ειδικότερα, πραγματοποιεί έως δύο επαναπροσπάθειες, καθεμία από τις οποίες ενεργοποιείται με διαφορετικό κατώφλι. Για να θεωρηθεί το αποτέλεσμα έγκυρο και να συνεχίσει η διαδικασία απάντησης του ερωτήματος, χρειάζεται οι νέες βαθμολογίες να βελτιώνονται αισθητά. Σε διαφορετική περίπτωση, διατηρείται η αρχική δεξαμενή, έτσι ώστε να μην υποβαθμιστεί το αποτέλεσμα.

Αναδιατύπωση σε ακαδημαϊκό ύφος (`canonical_rewrite.py`)

Η πρώτη προσπάθεια ενεργοποιείται όταν το κορυφαίο σκορ πέσει κάτω από το κατώφλι `REWRITE_TRIGGER_HI = 0.28`. Η διαδικασία αυτή στοχεύει κυρίως ερωτήματα που φέρουν έντονη προφορικότητα ή κλητική διατύπωση, αναδιατυπώνοντάς τα σε ακαδημαϊκή μορφή. Η μετατροπή αυτή εκτελείται μέσω του ίδιου μοναδιαίου στιγμιότυπου του γλωσσικού μοντέλου (Qwen3) που εξυπηρετεί και την τελική παραγωγή της απάντησης.

Ωστόσο, η μετατροπή αυτή δεν είναι απλή, καθώς υπάρχει ο κίνδυνος της απώλειας σημαντικών οντοτήτων του αρχικού ερωτήματος, όπως κωδικοί μαθημάτων, επώνυμα ή ονόματα μαθημάτων. Για τον λόγο αυτό χρησιμοποιείται ο `entity_guard.py`, ο οποίος εφαρμόζει τέσσερις ελέγχους:

- Διατήρηση ψηφίων: οι αριθμητικές μονάδες του αρχικού ερωτήματος πρέπει να παραμένουν αυτούσιες.
- Διατήρηση επωνύμων και ονομάτων μαθημάτων: οι αντίστοιχες οντότητες πρέπει να εμφανίζονται στη νέα διατύπωση ως άτονα προθέματα.
- Προστασία κεφαλαιογράμματος μονάδων: οι μονάδες που εμφανίζονται με κεφαλαίους χαρακτήρες παραμένουν αναλλοίωτες.
- Έλεγχος μήκους: το μήκος της νέας εκδοχής πρέπει να παραμένει εντός των προκαθορισμένων ορίων.

Η αποτυχία οποιουδήποτε από τους παραπάνω ελέγχους οδηγεί σε απόρριψη της αναδιατύπωσης. Ο ίδιος ελεγκτής χρησιμοποιείται και από τη γεννήτρια παραλλαγών του πλαισίου αξιολόγησης, ώστε η πολιτική διατήρησης οντοτήτων να παραμένει κοινή μεταξύ της πραγματικής λειτουργίας και της διαδικασίας αξιολόγησης.

Ορθογραφική επιδιόρθωση (`spell_rewrite.py`).

Η δεύτερη προσπάθεια ορθογραφικής επιδιόρθωσης πραγματοποιείται αφού αποτύχει η πρώτη και το αποτέλεσμα της υψηλότερης βαθμολογίας είναι μικρότερο από `RETRIEVAL_MIN_TOP_SCORE = 0.25`. Αν η ανάκτηση δεν βελτιώσει αυστηρά την αρχική βαθμολογία, η νέα δεξαμενή απορρίπτεται και η διαδικασία συνεχίζει με την ορθογραφική επιδιόρθωση. Απαγορεύεται οποιαδήποτε αλλαγή του νοήματος ή των επωνύμων, για αυτό εφαρμόζεται `temperature = 0` και όριο 64 λεκτικών μονάδων.

Η έξοδος του μοντέλου υπόκειται σε αυστηρή επικύρωση:

- Δομή και μήκος: δεν επιτρέπεται αλλαγή γραμμής, ο λόγος του μήκους της νέας εκδοχής προς την αρχική πρέπει να βρίσκεται στο διάστημα $[0,5, 2,0]$ και η διαφορά στον αριθμό των λεκτικών μονάδων δεν μπορεί να υπερβαίνει τις 3.
- Έλεγχος νέων λεκτικών μονάδων: κάθε νέα λεκτική μονάδα περιεχομένου (μήκους τουλάχιστον τεσσάρων χαρακτήρων) πρέπει να υπάρχει στο λεξιλόγιο του σώματος και ταυτόχρονα να απέχει το πολύ δύο χαρακτήρες από κάποια λέξη του αρχικού ερωτήματος.
- Διατήρηση επωνύμων: κάθε επώνυμο που έχει αναγνωριστεί στο αρχικό ερώτημα πρέπει να διατηρείται στη διορθωμένη εκδοχή.

Η νέα λέξη που θα προκύψει χρειάζεται να υπάρχει στο σώμα κειμένων, καθώς και να είναι αρκετά κοντά στη λέξη που έγραψε αρχικά ο χρήστης. Με αυτόν τον τρόπο μειώνεται η πιθανότητα να αντικατασταθεί μια λέξη με κάποια διαφορετική που απλώς υπάρχει στο σώμα. Αν δεν ικανοποιείται κάποιος από τους ελέγχους, η διόρθωση απορρίπτεται και διατηρείται το προηγούμενο ερώτημα.

Τέλος, κάθε επαναπροσπάθεια εκτελείται μόνο μία φορά και δεν μπορεί να ενεργοποιήσει νέα επαναπροσπάθεια. Έτσι, η διαδικασία παραμένει ελεγχόμενη και περιορίζεται στα προκαθορισμένα στάδια.

4.4 Αποσύνθεση σύνθετων ερωτημάτων

Μερικές φορές ένα ερώτημα του χρήστη μπορεί να έχει περισσότερες από μία ερωτήσεις στο ίδιο query, δηλαδή στην ίδια διατύπωση, όπως για παράδειγμα το ερώτημα «ποιος διδάσκει τα Δίκτυα και πότε γίνεται το εργαστήριο». Η αντιμετώπιση ολόκληρης της πρότασης ως ενός ενιαίου ερωτήματος μπορεί να οδηγήσει σε μερική ανάκτηση του ζητούμενου, καθώς η διανυσματική αναπαράσταση του

ερωτήματος συνδυάζει διαφορετικά θεματικά αντικείμενα. Για τον λόγο αυτό, υιοθετήθηκε ένας μηχανισμός εντοπισμού, αποσύνθεσης και παράλληλης ανάκτησης σύνθετων ερωτημάτων.

Η αρχιτεκτονική του μηχανισμού αποτελείται από δύο επίπεδα ελέγχου. Το πρώτο καθορίζει εάν θα ενεργοποιηθεί ο μηχανισμός αποσύνθεσης, ενώ το δεύτερο εξετάζει εάν το συγκεκριμένο ερώτημα μπορεί πράγματι να διαχωριστεί με ασφάλεια σε επιμέρους υποερωτήματα.

Πύλη σύνθετων ερωτημάτων (is_multi_hop)

Αφού προηγηθεί η αρχική επεξεργασία του κειμένου με μεταγραφή και αφαίρεση τόνων, ελέγχεται αν η πρόταση περιέχει πολλαπλά ερωτήματα μέσω της is_multi_hop(), όπου η συνάρτηση μετρά τα διακριτά ερωτηματικά λήμματα. Έχουν καθοριστεί συγκεκριμένα ερωτηματικά λήμματα, όπως «ποιος», «τι», «πώς» και «πόσος», όμως το «που» έχει εξαιρεθεί ρητά, καθώς μπορεί να χρησιμοποιηθεί και ως αναφορική αντωνυμία χωρίς να δηλώνει απαραίτητα αυτοτελές ερώτημα. Δεν απαιτείται τα ερωτήματα να βρίσκονται σε συγκεκριμένη μορφή, καθώς ο μηχανισμός μπορεί να τα εντοπίσει ακόμη και αν δεν συνδέονται μεταξύ τους με σύνδεσμο, δεν γειτνιάζουν και δεν εμφανίζονται σε συγκεκριμένη σειρά.

Για να ενεργοποιηθεί ο μηχανισμός χρειάζεται να είναι ενεργές αυτές οι δύο σημαίες:

ENABLE_AGENTIC_RAG

και (όχι AGENTIC_ADAPTIVE_GATE ή is_multi_hop(ερώτημα))

Διαχωρισμός του ερωτήματος (decompose_query)

Ο δεύτερος έλεγχος είναι πιο αυστηρός από τον πρώτο, καθώς, για να πραγματοποιηθεί πραγματική διάσπαση, απαιτείται η ύπαρξη μίας και μόνο γειτνίασης μεταξύ συνδέσμου, δηλαδή «και» ή «κι», και αμέσως επόμενου ερωτηματικού λήμματος σε ολόκληρο το ερώτημα. Έχει πολλές παραμέτρους που ακυρώνουν την αποσύνθεση, πιο συγκεκριμένα όταν:

- δεν εντοπίζεται καμία γειτνίαση συνδέσμου και ερωτηματικού ή εντοπίζονται περισσότερες από μία, γεγονός που δημιουργεί αμφισημία,
- το αριστερό ή το δεξί σκέλος είναι κενό μετά τον διαχωρισμό,
- δεν εντοπίζεται εξαγωγήμη οντότητα στο αριστερό σκέλος,
- το αντίστοιχο σκέλος καταλήγει σε κενή συμβολοσειρά μετά τον καθαρισμό.

Αν ισχύει μία από αυτές τις περιπτώσεις, η διαδικασία της αποσύνθεσης ακυρώνεται και το αρχικό ερώτημα επιστρέφεται ως μοναδική υποερώτηση, ώστε η κανονική διαδικασία ανάκτησης να συνεχιστεί χωρίς παράλληλη διάσπαση.

Προώθηση οντοτήτων

Ο απλός διαχωρισμός μιας ερώτησης σε δύο σκέλη μπορεί να οδηγήσει σε δεύτερη υποερώτηση που δεν περιέχει το αντικείμενο στο οποίο αναφέρεται. Για παράδειγμα, από το «ποιος διδάσκει τα Δίκτυα και πότε γίνεται το εργαστήριο» το δεύτερο σκέλος «πότε γίνεται το εργαστήριο» δεν προσδιορίζει το μάθημα. Για την επίλυση αυτού του προβλήματος, εφαρμόζεται μηχανισμός προώθησης οντότητας. Από το αριστερό σκέλος εξάγονται λεκτικές μονάδες οι οποίες δεν ανήκουν στο σύνολο αφαίρεσης,

όπως ερωτηματικές και λειτουργικές λέξεις ή ρήματα ερώτησης, και διαθέτουν τουλάχιστον δύο αλφαριθμητικούς χαρακτήρες. Από τις διαθέσιμες μονάδες διατηρούνται οι πέντε τελευταίες και προσαρτώνται στο δεξί σκέλος. Πριν από την προώθηση πραγματοποιείται έλεγχος αποδιπλασιασμού. Εάν η συγκεκριμένη οντότητα υπάρχει ήδη στο δεξί σκέλος, μετά από μετατροπή σε άτονη μορφή, η προσθήκη παραλείπεται. Με τον τρόπο αυτό αποφεύγεται η επανάληψη ήδη υπάρχουσας πληροφορίας στη δεύτερη υποερώτηση.

Παράλληλη ανάκτηση και συγχώνευση (parallel_retrieve)

Κάθε υποερώτηση που προκύπτει από τη διαδικασία αποσύνθεσης ακολουθεί τη δική της πλήρη και ανεξάρτητη διαδικασία ανάκτησης. Κάθε διαδρομή διαθέτει τη δική της δρομολόγηση, σύντηξη αποτελεσμάτων και επανακατάταξη. Η εκτέλεση πραγματοποιείται παράλληλα μέσω ξεχωριστών νημάτων και, έπειτα από την ολοκλήρωσή τους, τα αποτελέσματα των υποερωτημάτων συνενώνονται χωρίς να πραγματοποιείται νέα επανακατάταξη. Παράλληλα, η σειρά τους διατηρείται με βάση τη σειρά ολοκλήρωσης των αντίστοιχων νημάτων. Για την αποφυγή διπλοτύπων χρησιμοποιείται αποτύπωμα που βασίζεται στους πρώτους 100 χαρακτήρες του κειμένου.

Σχέση της πύλης με την πραγματική αποσύνθεση

Επομένως, για να γίνει η διάσπαση ενός σύνθετου ερωτήματος απαιτείται να ισχύει η συνάρτηση `is_multi_hop()`. Ωστόσο, δεν αρκεί μόνο να ισχύει η `is_multi_hop()`, καθώς χρειάζεται να ισχύει και η αυστηρότερη συνθήκη `decompose_query()`. Αν δεν ισχύει η δεύτερη συνθήκη, το ερώτημα επιστρέφεται στην αρχική του μορφή ως μοναδικό υποερώτημα και η ανάκτηση παραμένει λειτουργικά ισοδύναμη με την κανονική μη παράλληλη διαδρομή, παρά το γεγονός ότι έχει περάσει από τον επιπλέον μηχανισμό δρομολόγησης.

4.5 Παραγωγή της απάντησης

4.5.1 Η πύλη απάντησης

Αφού έχει τελειώσει η διαδικασία της ανάκτησης και της κανονικοποίησης του ερωτήματος, εφαρμόζεται ένας τελευταίος έλεγχος εγκυρότητας στα αποτελέσματα που προέκυψαν. Η προσέγγιση αυτή υπογορεύει ότι η απουσία απάντησης είναι προτιμότερη από την παροχή μιας εσφαλμένης.

Η πύλη εξετάζει δύο συνθήκες:

- Κενή δεξαμενή: κανένα τμήμα δεν παρέμεινε μετά την εφαρμογή των φίλτρων και της επανακατάταξης.
- Χαμηλή βαθμολογία: η υψηλότερη βαθμολογία της επανακατάταξης είναι μικρότερη από `RETRIEVAL_MIN_TOP_SCORE = 0,25`.

Και στις δύο περιπτώσεις το σύστημα επιστρέφει μια προκαθορισμένη αρνητική απάντηση. Παράλληλα, η αιτία της άρνησης καταγράφεται ξεχωριστά σε αρχεία logs ως `empty_pool` ή `min_top_score`, ώστε να αναλυθεί αργότερα κατά το στάδιο αξιολόγησης και των δοκιμών του συστήματος (5.1.2).

4.5.2 Δομή της προτροπής και απόδοση των συμφραζομένων

Εφόσον η πύλη απάντησης επιτρέψει τη συνέχιση της διαδικασίας, τα επιλεγμένα τμήματα εισάγονται στην είσοδο του γλωσσικού μοντέλου. Η δομή της προτροπής διαχωρίζεται σε σταθερό και μεταβλητό τμήμα, με σκοπό τη σωστή καθοδήγηση του μοντέλου και την αποτελεσματική επαναχρησιμοποίηση της προσωρινής μνήμης.

(βλ. Παράθεμα Δ.1 στο Παράρτημα Δ.)

Αρχικά, η πληροφορία που βλέπει το μοντέλο είναι σταθερή και σχετίζεται με τον ρόλο και την ταυτότητα του συστήματος, καθώς και με πέντε βασικούς κανόνες λειτουργίας (system prompt). Η υπόλοιπη είσοδος περιλαμβάνει την ημερομηνία, τα ανακτηθέντα τμήματα, το τρέχον ερώτημα και το ιστορικό της συνομιλίας. Η συγκεκριμένη σειρά δεν είναι τυχαία, καθώς είναι σημαντικό στοιχείο για την αποτελεσματική επαναχρησιμοποίηση της προσωρινής μνήμης του prompt, καθώς το κοινό πρόθεμα πρέπει να παραμένει σταθερό μεταξύ διαδοχικών κλήσεων (4.5.3).

Οι πέντε βασικοί κανόνες της προτροπής καλύπτουν διαφορετικές κατηγορίες πιθανών σφαλμάτων:

- Αποκλειστικότητα πηγών: η απάντηση πρέπει να βασίζεται αποκλειστικά στα ανακτηθέντα τμήματα και, όταν αυτά δεν επαρκούν, να επιστρέφεται η προκαθορισμένη άρνηση.
- Παραπομπές: οι πληροφορίες της απάντησης πρέπει να συνοδεύονται από αναφορά στην αντίστοιχη πηγή ή διεύθυνση.
- Μορφή απάντησης: Η απάντηση πρέπει να διατυπώνεται αποκλειστικά στην ελληνική γλώσσα, χωρίς κώδικα ή προγραμματιστικά σύμβολα.
- Προτεραιότητα πηγών: σε περιπτώσεις όπου συνυπάρχουν ανακοινώσεις, ωρολόγια και προγράμματα σπουδών, ορίζεται προτεραιότητα της ανακοίνωσης και ζητείται ρητή παρουσίαση όλων των ομάδων ενός μαθήματος.
- Ισχύουσα έναντι παλαιότερης πληροφορίας: όταν υπάρχουν αντικρουόμενες πληροφορίες, το μοντέλο καθοδηγείται να προτιμά την ισχύουσα.

Οι δύο τελευταίοι κανόνες αποτελούν οδηγίες της προτροπής του μοντέλου και όχι απόλυτες εγγυήσεις του κώδικα. Αυτό σημαίνει ότι εξαρτώνται από την τήρηση της προτροπής από το ίδιο το μοντέλο και δεν εφαρμόζονται ντετερμινιστικά όπως οι αντίστοιχοι μηχανισμοί που υλοποιούνται σε προηγούμενα στάδια.

Το ιστορικό της συνομιλίας περιορίζεται στις τέσσερις τελευταίες σειρές (turns), με μέγιστο μήκος 1000 χαρακτήρες η καθεμία. Ο περιορισμός αυτός αποτρέπει την υπερβολική κατανάλωση του διαθέσιμου πλαισίου και διατηρεί χώρο για τα ανακτηθέντα συμφραζόμενα που αποτελούν τη βασική πηγή τεκμηρίωσης της απάντησης.

Ντετερμινιστική απόδοση ωρολογίων

Για περιπτώσεις που τα ανακτημένα τμήματα αφορούν ωρολόγιο πρόγραμμα, εφαρμόζεται ειδικός μηχανισμός απόδοσης, ο οποίος χρησιμοποιεί την τυπική μορφοποίηση των συγκεκριμένων chunks. Η `render_schedule_chunks()` μετατρέπει τις πληροφορίες του ωρολογίου σε προκαθορισμένη μορφή, περιλαμβάνοντας στοιχεία όπως ημέρα, ώρα, αίθουσα, είδος και διδάσκοντα. Το παραγόμενο περιεχόμενο προτάσσεται στα συμφραζόμενα της προτροπής μαζί με τα υπόλοιπα τμήματα, ώστε το γλωσσικό μοντέλο να λαμβάνει ήδη οργανωμένες τις συγκεκριμένες πληροφορίες αντί να χρειάζεται

να ανακατασκευάσει μόνο του ώρες και αίθουσες. Η `render_schedule_chunks()` επιστρέφει κενό αποτέλεσμα όταν δεν υπάρχει κανένα τμήμα ωρολογίου στη δεξαμενή ή όταν τα διαθέσιμα τμήματα δεν περιέχουν επαρκή στοιχεία για την εξαγωγή τίτλου και γραμμών. Και στις δύο θέσεις κλήσης, το αποτέλεσμα χρησιμοποιείται μόνο όταν είναι διαθέσιμο, ώστε μια λανθασμένη ενεργοποίηση του μηχανισμού να παραμένει αδρανής και να μην επηρεάζει την υπόλοιπη διαδικασία.

4.5.3 Προσωρινή μνήμη προθέματος και ροή εξόδου

Το μοντέλο κατά το στάδιο παραγωγής παράγει μία απάντηση που είναι βασισμένη σε προκαθορισμένες ρυθμίσεις και συγκεκριμένα τμήματα κειμένων από τα προηγούμενα στάδια. Πιο συγκεκριμένα, οι παράμετροι που έχουν επιλεγεί για το μοντέλο είναι: `temperature = 0`, `repetition_penalty = 1,05`, ανώτατο όριο 2.048 λεκτικών μονάδων και απενεργοποιημένη λειτουργία συλλογιστικής (`reasoning mode`). Η επιλογή των συγκεκριμένων ρυθμίσεων του μοντέλου έγινε σκόπιμα, καθώς η απάντηση πρέπει να παράγεται μόνο από τα ανακτηθέντα τμήματα και όχι από την ελεύθερη δημιουργία περιεχομένου του γλωσσικού μοντέλου. Η αποφυγή δειγματοληψίας περιορίζει επομένως την τυχαιότητα και τον κίνδυνο παραγωγής πληροφορίας που δεν στηρίζεται στο διαθέσιμο υλικό. Παράλληλα, η απενεργοποίηση της λειτουργίας συλλογιστικής πραγματοποιείται κατά την κατασκευή της προτροπής και όχι μέσω απλής οδηγίας κειμένου.

Κατά το στάδιο παραγωγής της απάντησης, το μοντέλο επεξεργάζεται αρχικά ολόκληρη την είσοδο της προτροπής, δηλαδή τα συμφραζόμενα, το ερώτημα και τις υπόλοιπες πληροφορίες που περιλαμβάνονται σε αυτή. Η συγκεκριμένη διαδικασία ονομάζεται προπλήρωση (`prefill`). Κάθε φορά που το μοντέλο παράγει μία απάντηση, αναγκάζεται να διαβάσει το σταθερό τμήμα που αφορά τις οδηγίες του συστήματος. Για αυτό αποφασίστηκε να αποθηκεύεται την πρώτη φορά και στη συνέχεια να ανακτάται στις επόμενες κλήσεις, ώστε να μην επανυπολογίζεται. Με αυτόν τον τρόπο μειώνεται το υπολογιστικό κόστος και ο χρόνος που απαιτείται για την παραγωγή μιας απάντησης.

(βλ. Παράθεμα Δ.2 στο Παράρτημα Δ.)

Η επαναχρησιμοποίηση του προθέματος είναι ισοδύναμη με την εκ νέου επεξεργασία του, εφόσον διατηρούνται τα ίδια βάρη, οι ίδιες λεκτικές μονάδες και ο ίδιος ντετερμινιστικός τρόπος υπολογισμού. Βασική προϋπόθεση είναι το αποθηκευμένο πρόθεμα να παραμένει απολύτως ίδιο με το αντίστοιχο τμήμα της τρέχουσας προτροπής. Για τον λόγο αυτό, εφαρμόζονται δύο ξεχωριστοί έλεγχοι εγκυρότητας:

- Έλεγχος κειμένου: πραγματοποιείται σύγκριση του κατακερματισμού SHA-256 της σταθερής συμβολοσειράς πριν από την τμηματοποίηση. Με τον τρόπο αυτό εντοπίζεται οποιαδήποτε αλλαγή στο ίδιο το κείμενο του προτύπου.
- Έλεγχος λεκτικών μονάδων: οι αποθηκευμένες λεκτικές μονάδες του προθέματος πρέπει να αντιστοιχούν σε πραγματικό πρόθεμα της τρέχουσας προτροπής. Ο έλεγχος αυτός εντοπίζει περιπτώσεις όπου το κείμενο φαίνεται αμετάβλητο, αλλά έχει εισαχθεί δυναμικό περιεχόμενο στην περιοχή που θεωρείται σταθερή.

Η αποτυχία οποιουδήποτε από τους δύο ελέγχους οδηγεί σε εξαίρεση και όχι σε σιωπηλή επιστροφή σε εκ νέου υπολογισμό. Με αυτόν τον τρόπο αποτρέπεται η χρήση δυνητικά παρωχημένης ή ασύμβατης προσωρινής μνήμης.

Αν αποτύχουν και οι δύο έλεγχοι, προκαλείται εξαίρεση και όχι εκ νέου επανυπολογισμός. Αυτό διασφαλίζει ότι το σύστημα δεν θα αποτύχει από την ενδεχόμενη επαναχρησιμοποίηση παρωχημένης μνήμης.

4.5.4 Έλεγχοι μετά την παραγωγή

Αφού ολοκληρωθεί η παραγωγή της απάντησης, πραγματοποιούνται δύο ακόμα έλεγχοι ως προς τις παραπομπές και τη θεμελίωση της απάντησης.

A. Επικύρωση παραπομπών (`validate_citations`) — Επιβλητικός έλεγχος:

Οι παραπομπές είναι σημαντικό στοιχείο μιας δομημένης απάντησης και είναι σημαντικό να αναφέρεται η προέλευσή της. Για τον λόγο αυτό, το ίδιο το γλωσσικό μοντέλο αναφέρει την πηγή από την οποία προέκυψε η απάντηση, μέσω της ένδειξης [SOURCE] ή της διεύθυνσης [URL] του αντίστοιχου ανακτημένου τμήματος. Μετά την παραγωγή της απάντησης, η `validate_citations()` ελέγχει τις URLs που παρήγαγε το γλωσσικό μοντέλο. Στη συνέχεια δημιουργείται ένα σύνολο έγκυρων URLs από τα ανακτημένα τμήματα, αξιοποιώντας τόσο τα μεταδεδομένα τους όσο και τις URLs που περιέχονται στο κείμενό τους. Εάν διαπιστωθεί ότι μια πηγή της απάντησης δεν ανήκει στο συγκεκριμένο σύνολο, τότε αντικαθίσταται από την ένδειξη [πηγή μη επαληθεύσιμη].

B. Έλεγχος θεμελίωσης (`find_ungrounded_facts`) — Καταγραφικός έλεγχος:

Ο συγκεκριμένος έλεγχος αφορά γεγονότα τα οποία παρήγαγε το μοντέλο στην απάντησή του χωρίς να υφίστανται στα ανακτηθέντα συμφραζόμενα. Ο έλεγχος επικεντρώνεται σε τρεις κατηγορίες:

- Ημερομηνίες: Κανονικοποιούνται πριν από τη σύγκριση, ώστε οι αριθμητικές (π.χ. 12/05) και οι ολογράφως εκφρασμένες (π.χ. 12 Μαΐου) μορφές της ίδιας ημερομηνίας να αντιμετωπίζονται ως ισοδύναμες.
- Κωδικοί μαθημάτων: Συγκρίνονται αυτούσιοι, με ρητή εξαίρεση τους τετρανήφιους αριθμούς, οι οποίοι ερμηνεύονται και αγνοούνται ως ενδεχόμενα έτη.
- Ηλεκτρονικές διευθύνσεις (Emails): Συγκρίνονται αυτούσιες.

Αφού ολοκληρωθεί ο έλεγχος, παράγεται ένας αριθμός εμπιστοσύνης, ο οποίος δηλώνει τη θεμελίωση της πληροφορίας στα ανακτημένα συμφραζόμενα και όχι τη σημασιολογική ορθότητά της. Αυτό συμβαίνει διότι μια απάντηση μπορεί να είναι πλήρως θεμελιωμένη σε ένα ανακτημένο τμήμα και παρ' όλα αυτά να είναι λανθασμένη, εάν το ίδιο το τμήμα δεν αποτελεί τη σωστή πηγή. Η βαθμολογία αυτή χωρίζεται σε τρεις ζώνες: υψηλή, για τιμή από 0,4 και άνω, μεσαία, για τιμή από 0,1 έως μικρότερη από 0,4, και χαμηλή, για τιμή μικρότερη από 0,1. Αυτός ο μηχανισμός δεν έχει ως σκοπό να κρατήσει ή να απορρίψει την απάντηση, παρά μόνο να εκφράσει στον χρήστη ένα ποσοστό εμπιστοσύνης της απάντησης με βάση τις πηγές στις οποίες στηρίζεται.

4.6 Παράδοση: η διαδικτυακή εφαρμογή

4.6.1 Διεπαφή προγραμματισμού, ροή συμβάντων και κατάσταση συνομιλίας

Το μέρος του συστήματος που αποτελεί τη διαδικτυακή εφαρμογή (web application) υιοθετεί μια αρχιτεκτονική διαχωρισμού (decoupled architecture) και αποτελείται από δύο ανεξάρτητες διεργασίες: έναν διακομιστή υποστήριξης (backend) γραμμένο σε FastAPI (ο οποίος εκτελείται στη θύρα 8000) και

μια διεπαφή χρήστη (frontend) υλοποιημένη σε Next.js (στη θύρα 3000). Ο διακομιστής FastAPI διαθέτει μια Διεπαφή Προγραμματισμού Εφαρμογών (API) μορφής JSON, χωρίς να εξυπηρετεί στατικές σελίδες HTML. Η επικοινωνία που πραγματοποιείται μεταξύ τους ελέγχεται μέσω πολιτικής διασταυρούμενης προέλευσης (CORS), με τη λίστα των επιτρεπόμενων προελεύσεων να είναι αυστηρά περιορισμένη στις δύο τοπικές διευθύνσεις της διεπαφής, διασφαλίζοντας έτσι την απομόνωση του συστήματος.

Σημαντικό χαρακτηριστικό αυτής της εφαρμογής αποτελεί η `answer_events()` του `webapp/pipeline.py`, η οποία είναι μία γεννήτρια συμβάντων και ακολουθεί τα βασικά στάδια της διαδικασίας απάντησης όπως αυτά εκτελούνται και στη γραμμή εντολών. Η επιλογή αυτής της αρχιτεκτονικής έναντι της ασύγχρονης υλοποίησης οφείλεται στο γεγονός ότι οι κλήσεις προς τα μοντέλα είναι μπλοκάρουσες, με αποτέλεσμα ο ασύγχρονος προγραμματισμός να μην προσφέρει σημαντικό όφελος στο συγκεκριμένο στάδιο. Έτσι, διατηρείται η ίδια λογική εκτέλεσης τόσο στη γραμμή εντολών όσο και στη διαδικτυακή εφαρμογή.

Η γεννήτρια παράγει τρεις διακριτούς τύπους συμβάντων:

- `status`: Ενδείξεις τρέχοντος σταδίου για την ενημέρωση του χρήστη (π.χ. «Αναζήτηση εγγράφων...», «Σκέφτομαι...»).
- `token`: Μεμονωμένα τμήματα (λέξεις/χαρακτήρες) του παραγόμενου κειμένου, εκπεμπόμενα σε πραγματικό χρόνο.
- `done`: Το τελικό συμβάν ολοκλήρωσης, το οποίο περιέχει το πλήρες ωφέλιμο φορτίο (payload).

(βλ. Παράθεμα Δ.4 στο Παράρτημα Δ.)

Καθώς τυπώνεται το κείμενο στη διεπαφή του χρήστη, αποθηκεύεται προσωρινά λέξη-λέξη και στη συνέχεια συντίθεται όλο μαζί. Η μετάδοση πραγματοποιείται μέσω Server-Sent Events (SSE). Τα δεδομένα στέλνονται σταδιακά από τον διακομιστή σε μορφή event: και `data:`, τα οποία η εφαρμογή Next.js διαβάζει από την απόκριση. Η συγκεκριμένη προσέγγιση χρησιμοποιείται επειδή το ερώτημα αποστέλλεται ως JSON. Η εφαρμογή δεν έχει κάποιο μηχανισμό σύνδεσης. Κατά την πρώτη επίσκεψη δημιουργείται ένα τυχαίο αναγνωριστικό, το οποίο αποθηκεύεται στον περιηγητή και αποστέλλεται σε κάθε αίτημα. Το ιστορικό των συνομιλιών αποθηκεύεται σε βάση PostgreSQL σε δύο πίνακες, έναν για τις συνομιλίες και έναν για τα μηνύματα, με σχέση ένα-προς-πολλά. Η διαγραφή μίας συνομιλίας σημαίνει διαγραφή και των μηνυμάτων της. Οι πηγές και τα μεταδεδομένα αποθηκεύονται σε μορφή JSON, καθώς η βάση επιτρέπει την αναζήτηση, την ταξινόμηση κατά χρόνο και την οργανωμένη διαχείριση των συνομιλιών.

Κεφάλαιο 5ο: Μεθοδολογία και όργανα αξιολόγησης

5.1 Μεθοδολογία και όργανα αξιολόγησης

Η αξιολόγηση του συστήματος δεν βασίστηκε μόνο σε ένα ενιαίο σύνολο δοκιμών, αλλά πραγματοποιήθηκε μέσω τεσσάρων διακριτών εργαλείων αξιολόγησης (evaluation instruments), καθένα από τα οποία σχεδιάστηκε για να εξετάζει διαφορετική πτυχή της λειτουργίας του. Τα εργαλεία χρησιμοποιούν διαφορετικές μετρικές, συνδέσεις και αρχεία εισόδου, ανάλογα με το είδος της αξιολόγησης. Λειτουργούν, όμως, συμπληρωματικά, καθώς καλύπτουν διαφορετικές διαστάσεις της λειτουργίας του ίδιου συστήματος.

Ο διαχωρισμός αυτός κρίθηκε απαραίτητος, καθώς υπάρχουν πολλές διαφορετικές πτυχές του συστήματος που χρειάζονται μετρήσεις και αξιολόγηση. Ειδικότερα, αξιολογούνται χωριστά η ανάκτηση, η ευρωστία απέναντι σε διαφορετικές διατυπώσεις, η συμπεριφορά σε ερωτήματα εκτός πεδίου και η ποιότητα της τελικής απάντησης.

Παρόλο που καταγράφουν ξεχωριστά πεδία, αρκετά από τα δεδομένα τους επικαλύπτονται ως προς το περιεχόμενο. Πιο συγκεκριμένα, τα 45 πρότυπα του οργάνου Β προέρχονται από το όργανο Α, ενώ οι 791 παραλλαγές του οργάνου Δ περιλαμβάνουν ήδη τα 113 βασικά ερωτήματά του. Για τον λόγο αυτό, τα μεγέθη των τεσσάρων εργαλείων δεν αθροίζονται μεταξύ τους.

Πίνακας 5.1: Συγκριτική εικόνα των τεσσάρων οργάνων αξιολόγησης

Χαρακτηριστικό	Όργανο Α: Πύλη ανάκτησης	Όργανο Β: Ευρωστία ερωτήματος	Όργανο Γ: Εκτός εμβέλειας (OOS)	Όργανο Δ: Ευρωστία απάντησης
Αρχείο εισόδου	golden_set.jsonl	robustness_suite.jsonl	oos_suite.jsonl	variants_*_v4.jsonl
Εγγραφές	131 (12 κατηγορίες)	390 παραλλαγές / 45 πρότυπα	54	791 παραλλαγές / 113 πρότυπα
Διάρθρωση συνόλων	Καθαρή: 70 Ανάπτυξης: 33 Ελέγχου: 28	Ανάπτυξης: 214 Ελέγχου: 176	Ανάπτυξης: 30 Ελέγχου: 24	Ανάπτυξης: 71 / 497 Ελέγχου: 37 / 259
Σπόρος (Seed)	— (χειροκίνητη ανάθεση)	robust-split-v1	oos-split-v1	20260730
Στρώματα / Τύποι	—	7 είδη · 3 γεννήτριες	5 στρώματα	7 τύποι κάλυψης

Κύριες μετρικές	Recall@K, MRR, nDCG@K	Μέση / Ελάχιστη ανάκληση, Συμφωνία δρομολόγησης	Ποσοστό άρνησης, Διαρροή, Αβάσιμα γεγονότα	nFR, behavior_ok, Διαρροή
-----------------	-----------------------------	---	--	------------------------------

Οι τιμές της διαίρεσης που διακρίνονται στον πίνακα για το όργανο Δ αφορούν τα τελικά βαθμολογημένα υποσύνολα και, για τον λόγο αυτό, δεν αθροίζονται στις 791 συνολικές παραλλαγές. Οι μετασχηματισμοί που οδήγησαν στη διαμόρφωση του συνόλου παρουσιάζονται αναλυτικά στην Υποενότητα 5.7.

5.1.1 Όργανο Α — Πύλη ανάκτησης

Ο βασικός ρόλος του οργάνου Α είναι να αξιολογεί την ανάκτηση των εγγράφων και πώς αυτή απέδωσε. Για κάθε ερώτηση ελέγχει αν το ζητούμενο έγγραφο εμφανίζεται στα πρώτα οκτώ αποτελέσματα και σε ποια θέση κατατάσσεται.

Το υποσύνολο που χρησιμοποιείται ως κύρια πύλη αξιολόγησης (gate) αξιοποιείται μόνο για την καθαρή διαίρεση και αποτελείται από οκτώ κατηγορίες, συνολικά 51 εγγραφές. Από αυτές, οι 44 διαθέτουν δηλωμένο έγγραφο αλήθειας (ground truth), ενώ οι υπόλοιπες δεν διαθέτουν κάποιο έγγραφο-στόχο και η σωστή συμπεριφορά είναι η άρνηση.

Επιπλέον, πρέπει να σημειωθεί ότι η διαίρεση των δεδομένων του μηχανισμού σε test-set και development-set δεν έγινε τυχαία από ειδικό σύστημα γεννήτριας τυχαίων αριθμών, αλλά πραγματοποιήθηκε χειροκίνητα στο αρχείο δεδομένων. Αυτό το χαρακτηριστικό σημειώνεται ως περιορισμός της μεθοδολογίας.

Πέρα από τις βασικές μετρικές ανάκτησης, χρησιμοποιούνται πρόσθετοι έλεγχοι παλινδρόμησης για την παρακολούθηση συγκεκριμένων περιπτώσεων που θεωρούνται κρίσιμες. Αυτό αποσκοπεί στη διατήρηση του επιπέδου απόδοσης που έχει διαμορφωθεί, ώστε πιθανές υποβαθμίσεις να εντοπίζονται ακόμη και όταν δεν είναι εμφανείς στις συγκεντρωτικές μετρικές.

5.1.2 Όργανο Β — Ευρωστία του ερωτήματος

Το όργανο Β αξιολογεί ουσιαστικά την ανάκτηση των ερωτημάτων όταν αυτά διατυπώνονται με διαφορετικό τρόπο. Υπάρχουν 390 παραλλαγές ερωτήσεων που βασίστηκαν σε 45 πρότυπες ερωτήσεις και κατανέμονται σε επτά κατηγορίες: αναδιάταξη λέξεων, καθομιλουμένη παράφραση, τυπική παράφραση, απώλεια τόνων, μεικτό greeklish, καθαρό greeklish και ορθογραφικό λάθος.

Οι πολλαπλές εκδοχές των ερωτήσεων δημιουργήθηκαν μέσω ντετερμινιστικών μετασχηματισμών καθώς και με τα γλωσσικά μοντέλα Qwen και Kriki. Η διαίρεση των δεδομένων σε σύνολο ανάπτυξης (development-set) και σύνολο ελέγχου (test-set) έγινε με τρόπο ώστε όλες οι διαφορετικές εκδοχές μίας ερώτησης να παραμείνουν σε ένα από τα δύο σύνολα. Με αυτόν τον τρόπο αποφεύγεται η εμφάνιση ισοδύναμων ή πολύ παρόμοιων διατυπώσεων και στα δύο σύνολα.

5.1.3 Όργανο Γ — Ερωτήματα εκτός εμβέλειας

Το όργανο Γ έχει ως στόχο την αξιολόγηση του συστήματος όταν υπάρχει έλλειψη γνώσης με βάση τις διαθέσιμες πηγές. Η σωστή απάντηση σε τέτοιες περιπτώσεις είναι η κατάλληλη άρνηση. Σε αυτό το

σύνολο αξιολόγησης υπάρχουν 54 εγγραφές, οι οποίες κατανέμονται σε πέντε διαφορετικά στρώματα: γειτονικό πεδίο, πιθανή αλλά ανύπαρκτη πολιτική, χρονικά ερωτήματα, ανύπαρκτη οντότητα και γενική γνώση. Τα αποτελέσματα κάθε ομάδας είναι σκόπιμα ξεχωριστά, καθώς έτσι διακρίνεται καλύτερα ο τύπος των ερωτήσεων που δυσκολεύει το σύστημα. Επιπροσθέτως, ο διαχωρισμός των ερωτήσεων δεν σχετίζεται με τους προηγούμενους μηχανισμούς αξιολόγησης, ώστε η τοποθέτηση ενός αναγνωριστικού στο σύνολο ανάπτυξης ή ελέγχου του ενός εργαλείου να μην επηρεάζει τη διαίρεσή του στο άλλο.

5.1.4 Όργανο Δ — Ευρωστία της απάντησης

Το όργανο Δ είναι το μόνο από τους μηχανισμούς αξιολόγησης που εξετάζει το περιεχόμενο της παραγόμενης απάντησης. Αποτελείται από ένα σύνολο 113 βασικών ερωτήσεων, με κάθε ερώτηση να συνοδεύεται από επτά τύπους κάλυψης: βασική διατύπωση, παράφραση, greeklish, ορθογραφικό λάθος, ελλειπτική διατύπωση, καθομιλουμένη διατύπωση και φλύαρη διατύπωση. Από αυτές τις ερωτήσεις, οι 76 αποτελούν το σύνολο ανάπτυξης και οι 37 το σφραγισμένο σύνολο ελέγχου. Στην τελική βαθμολογία συμμετείχαν 71 ερωτήσεις, καθώς για τις υπόλοιπες δεν ήταν δυνατή η απόδοση βαθμολογίας.

Η μετρική στην οποία βασίζεται αυτός ο μηχανισμός είναι η κανονικοποιημένη ανάκληση γεγονότων (normalised fact recall, nFR). Ωστόσο, η συγκεκριμένη μετρική δεν πρέπει να ερμηνεύεται ως απλό ποσοστό ορθότητας, καθώς κανονικοποιείται ως προς ένα διαφορετικό ανώτατο όριο για κάθε ερώτημα. Παρουσιάζεται πιο αναλυτικά στην Υποενότητα 5.2 πριν από την παράθεση οποιασδήποτε τιμής.

Σε αυτό το σημείο είναι σημαντικό να σημειωθεί ότι έξι από τους επτά τύπους κάλυψης λειτουργούν ως δοκιμασίες πίεσης και δεν λειτουργούν ως αντιπροσωπευτικό δείγμα της αναμενόμενης χρήσης του συστήματος. Σκοπός αυτού του οργάνου είναι να δοκιμάσει το σύστημα σε συνθήκες εκτός της προβλεπόμενης λειτουργίας του και να εντοπίσει ενδεχόμενες αδυναμίες και όχι να εκτιμήσει μια μέση εμπειρία χρήστη. Για τον λόγο αυτό, η βασική διατύπωση παρουσιάζεται και ερμηνεύεται ξεχωριστά στα αποτελέσματα.

5.1.5 Καθεστώς των σφραγισμένων συνόλων ελέγχου

Η αξία ενός σφραγισμένου συνόλου ελέγχου (sealed held-out set) εξαρτάται από το αν έχει παραμείνει ανεξάρτητο από τη διαδικασία ανάπτυξης. Ένα σύνολο ελέγχου θεωρείται σφραγισμένο μόνο όταν τα αποτελέσματά του δεν έχουν χρησιμοποιηθεί προηγουμένως για την επιλογή ή προσαρμογή του συστήματος. Στην περίπτωση αυτού του έργου, εξετάστηκε το ιστορικό των εκτελέσεων των τεσσάρων οργάνων αξιολόγησης και καταγράφηκε ο βαθμός στον οποίο τα αντίστοιχα σύνολα ελέγχου είχαν χρησιμοποιηθεί κατά την ανάπτυξη. Ο Πίνακας 5.2 συνοψίζει το καθεστώς τους.

Πίνακας 5.2: Καθεστώς των σφραγισμένων συνόλων ελέγχου

Όργανο	Μέγεθος συνόλου ελέγχου	Αναγνώσεις	Καθεστώς
A - Πύλη ανάκτησης	28	36	Χρησιμοποιήθηκε επανειλημμένα κατά την ανάπτυξη
B - Ευρωστία ερωτήματος	176	≥4	Χρησιμοποιήθηκε τουλάχιστον μία φορά για λήψη απόφασης

Γ - Εκτός εμβέλειας	24	1	Ανοίχθηκε μία φορά μετά την ολοκλήρωση της ανάπτυξης
Δ - Ευρωστία απάντησης	37 πρότυπα / 259 παραλλαγές	1	Ανοίχθηκε μία φορά μετά την ολοκλήρωση της ανάπτυξης

Το σύνολο ελέγχου του οργάνου Α χρησιμοποιήθηκε επανειλημμένα κατά τη διάρκεια της ανάπτυξης και, επομένως, δεν μπορεί να θεωρηθεί ανεξάρτητο held-out σύνολο. Αντίστοιχα, και το όργανο Β αξιοποιήθηκε σε προηγούμενες εκτελέσεις και τουλάχιστον μία από αυτές συνέβαλε στη λήψη απόφασης. Το όργανο Δ, από την άλλη, ανοίχθηκε μόνο μία φορά μετά την ολοκλήρωση των παρεμβάσεων και αποτελεί τη βασική ανεξάρτητη μέτρηση της τελικής κατάστασης του συστήματος. Αντίστοιχα, το σφραγισμένο σύνολο του Οργάνου Γ εκτελέστηκε μία φορά μετά την ολοκλήρωση της ανάπτυξης. Μετά την εκτέλεση αυτή δεν πραγματοποιήθηκαν νέες αλλαγές στην υπό αξιολόγηση έκδοση.

Αυτό αναγνωρίζεται και στις επόμενες ενότητες και παίζει καθοριστικό ρόλο για το πώς παρουσιάζονται τα αποτελέσματα σύμφωνα με το πραγματικό επίπεδο ανεξαρτησίας τους.

5.1.6 Πειθαρχία και διαδικασία των μετρήσεων

Στη διαδικασία αξιολόγησης είναι ιδιαίτερα σημαντικό να τηρηθούν μερικοί κανόνες, ώστε τα αποτελέσματα που θα προκύψουν να είναι βάσιμα και να μπορούν να συγκριθούν σωστά μεταξύ τους. Στο συγκεκριμένο σύστημα υιοθετήθηκαν αυστηροί κανόνες, ώστε οι μετρήσεις να μην είναι παραπλανητικές. Οι κανόνες αυτοί είναι:

- Πρώτον, οι αλλαγές που γίνονταν στην εφαρμογή ήταν μεθοδικές και συνοδεύονταν από προγραμματισμένες μετρήσεις, καθώς ήταν απαραίτητο να αποτιμάται προηγουμένως η επίδραση της εκάστοτε τροποποίησης.
- Πριν από σημαντικούς πειραματικούς κύκλους καταγραφόταν η υπόθεση, ο πληθυσμός αξιολόγησης, το κριτήριο επιτυχίας και η συνθήκη τερματισμού. Στο αρχείο του έργου διατηρούνται δεκατέσσερα τέτοια προεγγεγραμμένα πρωτόκολλα, τα οποία αντιστοιχούν σε οκτώ από τους δεκαπέντε καταγεγραμμένους κύκλους του οργάνου Δ.
- Τρίτον, επιβαλλόταν ο ανθρώπινος επανέλεγχος των μετρικών καθώς και της απάντησης. Η ανάγκη αυτή προέκυψε από περιπτώσεις στις οποίες το behavior_ok αξιολογούσε θετικά λανθασμένες απαντήσεις, καθώς και από περιπτώσεις στις οποίες η ανάκληση γεγονότων παρέμενε αμετάβλητη παρότι η συμπεριφορά του συστήματος είχε βελτιωθεί ουσιαστικά.
- Μετά από κάθε τεχνική αλλαγή ακολουθούσε ένα βασικό σύνολο ελέγχων, ώστε να εντοπίζονται πιθανές παλινδρομήσεις στη λειτουργία του συστήματος.
- Αφού ανοίχθηκαν μερικά σύνολα ελέγχου (όπως αυτό του οργάνου Δ), δεν έγιναν τροποποιήσεις ακόμη και με τη διαπίστωση αδυναμιών στο σύστημα και το σύστημα παρέμεινε όπως ήταν πριν από τον έλεγχο.
- Οι αριθμητικές μετρήσεις χωρίζονται σε δύο κατηγορίες, τις τεκμηριωμένες και τις αναφερόμενες, όπου στις τεκμηριωμένες υπάρχει αρχείο αποτελεσμάτων και ημερομηνία, ενώ στις αναφερόμενες έγινε η μέτρηση, αλλά το πρόγραμμα δεν αποθήκευσε το αποτέλεσμα.

5.2 Μετρικές, ορισμοί και διαδικασία βαθμολόγησης

Οι μετρικές του κεφαλαίου εξετάζουν διαφορετικές πτυχές της λειτουργίας του συστήματος. Η πρώτη αφορά την απόδοση της ανάκτησης, ενώ οι υπόλοιπες την κάλυψη των ζητούμενων πληροφοριών και τη συμπεριφορά της παραγόμενης απάντησης. Οι τελευταίες βασίζονται σε αυτοματοποιημένους λεξιλογικούς ελέγχους και, επομένως, δεν αποτελούν από μόνες τους μέτρο πραγματολογικής ορθότητας. Για κάθε μετρική γίνεται συγκεκριμένη αναφορά στο πώς αυτή χρησιμοποιείται για να μετρήσει την απόδοση του συστήματος.

5.2.1 Μετρικές ανάκτησης [45, 46]

Η επίδοση της ανάκτησης αξιολογείται κατά κύριο λόγο με τις μετρικές Recall@8, Μέση Αντίστροφη Κατάταξη (Mean Reciprocal Rank, MRR) και Κανονικοποιημένο Σωρευτικό Κέρδος Έκπτωσης (nDCG@8), και στις τρεις περιπτώσεις επί των πρώτων οκτώ αποτελεσμάτων. Το αποτέλεσμα αντιστοιχίζεται με την αναμενόμενη πηγή μέσω του αρχείου προέλευσης (source_file) του ανακτημένου αποσπάσματος.

Ωστόσο, το Recall@8 = 1,0000 μπορεί εύκολα να ερμηνευθεί ως απόλυτη επιτυχία ανάκτησης και ενέχει τον κίνδυνο παρερμηνείας. Για αυτό κρίνεται απαραίτητο να αποσαφηνιστούν δύο ιδιαιτερότητες της υλοποίησης που οδηγούν σε αυτόν τον αριθμό.

Πρώτον, η συγκεκριμένη ανάκτηση μετράει τα τμήματα που ανακτήθηκαν και όχι τον αριθμό εγγράφων. Πιο συγκεκριμένα, υπολογίζει τμήματα (chunks) που προήλθαν από σχετικές πηγές, ανεξάρτητα από το αν αυτά αντλήθηκαν από το ίδιο έγγραφο. Αυτό σημαίνει ότι στις πρώτες οκτώ θέσεις εμφανίστηκε επαρκής αριθμός σχετικών αποτελεσμάτων, όχι ότι ανακτήθηκαν όλα τα διακριτά αρχεία αναφοράς.

Δεύτερον, η ιδανική κατάταξη που χρησιμοποιείται για την κανονικοποίηση του nDCG σχηματίζεται από τα σχετικά αποτελέσματα που πράγματι ανακτήθηκαν και όχι από το πλήρες σύνολο των αναμενόμενων εγγράφων. Αν ένα σχετικό έγγραφο απουσιάζει από τα αποτελέσματα, δεν επηρεάζει τη μετρική όπως θα συνέβαινε στη θεωρητική εκδοχή της βιβλιογραφίας. Οι δύο αυτές ιδιαιτερότητες εξετάστηκαν και με αυστηρότερους ορισμούς. Στις ίδιες 44 ερωτήσεις, με υπολογισμό σε επίπεδο διακριτού εγγράφου και σε παρονομαστή πενήντα ζευγών ερωτήματος-εγγράφου, προκύπτει Recall@8 = 0,9600 αντί 1,0000 και nDCG@8 = 0,8966 αντί 0,8923. Η διαφορά οφείλεται στο ότι ο επίσημος υπολογισμός μετρά θέσεις ανάκτησης και όχι διακριτά έγγραφα, με αποτέλεσμα δύο αντίγραφα του ίδιου εγγράφου να καλύπτουν έναν δεύτερο, μη ανακτημένο στόχο.

Το MRR, από την άλλη, λαμβάνει υπόψη τη θέση της πρώτης σχετικής εμφάνισης και διακρίνει την περίπτωση όπου η κατάλληλη πηγή βρίσκεται πρώτη από εκείνη όπου εμφανίζεται προς το τέλος των οκτώ θέσεων.

Εν κατακλείδι, αυτές οι μετρικές αξιολογούν μόνο την ποιότητα της ανάκτησης. Η υψηλή τιμή στην ανάκτηση δεν σημαίνει και ότι η απάντηση που θα παραχθεί θα είναι σωστή, ούτε και ότι το απόσπασμα που ανακτήθηκε θα αξιοποιηθεί.

5.2.2 Κανονικοποιημένη ανάκληση γεγονότων (nFR)

Η κύρια αρμοδιότητα του οργάνου Δ είναι η αξιολόγηση της τελικής απάντησης από το σύστημα. Για κάθε ερώτημα έχει οριστεί μία πρότυπη απάντηση, καθώς και ένα σύνολο συγκεκριμένων πληροφοριών ή γεγονότων-στόχων (gold_facts) που θεωρούνται απαραίτητα για την απάντησή του. Για παράδειγμα, σε ένα ερώτημα σχετικά με τα στοιχεία επικοινωνίας μιας υπηρεσίας, ως γεγονότα-στόχοι μπορούν να έχουν οριστεί η ονομασία της υπηρεσίας, ένας αριθμός τηλεφώνου και μία ηλεκτρονική διεύθυνση.

Ο βασικός στόχος αυτής της αξιολόγησης είναι η παραγόμενη απάντηση να έχει τα ίδια γεγονότα-στόχους με την πρότυπη απάντηση. Δηλαδή, εξετάζει πόσες από τις προκαθορισμένες πληροφορίες βρίσκονται μέσα στην απάντηση. Δύο απαντήσεις μπορούν να έχουν διαφορετική διατύπωση και παρ' όλα αυτά να θεωρηθούν ισοδύναμες ως προς μία πληροφορία, εφόσον περιέχουν τα απαραίτητα στοιχεία της.

Τα γεγονότα που απαιτείται να επαληθευτούν ελέγχονται από έναν λεξιλογικό αντιστοιχιστή, ο οποίος αρχικά παράγει τη μετρική fact_recall, δηλαδή το ποσοστό των γεγονότων-στόχων που θεωρεί ότι καλύπτονται από την απάντηση. Ο αντιστοιχιστής, όμως, δεν πραγματοποιεί πλήρη σημασιολογική κατανόηση του κειμένου. Ως αποτέλεσμα, ακόμη και η ίδια η πρότυπη απάντηση μπορεί σε ορισμένες περιπτώσεις να μην επιτυγχάνει fact_recall ίσο με 1. Για να ληφθεί υπόψη αυτός ο περιορισμός, η ακατέργαστη ανάκληση κανονικοποιείται ως προς τη βαθμολογία που επιτυγχάνει η ίδια η πρότυπη απάντηση στον ίδιο αντιστοιχιστή. Η τιμή αυτή χρησιμοποιείται ως ταβάνι αναφοράς (ceiling) για το συγκεκριμένο ερώτημα.

Η κανονικοποιημένη ανάκληση γεγονότων (normalized Fact Recall — nFR) ορίζεται ως:

$$\text{nFR} = \min \left(1, \frac{\text{fact_recall}(\text{answer}, F)}{\text{fact_recall}(\text{reference}, F)} \right) \quad (5.1)$$

όπου F είναι το σύνολο των γεγονότων-στόχων του ερωτήματος. Με αυτόν τον τρόπο αξιολογείται πιο δίκαια η κάλυψη των γεγονότων που έχει μια απάντηση.

Για να θεωρείται μια απάντηση ότι καλύπτει τα γεγονότα που προβλέπονται, πρέπει να έχει τουλάχιστον 60% των στοιχείων που χρησιμοποιούνται για την αναπαράστασή του. Το κατώφλι αυτό ορίστηκε από την αρχή ανάπτυξης αυτής της εργασίας και δεν προσαρμόστηκε εκ των υστέρων με βάση τις επιδόσεις του συστήματος. Τα στοιχεία που αναζητούνται εξάγονται με σταθερούς κανόνες και περιλαμβάνουν αναγνωριστικά, αριθμητικές τιμές, ελληνικές λέξεις τουλάχιστον τεσσάρων χαρακτήρων και λατινικές λέξεις τουλάχιστον τριών χαρακτήρων.

Τρεις επιπλέον κανόνες εφαρμόζονται κατά τον υπολογισμό της nFR:

- Όταν το ταβάνι είναι ίσο με μηδέν ή δεν μπορεί να υπολογιστεί, η τιμή nFR θεωρείται απροσδιόριστη και η συγκεκριμένη εγγραφή δεν συμμετέχει στον υπολογισμό του μέσου όρου.
- Η τιμή περιορίζεται στο 1,0, ώστε μία απάντηση που επιτυγχάνει υψηλότερο fact_recall από την πρότυπη να μη λαμβάνει βαθμολογία μεγαλύτερη της μονάδας.
- Η τιμή nFR στρογγυλοποιείται σε τρία δεκαδικά ψηφία για κάθε απάντηση πριν από τη συνάθροιση των αποτελεσμάτων.

Ο αριθμός που αντιστοιχίζεται στο ταβάνι δεν είναι πάντα αντιπροσωπευτικός, καθώς καθορίζεται περισσότερο από την ικανότητα του λεξιλογικού αντιστοιχιστή καθώς και από την ποιότητα των γεγονότων-στόχων. Σε παλαιότερη έκδοση παρατηρήθηκε ότι ετικέτες και η προηγούμενη έκδοση του αντιστοιχιστή έδιναν χαμηλές ή μηδενικές τιμές ακόμη και στην πρότυπη απάντηση. Για τον λόγο αυτό, έγιναν διορθώσεις των ετικετών καθώς και του μηχανισμού αντιστοίχισης που είχε πρόβλημα και, μετά από τις απαραίτητες διορθώσεις, τα ταβάνια επανυπολογίστηκαν. Επομένως, παλαιότερες τιμές, όπως ο μέσος όρος 0,6459 που είχε προκύψει με προηγούμενη έκδοση του βαθμολογητή, αποτελούν ιστορικές μετρήσεις και δεν χρησιμοποιούνται ως τελικές τιμές αναφοράς.

Πίνακας 5.3: Ταβάνια της κανονικοποιημένης ανάκλησης γεγονότων

Πληθυσμός	Πρότυπα	Μέσο ταβάνι*	Διάμεσος*	Ταβάνι = 1,0	0<ε<1	Ταβάνι = 0	Απροσδιόριστο
Σύνολο ανάπτυξης	71	0,9597	1,0000	59	1	2	9
Σφραγισμένο σύνολο ελέγχου	37	0,8431	1,0000	27	4	3	3
Σύνολο	108	0,9184	1,0000	86	5	5	12

Είναι σημαντικό να σημειωθεί ότι ο μέσος όρος και η διάμεσος υπολογίζονται μόνο στις περιπτώσεις όπου το ταβάνι είναι ορισμένο.

Από τα 113 βασικά πρότυπα του οργάνου Δ, τα 108 συμμετείχαν στους τελικούς βαθμολογημένους πληθυσμούς. Τα υπόλοιπα πέντε δεν χρησιμοποιήθηκαν στην τελική βαθμολόγηση λόγω αποκλεισμού ή προβλημάτων στις αντίστοιχες ετικέτες.

Οι τιμές του συνόλου ανάπτυξης και του σφραγισμένου συνόλου ελέγχου δεν πρέπει να συγκρίνονται απευθείας ως δύο ισοδύναμες μετρήσεις επίδοσης. Τα αποτελέσματα του συνόλου ανάπτυξης είχαν υποστεί εκτενέστερο έλεγχο και διόρθωση, για αυτό υπάρχει αυτή η διαφορά μεταξύ τους. Επιπλέον, εκείνη την περίοδο η εφαρμογή είχε αρκετές αλλαγές ως προς την αρχιτεκτονική της, οι οποίες μπορεί να δημιούργησαν αυτή την απόκλιση. Επομένως, η διαφορά μεταξύ των δύο τιμών nFR δεν συνεπάγεται πραγματική μεταβολή στην αποδοτικότητα του συστήματος ανάμεσα στα δύο σύνολα δεδομένων.

Τέλος, είναι σημαντικό να αναφερθεί ότι η μετρική nFR μετράει πόσα αναμενόμενα στοιχεία υπάρχουν στην απάντηση. Αυτό, όμως, δεν σημαίνει ότι η τελική απάντηση έχει σωστό νόημα. Μία απάντηση μπορεί να περιέχει τις αναμενόμενες λέξεις ή αριθμούς αλλά να τους χρησιμοποιεί με λανθασμένο νόημα, ενώ αντίστροφα μία σωστή πληροφορία μπορεί να διατυπωθεί με τρόπο που δεν αναγνωρίζεται από τον αντιστοιχιστή. Οι συγκεκριμένοι περιορισμοί αναλύονται στην Υποενότητα 5.2.5.

5.2.3 Ορθότητα συμπεριφοράς (behavior_ok)

Το όργανο Δ επιπλέον έχει μία ακόμη τιμή που δηλώνει την αναμενόμενη επιστροφή της απάντησης. Κάθε ερώτημα φέρει μία δηλωμένη αναμενόμενη συμπεριφορά: κανονική απάντηση (answer), άρνηση (refuse), αίτημα διευκρίνισης (clarify), διόρθωση ψευδούς προϋπόθεσης (correct_premise) ή μερική

απάντηση (partial). Η συγκεκριμένη μετρική ελέγχει αν η τελική απάντηση αντιστοιχεί στον αναμενόμενο τύπο.

Παράθεμα 5.1: run_eval_v2_1.py — βαθμολόγηση της συμπεριφοράς

```
def detect(a):
    a = norm(a); cl = any(c in a for c in CLARIFY_CUES)
    return {
        "clarify": cl,
        "refuse": (not cl) and any(c in a for c in REFUSE_CUES),
        "premise": any(c in a for c in PREMISE_CUES)
    }

def behavior_ok(r, ans, fr):
    d = detect(ans); b = r["expected_behavior"]
    if b == "answer": return not (d["refuse"] or d["clarify"])
    if b == "refuse": return d["refuse"]
    if b == "clarify": return d["clarify"]
    if b == "correct_premise": return d["premise"]
    if b == "partial":
        return (
            not leaks(ans, r["must_not_contain_rx"])) and (fr or 0) > 0
    return True
```

Δύο ιδιότητες προκύπτουν άμεσα από τον παραπάνω ορισμό και είναι απαραίτητες για την ερμηνεία των αποτελεσμάτων.

Πρώτον, η ανίχνευση επιτυγχάνεται με την απλή αναζήτηση προκαθορισμένων λεκτικών ενδείξεων και όχι με τη σημασιολογική κατανόηση της απάντησης. Επίσης, χρησιμοποιούνται διαφορετικές λίστες ενδείξεων για άρνηση, αίτημα διευκρίνισης και διόρθωση λανθασμένης προϋπόθεσης.

Δεύτερον, όταν η αναμενόμενη συμπεριφορά είναι answer, ο έλεγχος εξετάζει ουσιαστικά μόνο αν η απόκριση δεν ταξινομήθηκε ως άρνηση ή αίτημα διευκρίνισης. Δηλαδή, δεν επαληθεύεται αν οι πληροφορίες της απάντησης είναι πραγματολογικά σωστές. Αυτό το συμπέρασμα προήλθε εμπειρικά κατά τη χειροκίνητη αξιολόγηση: από τις 160 απαντήσεις που χαρακτηρίστηκαν ως σίγουρα λανθασμένες, οι 106, δηλαδή το 66,2%, εξακολουθούσαν να λαμβάνουν θετική τιμή behavior_ok.

Στην περίπτωση του partial, ο έλεγχος είναι διαφορετικός, καθώς απαιτείται η ανάκληση τουλάχιστον ενός μέρους των γεγονότων καθώς και η απουσία προκαθορισμένου απαγορευμένου περιεχομένου.

Κατά συνέπεια, η behavior_ok είναι μια μετρική που αποτελεί μέτρο της αναμενόμενης συμπεριφοράς του συστήματος και όχι της πραγματολογικής ορθότητας, θεμελίωσης ή πληρότητας της απάντησης.

5.2.4 Διαρροή

5.2.5 Το όριο του λεξιλογικού ελέγχου

Κατά την ανάπτυξη του συστήματος, καταγράφηκαν ορισμένες δομικές αδυναμίες του λεξιλογικού αντιστοιχιστή. Τα περισσότερα από αυτά αποτελούν αρχιτεκτονική αδυναμία για τον τρόπο που η μετρική αποφασίζει ότι ένα γεγονός έχει καλυφθεί.

Η μετρική nFR μπορεί να αναγνωρίζει μια σωστή πληροφορία ως μη επαρκή και να πιστώσει μία λανθασμένη πρόταση όταν τα αναμενόμενα λεκτικά στοιχεία εξακολουθούν να εμφανίζονται. Από την άλλη, το behavior_ok δεν βλέπει τη νοηματική εγκυρότητα της απάντησης, για αυτό μπορεί να αξιολογήσει θετικά μία πραγματολογικά λανθασμένη απάντηση.

Για τον λόγο αυτό, καμία από τις μετρικές της ενότητας δεν αντιμετωπίζεται ως πλήρες μέτρο πραγματολογικής ορθότητας. Η ορθότητα αξιολογήθηκε επιπλέον μέσω χειροκίνητης ανάγνωσης του πληθυσμού ανάπτυξης, τα αποτελέσματα της οποίας παρουσιάζονται στην Υποενότητα 5.4.

5.2.6 Η διαδικασία μιας μέτρησης

Πέρα από την περιγραφή των μετρικών που χρησιμοποιούνται, είναι σημαντικό να αναφερθεί πώς πραγματοποιείται η διαδικασία αξιολόγησης και πώς προσδιορίζεται η πραγματολογική ορθότητα των παραγόμενων απαντήσεων.

Μία διαδρομή ανάκτησης για την αξιολόγηση και την παραγωγή. Ο evaluator χρησιμοποιεί την ίδια διαδρομή υβριδικής ανάκτησης και κατά την παραγωγή της τελικής απάντησης. Παρόλα αυτά, η πύλη δεν εκτελεί το ντετερμινιστικό επίπεδο γεγονότων ούτε τον ειδικό μηχανισμό απόδοσης του ωρολογίου προγράμματος. Για τον λόγο αυτό, αλλαγές στους μηχανισμούς δεν μπορούν να αξιολογηθούν άμεσα μέσω των μετρικών της πύλης ανάκτησης.

Παραγωγή και βαθμολόγηση ως χωριστά βήματα. Παράλληλα, οι απαντήσεις που παράγονται κατά την αξιολόγηση αποθηκεύονται, ώστε να μπορούν να χρησιμοποιηθούν ξανά σε μεταγενέστερη βαθμολόγηση χωρίς να απαιτείται νέα παραγωγή τους. Αυτό συμβαίνει επειδή στο στάδιο της αξιολόγησης πολλές φορές χρησιμοποιούνται ερωτήσεις που έχουν ήδη χρησιμοποιηθεί. Έτσι, εξοικονομούνται υπολογιστικοί πόροι, ενώ παράλληλα είναι δυνατή η επαναξιολόγηση των ίδιων ακριβώς απαντήσεων σε περίπτωση αλλαγής του τρόπου βαθμολόγησης. Για τον λόγο αυτό, οι αποθηκευμένες απαντήσεις αξιοποιούνται για μεταγενέστερη επαναξιολόγηση του συστήματος.

Χάρη στον διαχωρισμό της παραγωγής από τη βαθμολόγηση, όταν άλλαξε ο evaluator στο σύστημα, μεταβλήθηκε και το αποτέλεσμα των μετρικών χωρίς επιπλέον αλλαγή στον κώδικα. Οι ίδιες αποθηκευμένες απαντήσεις βαθμολογήθηκαν με τον νέο αντιστοιχιστή και η τιμή nFR μεταβλήθηκε από 0,339 σε 0,370, ενώ η συμπεριφορική βαθμολογία παρέμεινε αμετάβλητη. Η μεταβολή αυτή καταγράφεται ως διόρθωση του οργάνου μέτρησης και όχι ως βελτίωση του συστήματος.

Η σφραγισμένη εκτέλεση. Κατά την τελική εκτέλεση παρήχθησαν 301 απαντήσεις χωρίς σφάλματα παραγωγής, με μέσο χρόνο 161,2 δευτερόλεπτα ανά γραμμή, διάμεσο 155,1 δευτερόλεπτα και συνολικό χρόνο περίπου 13,5 ώρες. Η διαδικασία διακόπηκε λίγο πριν ολοκληρωθεί και συνεχίστηκε από το ίδιο σημείο, αξιοποιώντας τις ήδη αποθηκευμένες απαντήσεις, χωρίς αλλαγή κώδικα ή ρυθμίσεων μεταξύ των δύο τμημάτων.

Η αρχική έκδοση του αρχείου της συγκεκριμένης εκτέλεσης περιείχε 43 βασικές ερωτήσεις, οι οποίες μαζί με τις παραλλαγές τους ανέρχονταν συνολικά σε 301. Ωστόσο, διαπιστώθηκε ότι έξι από τις 43 είχαν ήδη χρησιμοποιηθεί κατά το στάδιο της ανάπτυξης και, επομένως, δεν μπορούσαν πλέον να θεωρηθούν μέρος του σφραγισμένου συνόλου ελέγχου. Έτσι, αφαιρέθηκαν αυτές οι ερωτήσεις και

έμειναν 37 βασικές και συνολικά 259. Πριν από τη βαθμολόγηση επανελέγχθηκαν επίσης τα ταβάνια των αντίστοιχων εγγραφών, ώστε παρωχημένες τιμές να μην επηρεάσουν το τελικό αποτέλεσμα.

Ποιος κρίνει τι είναι σωστό. Καμία από τις μετρικές που παρουσιάστηκαν στην υποενότητα 5.2 δεν μπορεί από μόνη της να προσδιορίσει με ασφάλεια την πραγματολογική ορθότητα των απαντήσεων. Για τον λόγο αυτό πραγματοποιήθηκε χειροκίνητη ανάγνωση και των 497 απαντήσεων του συνόλου ανάπτυξης. Πριν από τη διαδικασία αυτή ορίστηκαν πέντε κατηγορίες: σωστή απάντηση, δικαιολογημένη άρνηση όταν η απαιτούμενη πληροφορία δεν υπήρχε στη συλλογή, αδικαιολόγητη άρνηση όταν η πληροφορία υπήρχε αλλά δεν παραδόθηκε, σίγουρα λανθασμένη απάντηση και μη αποφασίσιμη περίπτωση. Συνολικά καταγράφηκαν 217 σωστές απαντήσεις, 29 δικαιολογημένες αρνήσεις, 86 αδικαιολόγητες αρνήσεις, 160 σίγουρα λανθασμένες απαντήσεις και 5 μη αποφασίσιμες περιπτώσεις. Οι αντίστοιχες ετικέτες αποθηκεύτηκαν μαζί με τα αποτελέσματα, ώστε η ταξινόμηση να μπορεί να ελεγχθεί εκ των υστέρων.

Η διαδικασία αυτή έχει δύο βασικούς περιορισμούς. Πρώτον, οι απαντήσεις εξετάστηκαν από έναν μόνο αξιολογητή, επομένως δεν υπάρχει δεύτερη ανεξάρτητη κρίση ούτε μέτρηση συμφωνίας μεταξύ αξιολογητών. Δεύτερον, οι περισσότερες απαντήσεις είχαν παραχθεί στις αρχές Αυγούστου και δεκατρείς αναπαράχθηκαν στις 13 Αυγούστου. Σε μεταγενέστερο έλεγχο 201 περιπτώσεων, σε 50 είχε στο μεταξύ μεταβληθεί η δεξαμενή υποψηφίων χωρίς να αναπαράχθεί η αντίστοιχη απάντηση. Επομένως, η χειροκίνητη αξιολόγηση δεν αποτυπώνει με απόλυτη ακρίβεια τη μεταγενέστερη κατάσταση του συστήματος, καθώς σε ορισμένες περιπτώσεις η δεξαμενή υποψηφίων είχε στο μεταξύ μεταβληθεί. Όμως, τα αποτελέσματα αξιοποιούνται περισσότερο ως μια γενική εικόνα του συστήματος παρά ως ακριβής μέτρηση της μεταγενέστερης κατάστασής του.

Είναι σημαντικό να σημειωθεί ότι κατά το στάδιο της αξιολόγησης η ίδια ερώτηση δεν έδινε πάντα την ίδια απάντηση. Για τον λόγο αυτό υπήρχε μεταβλητότητα μεταξύ των εκτελέσεων, ενώ οι σύντομες αποκρίσεις και οι αρνήσεις παρουσίασαν σημαντικά μεγαλύτερη σταθερότητα. Η επιλογή να οριστεί αυτή η τυχαιότητα ως μετρική είναι λάθος, καθώς κατά τη διάρκεια της αξιολόγησης υπήρξαν αρκετές αλλαγές στον κώδικα, στο corpus, στις ερωτήσεις και γενικές αλλαγές που δεν θα προσδιόριζαν ακριβώς αυτή την τυχαιότητα. Οι σημαντικές μεταβολές αξιολογούνται μέσω των σταθερών μετρικών και, όπου είναι απαραίτητο, μέσω επαναληπτικών και χειροκίνητων ελέγχων. Οι περιορισμοί που σχετίζονται με τη μεταβλητότητα της παραγωγής εξετάζονται περαιτέρω στην Υποενότητα 5.7.

5.3 Απόδοση της ανάκτησης

5.3.1 Η τρέχουσα πύλη

Η τελευταία καταγεγραμμένη μέτρηση της πύλης ανάκτησης πραγματοποιήθηκε στις 14 Αυγούστου 2026, ενώ οι ίδιες τιμές διατηρήθηκαν σε δεκατέσσερις διαδοχικές εκτελέσεις, από τις 9 έως τις 14 Αυγούστου, σε συλλογή 8.677 σημείων. Ο Πίνακας 5.4 παρουσιάζει την απόδοση της ανάκτησης στις τρεις διαιρέσεις του οργάνου A.

Πίνακας 5.4: Τιμές της πύλης ανά διαίρεση του οργάνου A ($K = 8$)

Διαίρεση	Εγγραφές	Recall@8	MRR	nDCG@8
Καθαρή (πύλη)	44	1,0000	0,8877	0,8923
Φοιτητική, ανάπτυξης	33	0,9091	0,8040	0,8222
Φοιτητική, ελέγχου	28	0,9643	0,8631	0,8602

Σε αυτό το σημείο είναι σημαντικό να υπενθυμιστεί ο τρόπος υπολογισμού της μετρικής Recall@8 που παρουσιάστηκε στην Υποενότητα 5.2.1. Στη συγκεκριμένη υλοποίηση, η ανάκτηση Recall@8 πραγματοποιείται με βάση τα ανακτημένα τμήματα που σχετίζονται με τα σωστά έγγραφα και όχι με βάση τα διακριτά έγγραφα.

Στις δύο φοιτητικές διαίρεσεις παρατηρείται χαμηλότερη απόδοση. Η ανάκτηση μειώνεται κατά περίπου τέσσερις έως εννέα εκατοστιαίες μονάδες σε σχέση με την καθαρή διαίρεση, ενώ αντίστοιχη πτώση παρουσιάζεται και στο MRR. Τα αποτελέσματα δείχνουν ότι όσο πιο πρόχειρη είναι η διατύπωση της ερώτησης, τόσο περισσότερο επηρεάζεται η πιθανότητα ανάκτησης σχετικής πηγής καθώς και η θέση στην οποία αυτή εμφανίζεται. Η επίδραση της διατύπωσης εξετάζεται αναλυτικότερα με μεγαλύτερο αριθμό παραλλαγών στην Υποενότητα 5.3.3.

5.3.2 Η εξέλιξη της πύλης

Η διαδικασία της ανάκτησης πέρασε από αρκετές αλλαγές τόσο προγραμματιστικές όσο και αρχιτεκτονικές. Ο Πίνακας 5.5 αποτυπώνει τα κυριότερα καταγεγραμμένα σημεία της εξέλιξης της πύλης. Ορισμένες μεταβολές οφείλονται σε αλλαγή του ίδιου του συνόλου αξιολόγησης και όχι αποκλειστικά σε μεταβολή του συστήματος.

Πίνακας 5.5: Κύρια σημεία εξέλιξης της πύλης ανάκτησης

Ημερομηνία	Μεταβολή	Recall@8	MRR	nDCG@8	Αξιολογήσιμες
09/06	Ονομαστικός κατάλογος 388 σελίδων στο ευρετήριο	0,9886	0,9087	0,9046	63
09/06	Ενεργοποίηση ανιχνευτή καταλόγων	1,0000	0,9087	0,9045	63
18/06	Αποκατάσταση τοπικού αναλυτή και ευρύτερος συλλέκτης	1,0000	0,8871	0,8988	44
22/06	Μεταβολή χωρίς καταγεγραμμένη αιτία	1,0000	0,8871	0,8967	44
28/06	Αναδρομικός τεμαχισμός αντί σημασιολογικού	1,0000	0,8902	0,8955	44

30/06	Προσθήκη σελίδων υπουργείου και διόρθωση ετικέτας	1,0000	0,8915	0,8965	45
27/07	Εκκαθάριση 21 τμημάτων διαφημιστικού περιεχομένου	1,0000	0,8915	0,8965	45
06/08	Υποβάθμιση χωρίς αλλαγή κώδικα	0,9778	0,8569	0,8685	45
07/08	Τεμαχισμός αριθμημένων ερωταπαντήσεων	1,0000	0,8902	0,8947	45
09/08	Αφαίρεση εγγραφής από το καθαρό σύνολο και επαναπροσδιορισμός βάσης	1,0000	0,8877	0,8923	44
14/08	Τρέχουσα τιμή	1,0000	0,8877	0,8923	44

Ιδιαίτερο ενδιαφέρον παρουσιάζει η μείωση της τιμής της 6ης Αυγούστου, καθώς μεταβλήθηκε χωρίς να υπάρχει αλλαγή στον κώδικα της ανάκτησης. Αυτό οφείλεται στον αποκλεισμό μιας εγγραφής από τις πρώτες οκτώ θέσεις των αποτελεσμάτων. Η διερεύνηση έδειξε ότι η αιτία βρισκόταν στον τρόπο με τον οποίο το κείμενο δινόταν στον επανακατατάκτη.

Στη συγκεκριμένη περίπτωση, κατά την οποία η σχετική εγγραφή αποκλείστηκε από τα πρώτα οκτώ αποτελέσματα, ο reranker επεξεργαζόταν μόνο τους πρώτους 512 χαρακτήρες κάθε τμήματος. Ωστόσο, η χρήσιμη πληροφορία, η οποία αρχικά βρισκόταν στον χαρακτήρα 132, μετά την αναδημιουργία του ευρετηρίου βρέθηκε στον χαρακτήρα 1017. Ως αποτέλεσμα, η πληροφορία δεν ήταν πλέον ορατή στον reranker. Στη συνέχεια εξετάστηκε ολόκληρο το τμήμα χωρίς όριο και το αποτέλεσμα ήταν 0,968999, δηλαδή πολύ σχετικό, έναντι του 0,000042 που ήταν με το όριο των 512 χαρακτήρων.

Αυτό το πρόβλημα παρουσιάστηκε και σε άλλες εγγραφές. Πιο συγκεκριμένα, το μεγαλύτερο ποσοστό του ευρετηριασμένου υλικού, που είναι το 71,6%, υπερέβαινε τους 512 χαρακτήρες, ενώ το 46,9% του συνολικού ευρετηριασμένου κειμένου βρισκόταν πέρα από το τμήμα που μπορούσε να εξετάσει ο επανακατατάκτης. Μια επιλογή που δοκιμάστηκε ήταν η αύξηση του παραθύρου του επανακατατάκτη, όμως λόγω των δοκιμών που έγιναν τελικά απορρίφθηκε, όπως παρουσιάζεται στην Υποενότητα 5.6. Τελικά, η λύση προέκυψε από τη διόρθωση των ορίων τεμαχισμού, ώστε η σχετική πληροφορία να βρίσκεται νορίτερα μέσα στα παραγόμενα τμήματα.

Μετά την αλλαγή αυτή, το Recall@8 επανήλθε στο 1,0000, ωστόσο το nDCG@8 στις 7 Αυγούστου διαμορφώθηκε στο 0,8947, ελαφρώς χαμηλότερα από την προηγούμενη τιμή αναφοράς 0,8955. Μετά από εκτενή διερεύνηση, διαπιστώθηκε ότι οφειλόταν σε ερώτημα σχετικά με την κάρτα σίτισης, του οποίου η σχετική πηγή εμφανιζόταν μόλις στην έβδομη θέση. Η αιτία ήταν ότι ανακοινώσεις που αφορούσαν ορκωμοσία παρουσίαζαν σχετική ομοιότητα με το τμήμα της κάρτας σίτισης και προηγούνταν στην κατάταξη. Η συγκεκριμένη περίπτωση καταγράφηκε ως ανοικτή αδυναμία και δεν αντιμετωπίστηκε με τεχνητή προσαρμογή των μετρικών.

Παράλληλα, μειώθηκε ο αριθμός των αξιολογήσιμων ερωτήσεων από 45 σε 44, με αποτέλεσμα η μεταβολή αυτή να επηρεάσει τους συγκεντρωτικούς μέσους όρους. Οι νέες τιμές MRR και nDCG@8 προκύπτουν αριθμητικά από την αφαίρεση της συγκεκριμένης εγγραφής:

$$\frac{45 \cdot 0,8902 - 1,0}{44} = 0,8877 \quad (5.2)$$

και

$$\frac{45 \cdot 0,8947 - 1,0}{44} = 0,8923 \quad (5.3)$$

Επομένως, η συγκεκριμένη μεταβολή των συγκεντρωτικών τιμών δεν ερμηνεύεται ως υποβάθμιση του συστήματος, αλλά ως συνέπεια της αλλαγής του πληθυσμού αξιολόγησης.

5.3.3 Ευρωστία της ανάκτησης σε παραλλαγές του ερωτήματος

Το όργανο B ελέγχει τη σταθερότητα της ανάκτησης όταν το ίδιο ερώτημα αλλάζει διατύπωση. Το σύνολο περιλαμβάνει 390 παραλλαγές ερωτημάτων, οι οποίες έχουν δημιουργηθεί με ντετερμινιστικούς μετασχηματισμούς και με τη χρήση των δύο γλωσσικών μοντέλων που περιγράφηκαν στην Υποενότητα 5.1. Είναι ιδιαίτερης σημασίας να ειπωθεί ότι το σύνολο ελέγχου είχε ήδη εξεταστεί στο πλαίσιο προηγούμενης απόφασης ανάπτυξης και, επομένως, η συγκεκριμένη τιμή δεν αποτελεί πλήρως τυφλή μέτρηση, όπως αναφέρθηκε στην Υποενότητα 5.1.5.

Πίνακας 5.6: Απόδοση της ανάκτησης στις διαιρέσεις του οργάνου B

Διαίρεση	n	Recall	MRR	nDCG
Καθαρή (αναφορά)	45	1,0000	0,8915	0,8965
Ευρωστίας, ανάπτυξης	≈ 212	0,9579	0,8388	0,8504
Ευρωστίας, ελέγχου*	≈ 212	0,9205	0,8127	0,8072

Τα αποτελέσματα του οργάνου δείχνουν ότι η ανάκτηση διατηρεί σχετικά υψηλή ανθεκτικότητα απέναντι στις διαφορετικές διατυπώσεις, ωστόσο η τιμή δεν παραμένει αμετάβλητη. Το Recall μειώνεται από 1,0000 στην καθαρή διατύπωση σε 0,9579 στο σύνολο ανάπτυξης και 0,9205 στο σύνολο ελέγχου. Επομένως, η λεκτική διαφοροποίηση επηρεάζει την ποιότητα της ανάκτησης και τη σειρά εμφάνισής του στη λίστα.

Παράλληλα, υπάρχει μείωση και στο MRR, από 0,8915 στην καθαρή διαίρεση σε 0,8388 και 0,8127 αντίστοιχα. Αυτό υποδηλώνει ότι οι παραλλαγές στη διατύπωση επηρεάζουν άμεσα τη σειρά κατάταξης του ανακτημένου τμήματος. Συμπερασματικά, από αυτή την αξιολόγηση διαπιστώθηκε ότι η ευρωστία της ανάκτησης είναι αρκετά υψηλή, όμως επηρεάζεται από τον τρόπο με τον οποίο ο χρήστης διατυπώνει το ερώτημά του.

5.3.4 Η παράμετρος αναζήτησης του ευρετηρίου

Η διαδικασία αναζήτησης αποτελεί σημαντικό παράγοντα τόσο για την ανάκτηση των σχετικών αποτελεσμάτων όσο και για τον χρόνο εκτέλεσης. Για λόγους απόδοσης της ανάκτησης, η διανυσματική βάση άλλαξε τρόπο λειτουργίας από ενσωματωμένη σε λειτουργία εξυπηρετητή κατά το στάδιο της υλοποίησης. Στη νέα λειτουργία, καθοριστικός ήταν ο ρόλος μίας παραμέτρου που ρυθμίζει ουσιαστικά τη σχέση συμβιβασμού (trade-off) μεταξύ της ταχύτητας αναζήτησης και της ποιότητας (ακρίβειας) της ανάκτησης. Η αναφερόμενη παράμετρος είναι η ef , η οποία επηρεάζει το εύρος της αναζήτησης στον γράφο του ευρετηρίου και, κατά συνέπεια, τη σχέση μεταξύ υπολογιστικού κόστους και πιθανότητας εύρεσης των πλησιέστερων διανυσμάτων.

Για την εύρεση της κατάλληλης τιμής ακολούθησαν δοκιμές πάνω στην ίδια πύλη ανάκτησης που χρησιμοποιείται στην υπόλοιπη αξιολόγηση.

Πίνακας 5.7: Σάρωση της παραμέτρου αναζήτησης έναντι της πύλης

Ρύθμιση	Recall@8	MRR	nDCG@8
Εξαντλητική αναζήτηση (αναφορά)	1,0000	0,8915	0,8965
$ef = 128$	1,0000	0,8933	0,9022
$ef = 256$	1,0000	0,8915	0,9008
$ef = 512$	1,0000	0,8915	0,8965

Από τον πίνακα φαίνεται ότι η καλύτερη επιλογή είναι η τιμή 128, καθώς εμφανίζει ελαφρώς υψηλότερο MRR και nDCG@8. Η μεταβολή είναι αμελητέα, όμως, και η διαφορά προέρχεται από δύο μόνο ερωτήματα του συνόλου. Πιο συγκεκριμένα, η μικρή αλλαγή των τιμών μπορεί να οφειλόταν και στη φυσική μεταβλητότητα της προσεγγιστικής αναζήτησης και όχι σε σταθερά καλύτερη απόδοση.

Εν κατακλείδι, υιοθετήθηκε η τιμή 512, η οποία στη συγκεκριμένη σάρωση παρήγαγε τις ίδιες τιμές Recall@8, MRR και nDCG@8 με την εξαντλητική αναζήτηση. Ως προς τον χρόνο εκτέλεσης, η διαφορά της από την αμέσως μικρότερη τιμή που εξετάστηκε είναι μόλις 1,3%, δηλαδή αμελητέα επιβάρυνση για την ίδια ποιότητα ανάκτησης.

5.3.5 Περιορισμοί της τιμής Recall@8 = 1,0000

Η ανάκτηση των τμημάτων έχει το ισχυρότερο αποτέλεσμα μετρικής στην αξιολόγηση του συστήματος. Επιλέχθηκαν 40 τμήματα εγγράφων για την εξέταση της ανάκτησης σχετικών αποσπασμάτων στα πρώτα οκτώ αποτελέσματα. Σε 30 παραλλαγές ερωτήσεων, η πληροφορία υπήρχε κανονικά στο ευρετηριασμένο corpus, αλλά δεν εμφανίστηκε σε κανένα από τα 40 υποψήφια τμήματα. Επομένως, το πρόβλημα εντοπίστηκε στο στάδιο παραγωγής των υποψηφίων τμημάτων. Καθώς το κατάλληλο τμήμα δεν έφτανε ποτέ στον reranker, αυτός δεν μπορούσε να το επαναφέρει ή να διορθώσει την αστοχία.

5.4 Απόδοση της απάντησης και της άρνησης

5.4.1 Το σφραγισμένο αποτέλεσμα

Το σφραγισμένο σύνολο δεδομένων που φυλασσόταν για να αξιολογηθεί στο τέλος της υλοποίησης εξετάστηκε στις 17 Αυγούστου 2026. Πριν από την αξιολόγηση, η προβλεπόμενη συμπεριφορά ήταν να προκύψουν τιμές αντίστοιχες με αυτές του συνόλου ανάπτυξης.

Πίνακας 5.8: Σφραγισμένο σύνολο ελέγχου έναντι συνόλου ανάπτυξης

Μέγεθος	Σφραγισμένο	Ανάπτυξης	Πληθυσμός σφραγισμένου
Ορθότητα συμπεριφοράς	0,6216	0,6358	259 γραμμές / 37 πρότυπα
Ορθότητα συμπεριφοράς, μόνο ερωτήματα απάντησης	0,6774	0,6889	217 γραμμές
nFR	0,389	0,328	217 γραμμές / 31 πρότυπα
Διαρροές*	0	0	14 ελέγξιμες γραμμές στο σφραγισμένο

* Η τιμή αφορά τις γραμμές στις οποίες είχαν οριστεί συγκεκριμένα απαγορευμένα μοτίβα.

Η προβλεπόμενη συμπεριφορά επιβεβαιώθηκε στις δύο μετρικές συμπεριφοράς. Η μείωση της τιμής ήταν ελάχιστη, από 0,6358 σε 0,6216 και, στα ερωτήματα όπου αναμενόταν απάντηση, από 0,6889 σε 0,6774.

Αντίθετα, η μετρική nFR έχει διαφορετική συμπεριφορά στο σφραγισμένο σύνολο, καθώς οι τιμές των δύο διαιρέσεων δεν συγκρίνονται άμεσα. Όπως αναφέρθηκε στην Υποενότητα 5.2.2, το μέσο ταβάνι των ορισμένων εγγραφών είναι 0,9597 στο σύνολο ανάπτυξης και 0,8431 στο σφραγισμένο σύνολο, με αποτέλεσμα η κανονικοποίηση να πραγματοποιείται με διαφορετικούς παρονομαστές. Για αυτόν τον λόγο, η αύξηση της τιμής nFR δεν αποτελεί από μόνη της ένδειξη βελτίωσης της απόδοσης.

Η κάθε μετρική αφορά διαφορετικό σύνολο δεδομένων. Το αποτέλεσμα της συνολικής ορθότητας συμπεριφοράς απορρέει από ένα σύνολο 259 παραλλαγών ερωτήσεων, ενώ οι βασικές ερωτήσεις είναι 37. Τόσο η ορθότητα συμπεριφοράς όσο και η κανονικοποιημένη ανάκληση γεγονότων υπολογίζονται επί 217 παραλλαγών, πρόκειται όμως για δύο διαφορετικά υποσύνολα των 259, καθένα από τα οποία εξαιρεί διαφορετικές βασικές ερωτήσεις για διαφορετικό λόγο. Η σύμπτωση του πλήθους είναι τυχαία και οι δύο τιμές δεν αναφέρονται στις ίδιες γραμμές.

Τέλος, είναι σημαντικό να τονιστεί ότι ένα τμήμα των ερωτήσεων είχε αποκαλυφθεί κατά το στάδιο της ανάπτυξης και για αυτό δεν μπορούσε να συμπεριληφθεί στο σφραγισμένο σύνολο των ερωτήσεων αξιολόγησης. Ως αποτέλεσμα, αφαιρέθηκαν οι συγκεκριμένες ερωτήσεις και απέμειναν 37 πρότυπα και 259 παραλλαγές.

5.4.2 Απόδοση ανά τύπο διατύπωσης

Σε αυτή την υποενότητα αναλύεται η απόδοση του συστήματος σε διαφορετικούς τύπους ερωτήσεων. Ο Πίνακας 5.9 παρουσιάζει την ορθότητα συμπεριφοράς και το nFR για τις επτά διαφορετικές κατηγορίες διατύπωσης.

Πίνακας 5.9: Ορθότητα συμπεριφοράς και nFR ανά τύπο διατύπωσης

Τύπος	Συμπεριφορά (σφρ.)	Μόνο απάντηση	nFR (σφρ.)	Συμπεριφορά (ανάπτ.)	nFR (ανάπτ.)
Βασικός	0,7027	0,7742	0,5618	0,6620	0,3681
Ορθογραφικό λάθος	0,6216	0,6774	0,5269	0,7183	0,3903
Greeklish	0,6486	0,7097	0,4973	0,6620	0,3500
Ελλειπτικός	0,6486	0,7097	0,4705	0,7606	0,3625
Παράφραση	0,6486	0,7097	0,2742	0,5775	0,3070
Φλύαρος	0,6216	0,6774	0,2365	0,6338	0,3181
Καθομιλούμενος	0,4595	0,4839	0,1559	0,4366	0,2000
Συγκεντρωτικά	0,6216	0,6774	0,389	0,6358	0,328

Είναι σημαντικό να αναφερθεί ότι η ορθότητα συμπεριφοράς υπολογίζεται σε 37 παραλλαγές ανά τύπο και το nFR σε 31.

Παρατηρείται ότι η κανονική μορφή διατύπωσης της ερώτησης σημειώνει την υψηλότερη απόδοση στο σφραγισμένο σύνολο, με ορθότητα συμπεριφοράς 0,7027 και 0,7742 για τα ερωτήματα όπου αναμενόταν κανονική απάντηση. Αντίθετα, η καθομιλουμένη διατύπωση εμφανίζει τα χαμηλότερα ποσοστά, με αντίστοιχες τιμές 0,4595 και 0,4839.

Από αυτά τα αποτελέσματα απορρέει το συμπέρασμα ότι το σύστημα καταλαβαίνει καλύτερα μια σαφώς διατυπωμένη ερώτηση, ενώ δυσκολεύεται να αντιληφθεί ερωτήσεις με καθημερινό τόνο. Στις καθομιλουμένες ερωτήσεις, το σύστημα αρκετές φορές καταλαβαίνει τι είδους απόκριση πρέπει να δώσει, αλλά η απάντηση που δίνει έχει ελλειπείς πληροφορίες.

Τέλος, η κατηγορία των greeklish δείχνει να μην επηρεάζει πολύ την ποιότητα της απάντησης όπως κάνει η καθομιλουμένη. Η τιμή της ορθότητας συμπεριφοράς είναι λίγο μικρότερη από την κανονική διατύπωση, στο 0,6486, και το nFR στο 0,4973, γεγονός που αναδεικνύει την ικανότητα του συστήματος να αντιμετωπίζει τέτοιου είδους ερωτήματα.

5.4.3 Δύο μητρώα απόκρισης

Πέρα από τη συνολική ορθότητα συμπεριφοράς, εξετάστηκε η απόδοση του συστήματος για κάθε κατηγορία αναμενόμενων αποκρίσεων. Οι κατηγορίες περιλάμβαναν κανονική απάντηση, άρνηση, μερική απάντηση, αίτημα διευκρίνισης και διόρθωση λανθασμένης προϋπόθεσης.

Πίνακας 5.10: Ορθότητα συμπεριφοράς ανά αναμενόμενο τύπο απόκρισης

Αναμενόμενη απόκριση	Σφραγισμένο	n	Ανάπτυξης
Άρνηση	1,000 (14/14)	14	0,5357
Απάντηση	0,677	217	0,6889
Μερική απάντηση	0,000 (0/14)	14	0,0000
Αίτημα διευκρίνισης	0,000 (0/7)	7	0,0714
Διόρθωση προϋπόθεσης	0,000 (0/7)	7	0,1429

Τα αποτελέσματα αυτής της μέτρησης έδειξαν ότι το σύστημα λειτουργεί κυρίως με δύο μορφές απόκρισης, την κανονική απάντηση και την άρνηση. Στις υπόλοιπες κατηγορίες αναμενόμενης συμπεριφοράς απάντησης παρατηρήθηκε πλήρης αδυναμία του συστήματος να επιτύχει την αναμενόμενη συμπεριφορά για το σφραγισμένο σύνολο, κάτι που ήταν ήδη εμφανές και στο σύνολο ανάπτυξης και δείχνει σταθερή αδυναμία του συστήματος στις συγκεκριμένες κατηγορίες.

Οι αιτίες των αποτυχιών διαφέρουν ανά περίπτωση. Στα αιτήματα διευκρίνισης το σύστημα είτε αρνήθηκε είτε απάντησε χωρίς να ζητήσει τις απαιτούμενες πληροφορίες, ενώ στις περιπτώσεις λανθασμένης προϋπόθεσης συχνά αρνήθηκε ή υιοθέτησε την εσφαλμένη παραδοχή. Στις μερικές απαντήσεις δεν εντοπίστηκε διαρροή απαγορευμένου περιεχομένου, αλλά δεν απαντήθηκαν τα ζητούμενα γεγονότα.

Βέβαια, η πιο επιτυχημένη κατηγορία είναι αυτή της άρνησης. Παρόλο που οι περισσότερες κατηγορίες απέτυχαν στο να δώσουν την αναμενόμενη συμπεριφορά απάντησης, οι ερωτήσεις με άρνηση πέτυχαν και στις 14 περιπτώσεις, γεγονός που δείχνει τον ντετερμινισμό του συστήματος. Είναι προτιμότερο σε μερικές περιπτώσεις να δηλώνεται άρνηση παρά να παράγεται μια λανθασμένη απάντηση που μπορεί να οδηγήσει σε λάθος συμπεράσματα. Παράλληλα, η διαφορά από το χαμηλότερο αποτέλεσμα του συνόλου ανάπτυξης δεν ερμηνεύεται ως βελτίωση του συστήματος, καθώς το σφραγισμένο σύνολο περιείχε ευκολότερες περιπτώσεις άρνησης.

5.4.4 Άρνηση σε ερωτήματα εκτός πεδίου

Ο ρόλος του οργάνου Γ είναι να δοκιμάσει το σύστημα σε πολλές περιπτώσεις για να αξιολογηθεί η ικανότητά του να αρνείται σωστά σε ερωτήσεις εκτός πεδίου ή όταν η απαιτούμενη πληροφορία δεν ανακτήθηκε. Τα αποτελέσματα χωρίστηκαν σε κατηγορίες, καθώς έτσι διακρίνεται καλύτερα η απόδοση του συστήματος στις επιμέρους κατηγορίες.

Πίνακας 5.11: Ποσοστό άρνησης ανά κατηγορία, σύνολο ανάπτυξης (n = 30)

Κατηγορία	n	27/07/2026	28/07/2026
Γειτονικό πεδίο	13	0,615 (8/13)	1,000 (13/13)
Γενική γνώση	2	1,000	1,000
Ανύπαρκτη οντότητα	5	0,800	0,800
Χρονικό	2	0,500	0,500
Πιθανή αλλά ανύπαρκτη πολιτική	8	0,375	0,500
Συγκεντρωτικά	30	0,600	0,800

Από το σύνολο των μετρήσεων παρατηρήθηκε αύξηση της άρνησης στα ερωτήματα γειτονικού πεδίου, από 0,615 σε 1,000 μετά την εφαρμογή της πύλης πεδίου, η οποία αναγνωρίζει ερωτήσεις που αφορούν άλλα τμήματα και απορρίπτει το ερώτημα πριν από τη διαδικασία ανάκτησης. Η σημαντικότερη αδυναμία εμφανίστηκε στις πιθανές αλλά ανύπαρκτες πολιτικές, όπου το ποσοστό σωστής άρνησης ανήλθε από 0,375 σε 0,500 και παρέμεινε το χαμηλότερο όλων των κατηγοριών. Για τον λόγο αυτό το συνολικό αποτέλεσμα 0,800 δεν αντικατοπτρίζει ομοιόμορφη απόδοση μεταξύ των κατηγοριών.

Αφού ολοκληρώθηκε η ανάπτυξη της εφαρμογής, εκτελέστηκε το σύνολο του οργάνου Γ, σκοπός του οποίου είναι να αξιολογήσει την ικανότητα του συστήματος να αρνείται την παραγωγή απάντησης όταν το ερώτημα βρίσκεται εκτός του πεδίου λειτουργίας του ή όταν η απαιτούμενη πληροφορία δεν υπάρχει στη διαθέσιμη συλλογή. Επιπλέον, τα αποτελέσματα χωρίστηκαν σε κατηγορίες για την καλύτερη κατανόησή τους. Πρέπει να σημειωθεί ότι μια ερώτηση εξαιρέθηκε από το σύνολο, καθώς επιβεβαιώθηκε ότι η ζητούμενη πληροφορία υπήρχε στο corpus και για τον λόγο αυτό δεν αποτελούσε έγκυρη περίπτωση άρνησης.

Πίνακας 5.12: Ποσοστό άρνησης ανά κατηγορία, σφραγισμένο σύνολο (n = 23)

Κατηγορία	n	Ποσοστό άρνησης
Γειτονικό πεδίο	4	1,000
Γενική γνώση	5	1,000
Ανύπαρκτη οντότητα	4	1,000
Χρονικό	7	0,857 (6/7)
Πιθανή αλλά ανύπαρκτη πολιτική	3	1,000
Συγκεντρωτικά	23	0,9565 (22/23)

Το σφραγισμένο σύνολο παρουσίασε συνολικό ποσοστό άρνησης 0,9565, καθώς 22 από τις 23 έγκυρες περιπτώσεις αναγνωρίστηκαν ως αρνήσεις από τον αυτοματοποιημένο έλεγχο. Η μοναδική περίπτωση που δεν ταξινομήθηκε ως άρνηση αφορούσε χρονικό ερώτημα, χωρίς όμως να παράγει αβάσιμη πληροφορία, γεγονός που δείχνει έναν περιορισμό του λεξιλογικού κανόνα της μετρικής. Καλύτερη εικόνα παρουσίασε η κατηγορία των πιθανών αλλά ανύπαρκτων πολιτικών, όμως το μικρό μέγεθος του δείγματος δεν επιτρέπει την εξαγωγή ασφαλών γενικευμένων συμπερασμάτων. Συμπερασματικά, το γλωσσικό μοντέλο δεν έχει τη μοναδική ευθύνη για την παραγωγή της απάντησης, καθώς υπάρχουν πολλοί μηχανισμοί από τους οποίους περνά και επεξεργάζεται η πληροφορία πριν φτάσει σε αυτό, όπως η πύλη πεδίου και τα φίλτρα ανάκτησης.

5.4.5 Χειροκίνητη απογραφή ορθότητας

Η αξιολόγηση του συστήματος με αυτοματοποιημένες μετρικές δεν είναι πάντα η καλύτερη επιλογή, καθώς δεν μπορούν να προσδιορίσουν την πραγματική ορθότητα μιας απάντησης. Για τον λόγο αυτό, κρίθηκε αναγκαίο να πραγματοποιηθεί η χειροκίνητη αξιολόγηση και των 497 απαντήσεων του συνόλου ανάπτυξης. Ειδικότερα, οι απαντήσεις ταξινομήθηκαν σε προκαθορισμένες κατηγορίες, οι οποίες είναι: σωστή απάντηση, δικαιολογημένη άρνηση, αδικαιολόγητη άρνηση, σίγουρα λανθασμένη απάντηση και μη αποφασίσιμη περίπτωση.

Πίνακας 5.13: Χειροκίνητη ταξινόμηση 497 απαντήσεων ανά τύπο διατύπωσης

Κατηγορία	Βασ .	Παράφρ .	Greeklish	Τυπογρ .	Ελλειπτ .	Καθομ .	Φλύαρο ς	Σύνολο
Σωστή	39	27	31	34	35	25	26	217
Δικαιολογημένη άρνηση	5	4	4	4	3	4	5	29
Αδικαιολόγητη άρνηση	7	18	11	8	4	23	15	86
Σίγουρα λανθασμένη	19	21	24	25	29	18	24	160
Μη αποφασίσιμη	1	1	1	0	0	1	1	5

Το καλύτερο αποτέλεσμα αποδεικνύεται ότι έχει η βασική διατύπωση, με 39 σωστές απαντήσεις και 5 δικαιολογημένες αρνήσεις στις 71 περιπτώσεις. Οι παραλλαγές παρουσιάζουν διαφορετικά είδη αστοχίας, με την καθομιλουμένη και την παράφραση να εμφανίζουν περισσότερες αδικαιολόγητες αρνήσεις και την ελλειπτική και την ορθογραφικά λανθασμένη διατύπωση περισσότερες σίγουρα λανθασμένες απαντήσεις.

Πίνακας 5.14: Ανάλυση των 160 λανθασμένων απαντήσεων

Υποκατηγορία	Πλήθος	Περιγραφή
Λάθος αναφορικό	91	Σωστή τιμή συνδεδεμένη με λάθος αντικείμενο
Μη αποκρινόμενη	21	Σχετική απάντηση που δεν απαντά ουσιαστικά στο ερώτημα

Αντίφαση	19	Εσωτερικά ασυνεπής απάντηση ή αντίθεση με την πηγή
Παρωχημένη	16	Παλαιότερη πληροφορία που παρουσιάζεται ως τρέχουσα
Επινόηση	10	Ισχυρισμός που δεν υποστηρίζεται από κάποια πηγή
Σύγχυση οντοτήτων	3	Συγχώνευση δύο διαφορετικών οντοτήτων

Η συχνότερη μορφή λάθους είναι η περίπτωση της εσφαλμένης σύνδεσης μιας πραγματικής πληροφορίας με λάθος αντικείμενο, η οποία αποτελεί το μεγαλύτερο μέρος των λανθασμένων απαντήσεων. Οι επινοήσεις, από την άλλη, καταλαμβάνουν ένα αρκετά μικρό ποσοστό, μόλις το 2%, δηλαδή 10 από τις 497 συνολικές απαντήσεις.

Η διαπίστωση αυτή μπορεί να επαληθευτεί και από τον έλεγχο της ανάκτησης. Σε 104 από τις 135 ελέγξιμες λανθασμένες απαντήσεις, η σωστή πληροφορία υπήρχε ήδη στο ανακτημένο πλαίσιο. Ως αποτέλεσμα, μπορεί να διαπιστωθεί ότι στο μεγαλύτερο μέρος των λανθασμένων απαντήσεων, η αστοχία προκύπτει μετά την ανάκτηση, κατά την επιλογή ή σύνδεση των διαθέσιμων πληροφοριών. Αντίθετα, στις 81 από τις 86 αδικαιολόγητες αρνήσεις, παρατηρήθηκε ότι το σύστημα αρνιόταν λόγω έλλειψης επαρκούς πληροφορίας, καθώς μετά το φίλτρο εμπιστοσύνης του reranker δεν έμενε κανένα κατάλληλο chunk.

Τέλος, η χειροκίνητη αξιολόγηση δεν μπορεί να θεωρηθεί ως απόλυτο μέτρο σύγκρισης, καθώς πραγματοποιήθηκε από ένα μόνο άτομο και οι απαντήσεις είχαν παραχθεί πριν από κάποιες μεταγενέστερες βελτιώσεις του συστήματος.

5.4.6 Καθυστέρηση

Πέραν της ποιότητας των απαντήσεων, εξίσου καθοριστικός παράγοντας για την αξιολόγηση του συστήματος είναι ο χρόνος παραγωγής τους. Οι μετρήσεις που έγιναν χωρίζονται σε δύο κατηγορίες: αυτές για τις οποίες υπάρχει αποθηκευμένο αποτέλεσμα και αυτές που είχαν καταγραφεί μόνο κατά την εκτέλεση.

Πίνακας 5.15: Μετρήσεις καθυστέρησης και καθεστώς τεκμηρίωσης

Μέγεθος	Τιμή	Πληθυσμός	Καθεστώς
Ρυθμός αποκωδικοποίησης	8,14 μονάδες/δευτ. (2,2–13,8)	20 παραγωγές	Τεκμηριωμένο
Μήκος απάντησης	905 μονάδες, 92,8 δευτ. κατά μέσο όρο	20 παραγωγές	Τεκμηριωμένο
Επανακατατάκτης, p90 / p50	2.805,8 / 2.283,1 ms	20 κλήσεις, δεξαμενή 40	Τεκμηριωμένο
Σφραγισμένη εκτέλεση ανά γραμμή	161,2 δευτ. μέσος, 155,1 διάμεσος	301 γραμμές	Τεκμηριωμένο

Χρόνος πρώτης μονάδας, ψυχρός → θερμός	$\rightarrow \approx 28 \rightarrow \approx 1,3$ δευτ.	—	Αναφερόμενο
Μνήμη σημασιολογικών απαντήσεων	45 $\rightarrow$ 1,3 δευτ.	—	Αναφερόμενο

Η σημαντικότερη παρέμβαση απόδοσης αφορούσε τον χρόνο μέχρι την παραγωγή της πρώτης μονάδας, για τον οποίο αναφέρθηκε μείωση από περίπου 28 σε 1,3 δευτερόλεπτα μέσω της συμπίκνωσης της προτροπής και της επαναχρησιμοποίησης της στατικής μνήμης προθέματος. Αυτή η ένδειξη, όμως, αναγνωρίζεται ως αναφερόμενη, καθώς δεν υπάρχει κάποιο αρχείο που να αποδεικνύει τη γνησιότητά της.

Στις μετρήσεις που πραγματοποιήθηκαν, ο μέσος ρυθμός αποκωδικοποίησης ήταν 8,14 μονάδες ανά δευτερόλεπτο, ενώ ο επανακατατάκτης απαιτούσε περίπου 2–3 δευτερόλεπτα για την επεξεργασία 40 υποψηφίων.

Ο μέσος χρόνος 161,2 δευτερολέπτων της σφραγισμένης εκτέλεσης δεν αποτελεί αντιπροσωπευτική ένδειξη του χρόνου αναμονής του χρήστη. Η συγκεκριμένη μέτρηση λαμβάνει υπόψη την πλήρη παραγωγή της απάντησης χωρίς προοδευτική ροή και χωρίς τις βελτιστοποιήσεις μνήμης προθέματος που χρησιμοποιούνται στη ζωντανή λειτουργία του συστήματος.

5.5 Παρεμβάσεις που βελτίωσαν το σύστημα

Σε αυτή την ενότητα αναλύεται η επίδραση των παρεμβάσεων που έγιναν και για κάθε παρέμβαση εξετάζεται τόσο η βελτίωση που προκάλεσε όσο και το αν επηρέασε αρνητικά άλλα τμήματα του συστήματος.

Πίνακας 5.16: Συγκεντρωτική εικόνα των παρεμβάσεων

Μηχανισμός	Μέγεθος	Πριν	Μετά
Ανιχνευτής ονομαστικών καταλόγων	Recall@8	0,9886	1,0000
Φίλτρο εγγυμένου διαφημιστικού υλικού	Εντοπισμός / ψευδώς θετικά	—	21/21 · 0
Αναδιατύπωση σε ακαδημαϊκό ύφος	Ανάκληση, καθομιλούμενο	0,8378	0,9009
Τεμαχισμός ανά εγγραφή	Στόχοι σε θέση 1–2	0/5	5/5
Στένεμα επιπέδου γεγονότων	Ενεργοποιήσεις · ζημιά · ακρίβεια	19 · 13 · 0,316	6 · 0 · 1,000
Πύλη θεσμικής οντότητας	Διαρροή αριθμού προσώπου	10/10	0/10
Πύλη σχήματος αιτήματος	Αιτήματα γραφείου που εξυπηρετούνται	20/50	50/50

Αποκλιτισμός γενικής	Επίλυση επωνύμου σε γενική	56/62	62/62
----------------------	----------------------------	-------	-------

5.5.1 Φίλτρα εισόδου στη συλλογή

Κατά το στάδιο της αξιολόγησης διαπιστώθηκε ότι υπήρχαν δύο προβληματικές περιπτώσεις περιεχομένου στη συλλογή και για αυτό εφαρμόστηκαν μερικά φίλτρα για να αποκλείουν τέτοια τμήματα κειμένου από το ευρετήριο.

Η πρώτη περίπτωση αφορούσε έναν μεγάλο κατάλογο σίτισης 388 σελίδων, ο οποίος τεμαχιζόταν σε μεγάλο αριθμό σχεδόν πανομοιότυπων τμημάτων μορφής πίνακα και επηρέαζε την ανάκτηση σχετικών ανακοινώσεων. Για το πρόβλημα αυτό εφαρμόστηκε ένας έλεγχος που παρέλειπε ένα αρχείο αν περισσότερο από το 60% των γραμμών του είχαν μορφή πίνακα ή αποτελούνταν μόνο από αναγνωριστικά. Ως αποτέλεσμα, η μετρική Recall@8 σημείωσε αύξηση από 0,9886 σε 1,0000.

Η δεύτερη περίπτωση αφορούσε μία πηγή από το τμήμα, η οποία ήταν παραβιασμένη και είχε εισαχθεί διαφημιστικό περιεχόμενο τυχερών παιγνίων, το οποίο εντοπίστηκε σε ιστορικό στιγμιότυπο της συλλογής. Κατά συνέπεια, στο στάδιο της συλλογής των δεδομένων είχαν ευρετηριαστεί 21 τμήματα από αυτή τη σελίδα, praktiki.ihu.gr, ως doc_type: regulation και μπορούσαν να ανακτηθούν και να χρησιμοποιηθούν ως κανονισμοί του Τμήματος. Για την αντιμετώπιση του εν λόγω ζητήματος, εισήχθη ένα φίλτρο που απέκλειε τις συγκεκριμένες υπογραφές του ανεπιθύμητου περιεχομένου και εντόπισε και τα 21 τμήματα χωρίς κανένα ψευδώς θετικό αποτέλεσμα στη συλλογή των 8.545 τμημάτων.

Επιπλέον, πρέπει να σημειωθεί ότι πριν από την εφαρμογή των ειδικών φίλτρων δοκιμάστηκε μια πιο γενική ευρετική τεχνική, η οποία δεν είχε τα επιθυμητά αποτελέσματα, καθώς παρήγαγε ψευδώς θετικά αποτελέσματα σε έγκυρους πίνακες και έντυπα. Για τον λόγο αυτό προτιμήθηκε η πιο ειδική προσέγγιση, καθώς ένα υπερβολικά γενικό φίλτρο θα μπορούσε να αφαιρέσει πραγματικό περιεχόμενο από τη συλλογή.

5.5.2 Αναδιατύπωση σε ακαδημαϊκό ύφος

Όπως αναφέρθηκε και σε άλλες υποενότητες, η μορφή κειμένου σε καθομιλουμένη διατύπωση είναι μία από τις πιο απαιτητικές περιπτώσεις όσον αφορά την ανάκτηση του περιεχομένου και την απάντηση. Για την αντιμετώπιση αυτού του ζητήματος εφαρμόστηκε μηχανισμός αναδιατύπωσης, ο οποίος ενεργοποιείται μόνο όταν η υψηλότερη βαθμολογία της αρχικής ανάκτησης είναι μικρότερη από 0,28. Με αυτόν τον μηχανισμό, η ερώτηση αναδιατυπώνεται σε πιο προσιτή μορφή για το γλωσσικό μοντέλο με ακαδημαϊκό ύφος και στη συνέχεια η ανάκτηση εκτελείται ξανά. Όμως, για να αξιοποιηθούν τα νέα αποτελέσματα που θα παραχθούν, χρειάζεται να έχουν υψηλότερη βαθμολογία από τα αρχικά.

Πίνακας 5.17: Επίδραση της αναδιατύπωσης στη μέση ανάκληση του Οργάνου Β

Πληθυσμός	Πριν	Μετά
Καθομιλούμενη παράφραση, ανάπτυξης	0,8378	0,9009
Αναδιάταξη λέξεων, ανάπτυξης	0,9187	0,9431

Σύνολο ανάπτυξης	0,9424	0,9579
Σύνολο ελέγχου*	0,9091	0,9205

* Το αποτέλεσμα του συνόλου ελέγχου εξετάστηκε και κατά την απόφαση διατήρησης της συγκεκριμένης παρέμβασης.

Με αυτή την αλλαγή του μηχανισμού αναδιατύπωσης, υπήρξε γενική βελτίωση στο σύνολο ελέγχου, αλλά ειδικότερα στις καθομιλούμενες διατυπώσεις βελτιώθηκε η ανάκληση από 0,8378 σε 0,9009.

Παράλληλα, η κανονική μορφή διατύπωσης έμεινε αμετάβλητη, καθώς η αλλαγή δεν επηρέασε καθόλου τις ερωτήσεις που έχουν σωστή διατύπωση. Η ελάχιστη βαθμολογία που μετρήθηκε στο σύνολο του πρώτου τμήματος της λίστας κανονικής διατύπωσης ήταν 0,2889, με αποτέλεσμα ο μηχανισμός να μην ενεργοποιείται στις καθαρές ερωτήσεις. Ακόμη, δεν παρατηρήθηκε κάποια υποβάθμιση στις υπόλοιπες κατηγορίες παραλλαγών. Μία παλαιότερη τεχνική επιλογή ήταν η υποχρεωτική αναδιατύπωση ανεξαρτήτως της βαθμολογίας που θα προέκυπτε. Αυτό όχι μόνο δεν βελτίωσε τα αποτελέσματα αλλά τα επιδείνωσε και, ως τελική λύση, επιλέχθηκε αυτή η μέθοδος με την υψηλότερη βαθμολογία μικρότερη του 0,28.

5.5.3 Τεμαχισμός ανά εγγραφή

Σημαντική περίπτωση της συλλογής τμημάτων αποτελούσε μια κατηγορία εγγράφων που σε κάθε γραμμή της είχε ανεξάρτητες τιμές και δίπλα μια τιμή της μορφής «ετικέτα: τιμή». Το σύστημα δεν αναγνώριζε την ιδιαίτερη μορφή τους και τα τεμάχιζε όπως τα υπόλοιπα έγγραφα ως συνεχές κείμενο, με αποτέλεσμα ορισμένες εγγραφές να τοποθετούνται πέρα από τους πρώτους 512 χαρακτήρες του τμήματος και να μην είναι ορατές στον επανακατατάκτη. Σε αυτή την περίπτωση εγγράφων υιοθετήθηκε ο τεμαχισμός ανά εγγραφή, ώστε κάθε εγγραφή να δημιουργεί ξεχωριστό τμήμα.

Πίνακας 5.18: Θέση και βαθμολογία του φορέα της απάντησης πριν και μετά την παρέμβαση

Κλάση	Ερώτημα	Θέση πριν	Θέση μετά	Βαθμολογία
Στόχος	G03	κενή δεξαμενή	1	0,0029 → 0,898
Στόχος	I10	απών	1	— → 0,802
Στόχος	I07	απών	2	— → 0,622
Στόχος	I01	3	1	0,014 → 0,476
Στόχος	I06	1	1	0,162 → 0,393
Μάρτυρας	I05	1	1	0,674 → 0,683
Μάρτυρας	G05	2	2	0,780 → 0,781
Μάρτυρας	X14	1	1	0,238 → 0,688
Μάρτυρας	G04	1	1	0,925 → 0,987

Μετά την εφαρμογή του νέου τεμαχισμού, οι πέντε προβληματικές περιπτώσεις αποκαταστάθηκαν στην πρώτη ή δεύτερη θέση, ενώ παράλληλα δεν υποβαθμίστηκε κανένας από τους τέσσερις μάρτυρες ελέγχου. Χαρακτηριστικό παράδειγμα αποτελεί το G03, του οποίου η βαθμολογία αυξήθηκε από 0,0029 σε 0,898 χωρίς αλλαγή μοντέλου, κατωφλίου ή ερωτήματος.

Παράλληλα, αναμενόμενη συνέπεια αποτέλεσε η αύξηση του αριθμού των τμημάτων από 19 σε 39, χωρίς να επηρεαστεί κανένα άλλο αρχείο της συλλογής, η οποία τη στιγμή της μέτρησης αριθμούσε 8.677 σημεία. Η πύλη ανάκτησης παρέμεινε αμετάβλητη στις τιμές 1,0000 / 0,8877 / 0,8923 για το ίδιο στιγμιότυπο.

Είναι σημαντικό να αναφερθεί ότι η λύση αυτή εφαρμόστηκε μόνο στα δύο γνωστά αρχεία και όχι ως γενικός κανόνας του τεμαχιστή. Αυτό έγινε σκόπιμα, καθώς η γενίκευση αυτής της μεθόδου θα μπορούσε να δημιουργήσει αλλοίωση της συλλογής των τμημάτων.

5.5.4 Στένεμα του ντετερμινιστικού επιπέδου γεγονότων

Κατά το στάδιο της υλοποίησης εφαρμόστηκε ένας ντετερμινιστικός μηχανισμός, με στόχο την άμεση και ακριβέστερη απόκριση σε συγκεκριμένες κατηγορίες ερωτημάτων. Το επίπεδο αυτό μπορεί να παράγει απάντηση χωρίς τη χρήση του γλωσσικού μοντέλου, παρακάμπτοντας την ανάκτηση, τον επανακατατάκτη και τις επόμενες πύλες ελέγχου. Ωστόσο, σε περίπτωση λανθασμένης ενεργοποίησης, η παραγόμενη απάντηση δεν μπορεί να διορθωθεί από κάποιο μεταγενέστερο στάδιο. Στις 497 παραλλαγές του συνόλου ανάπτυξης, ο μηχανισμός ενεργοποιήθηκε αρχικά 19 φορές, από τις οποίες οι 13 ήταν λανθασμένες, με συνολική ακρίβεια 0,316.

Πίνακας 5.19: Διαδοχικές στενεύσεις του ντετερμινιστικού επιπέδου γεγονότων

Κύκλος	Αλλαγή	Ενεργοποιήσεις	Ζημιά	Ακρίβεια
Αφετηρία	—	19	13	0,316
1	Βέτο σε ερωτήματα προϋπόθεσης πιστωτικών μονάδων	15	9	0,400
2α	Αφαίρεση του γενικού «πότε» από τη σκανδάλη ωρολογίου	9	3	0,667
2β	Απαίτηση όρου ωρολογίου για την άρνηση	6	0	1,000

Στην πρώτη εκδοχή του μηχανισμού, υπήρχαν περιπτώσεις όπου επέστρεφε σωστή πληροφορία για λάθος ερώτημα. Χαρακτηριστικό παράδειγμα ήταν ερώτηση σχετικά με τις πιστωτικές μονάδες που απαιτούνται για την έναρξη πρακτικής άσκησης, στην οποία το σύστημα επέστρεφε τις μονάδες που αποδίδει η ίδια η πρακτική. Σε όλο το σύνολο ανάπτυξης που είναι 497 παραλλαγές ερωτήσεων, ο μηχανισμός ενεργοποιήθηκε 19 φορές, εκ των οποίων οι 6 ήταν σωστές, με ακρίβεια 0,316. Έπειτα από μερικές παραμετροποιήσεις στα κριτήρια ενεργοποίησης, οι ενεργοποιήσεις μειώθηκαν σε 15 και τελικά σε 6. Παράλληλα, η ακρίβεια βελτιώθηκε αισθητά, καθώς ανέβηκε σε 0,400 και έπειτα σε 1,000.

Οι ορθές ενεργοποιήσεις παρέμειναν οι ίδιες σε όλη τη διαδρομή εκείνο που εξαλείφθηκε ήταν οι εσφαλμένες. Εκείνες οι παραλλαγές των ερωτήσεων που δημιουργούσαν πρόβλημα στο σύστημα σταμάτησαν να ενεργοποιούν τον μηχανισμό και ακολούθησαν την κανονική διαδρομή του συστήματος, όπου η ανάκληση γεγονότων τους αυξήθηκε από 0,000 σε 1,000.

Στη συνέχεια έγιναν ακόμη δύο αλλαγές στον μηχανισμό που τον κατέστησαν ακόμη πιο αυστηρό στις ενεργοποιήσεις του. Στην τελική εκδοχή του ενεργοποιήθηκε έξι φορές στο σύνολό του χωρίς να καταγραφεί κάποιο λάθος. Η πύλη ανάκτησης παρέμεινε αμετάβλητη στο 1,0000 / 0,8877 / 0,8923, αποδεικνύοντας ότι δεν επηρέασε το επίπεδο της ανάκτησης αρνητικά. Ο βασικός έλεγχος, όμως, έγινε με τη χειροκίνητη εξέταση των παραλλαγών που απελευθερώθηκαν. Ωστόσο, μετά τη δεύτερη στένευση, ο κλάδος άρνησης του επιπέδου δεν ενεργοποιείται σε καμία από τις 497 παραλλαγές. Επομένως, δεν μπορεί να παραχθεί κάποιο συμπέρασμα για την αποτελεσματικότητα αυτού του μηχανισμού.

5.5.5 Οι δύο πύλες αποχής στον χειριστή επικοινωνίας

Κατά την υλοποίηση κρίθηκε απαραίτητο να δημιουργηθεί ένας ντετερμινιστικός μηχανισμός για τον χειρισμό ερωτημάτων που αφορούν στοιχεία επικοινωνίας, όπως αριθμούς τηλεφώνου και γραφεία. Ο βασικός σκοπός ήταν να αναγνωρίζει το άτομο ή την οντότητα στην οποία αναφέρεται η ερώτηση και να απαντάει αντίστοιχα χωρίς να απαιτείται παραγωγή από το γλωσσικό μοντέλο. Ωστόσο, κατά τη διαδικασία αξιολόγησης εντοπίστηκαν δύο περιπτώσεις όπου ο μηχανισμός ενεργοποιήθηκε λανθασμένα, γεγονός που οδήγησε στην προσθήκη δύο επιπλέον πυλών ελέγχου.

Πίνακας 5.20: Αποτελέσματα των δύο πυλών αποχής

Πληθυσμός	n	Πύλη οντότητας θεσμικής	Πύλη σχήματος αιτήματος
Κατασκευασμένη βλάβη	10	10/10 → 0/10	—
Μητρώο βλάβης	13	—	3 → 1
Αιτήματα γραφείου	50	—	20/50 → 50/50
Κανονικά θετικά	35	35/35 → 35/35	35/35 → 35/35
Ύψη αιτήματος	50	—	40/40 → 40/40
Φύλακας σύγκρουσης	9	9/9 → 9/9	9/9 → 9/9

Η πρώτη πύλη ελέγχου που εφαρμόστηκε ενεργοποιείται όταν το ερώτημα κατονομάζει θεσμική οντότητα. Χωρίς αυτήν την παραμετροποίηση, μπορούσε να αντιστοιχίσει πληροφορίες σε λανθασμένα πρόσωπα. Για παράδειγμα, ερώτημα όπως «το τηλέφωνο της γραμματείας για θέματα του X» μπορούσε να επιστρέψει τον αριθμό τηλεφώνου του προσώπου X αντί της γραμματείας. Μετά από αυτή την αλλαγή, και οι 10 περιπτώσεις βλάβης εξουδετερώθηκαν, χωρίς απώλεια στις 35 κανονικές θετικές περιπτώσεις.

Η δεύτερη αλλαγή είναι πιο μικρή αλλά εξίσου σημαντική, καθώς απαιτεί το ερώτημα να ζητάει πράγματι στοιχεία επικοινωνίας και όχι απλώς να περιέχει μια σχετική λέξη. Για παράδειγμα, η πρόταση «πρέπει να κλείσω το τηλέφωνο στη διάλεξη του Χ;» δεν αποτελεί αίτημα για αριθμό τηλεφώνου, παρότι περιέχει τη λέξη «τηλέφωνο».

Οι παρεμβάσεις αυτές δοκιμάστηκαν σε ειδικά διαμορφωμένο σύνολο ερωτήσεων, το οποίο περιλάμβανε προβληματικές και κανονικές περιπτώσεις χρήσης.

5.5.6 Αποκλιτισμός της γενικής πτώσης

Κατά το στάδιο της αξιολόγησης του μηχανισμού αντιστοίχισης προσωπικού διαπιστώθηκε πρόβλημα στην αναγνώριση επωνύμων όταν αυτά εμφανίζονταν στη γενική πτώση. Πιο συγκεκριμένα, 6 από τα 64 επώνυμα του ευρετηρίου δεν επιλύονταν σωστά στη γενική, παρότι αναγνωρίζονταν στην ονομαστική.

Η αστοχία αυτή δεν ήταν τυχαία, καθώς εμφανιζόταν σε επώνυμα που κατέληγαν σε «-ος». Όταν τα επώνυμα χρησιμοποιούνταν στη γενική, η βαθμολογία ομοιότητάς τους με την αποθηκευμένη ονομαστική μορφή μπορούσε να πέσει κάτω από το κατώφλι 0,90, ιδιαίτερα στα μικρότερα επώνυμα. Για την επίλυση αυτού του προβλήματος προστέθηκε ειδικό βήμα μετατροπής της γενικής στην αντίστοιχη ονομαστική μορφή πριν από την ασαφή αντιστοίχιση.

Σημαντική αλλαγή παρατηρήθηκε με την αλλαγή της πτώσης, καθώς η αναγνώριση στη γενική αυξήθηκε από 56/62 σε 62/62, ενώ η αναγνώριση στην ονομαστική παρέμεινε αμετάβλητη στο 64/64. Παράλληλα, οι υπόλοιποι πληθυσμοί ελέγχου παρέμειναν αμετάβλητοι.

Επιπλέον, δοκιμάστηκε και η μείωση του κατωφλίου από 0,90 σε 0,83 ώστε να αντιστοιχίζονται από τον μηχανισμό, όμως η αλλαγή αυτή μπορούσε να οδηγήσει σε λανθασμένη αντιστοίχιση διαφορετικών πραγματικών επωνύμων και για τον λόγο αυτό απορρίφθηκε.

Είναι σημαντικό να σημειωθεί ότι η παρέμβαση αυτή διορθώνει την αναγνώριση της ζητούμενης οντότητας, όμως δεν εγγυάται ότι θα παράγει την αναμενόμενη απάντηση, καθώς εξαρτάται και από την επίδοση άλλων μηχανισμών του συστήματος.

5.6 Παρεμβάσεις που απορρίφθηκαν και το εύρημα της απορρόφησης

5.6.1 Η απορρόφηση των βελτιώσεων από τον επανακατατάκτη

Η διαδικασία της ανάκτησης αποτελείται από ένα πυκνό και ένα αραιό σκέλος. Τα αποτελέσματα των δύο μεθόδων συγχωνεύονται σε μία κοινή δεξαμενή 40 υποψηφίων αποσπασμάτων. Στη συνέχεια, ο επανακατατάκτης αξιολογεί ξανά κάθε ζεύγος ερωτήματος και αποσπάσματος, χωρίς να αξιοποιεί τις αρχικές βαθμολογίες των δύο σκελών. Κατά την αξιολόγηση διαπιστώθηκε ότι η κατάταξη ενός επιμέρους σκέλους μπορεί να βελτιωθεί περαιτέρω χωρίς να επηρεαστεί η τελική κατάταξη.

Παρέμβαση	Μέτρηση στο σκέλος	Επίδραση στην πύλη	Καθεστώς
Ελληνικός στελεχωτής στο αραιό σκέλος	MRR +0,24	μηδενική	αναφερόμενο
Καθυστερημένος τεμαχισμός	MRR -0,07 στο πυκνό	δεν ενσωματώθηκε	αναφερόμενο
Προσαρμοστική στάθμιση συγχώνευσης	μόνο πυκνό $\approx$ υβριδικό	μηδενικό περιθώριο	αναφερόμενο
Σάρωση παραμέτρων ανάκτησης	MRR σταθερό για $k > 6$	καμία διαμόρφωση δεν ξεπέρασε τη βάση	τεκμηριωμένο

Οι βελτιώσεις στις βαθμολογίες του πυκνού και του αραιού σκέλους δεν μεταφέρονται απαραίτητα στην τελική κατάταξη. Εφόσον ένα σχετικό απόσπασμα έχει ήδη εισέλθει στη δεξαμενή των υποψηφίων, ο επανακατατάκτης το βαθμολογεί εκ νέου με βάση τη συνάφειά του με το ερώτημα. Αυτό δεν καθιστά τις βελτιώσεις των μηχανισμών ανάκτησης περιττές, καθώς εξακολουθούν να επηρεάζουν ποια αποσπάσματα θα εισέλθουν αρχικά στη δεξαμενή των υποψηφίων.

Αντίστοιχο είναι το συμπέρασμα για τα αποσπάσματα που δεν πρέπει να χρησιμοποιηθούν. Η απλή μείωση της αρχικής βαθμολογίας ενός αποσπάσματος δεν εξασφαλίζει τον αποκλεισμό του, καθώς ο επανακατατάκτης μπορεί να το τοποθετήσει εκ νέου σε υψηλή θέση.

Το παραπάνω εύρημα αφορά κυρίως αποσπάσματα που έχουν ήδη εισέλθει στη δεξαμενή υποψηφίων και δεν σημαίνει ότι έχει επιλυθεί συνολικά το πρόβλημα της ανάκτησης. Πραγματοποιήθηκαν σχετικές δοκιμές για τη διεύρυνση της δεξαμενής των υποψηφίων τμημάτων και αυτό αύξησε τον αριθμό των σχετικών φορέων που ανακτήθηκαν. Ωστόσο, δεν ήταν αντίστοιχη η αύξηση των φορέων που έφτασαν στις πρώτες θέσεις μετά την επανακατάταξη. Επομένως, δεν αρκεί μόνο η αύξηση του αριθμού υποψηφίων που επιλέγονται στην κατάταξη αλλά και η παραγωγή κατάλληλων υποψηφίων.

5.6.2 Παρεμβάσεις με μετρήσιμο κόστος

Σε αυτή την ενότητα θα αναφερθούν οι παρεμβάσεις που έγιναν στο σύστημα και το υποβάθμισαν.

Πίνακας 5.22: Επίδραση της διεύρυνσης του παραθύρου εισόδου του επανακατατάκτη

Κύκλος	Παράθυρο	Recall@8	MRR	nDCG@8	p90	Λόγος
19/07	512 χαρακτήρες	1,0000	0,8915	0,8965	3.955,8 ms	1,00×
19/07	1.657 χαρακτήρες	0,9778	0,8452	0,8615	14.284,3 ms	3,61×
19/07	Πλήρες τμήμα	0,9778	0,8341	0,8533	10.366,7 ms	2,62×
07/08	512 χαρακτήρες	1,0000	0,8902	0,8947	2.805,8 ms	1,00×
07/08	900 χαρακτήρες	0,9778	0,8574	0,8723	4.962,1 ms	1,77×

Στη συγκεκριμένη αξιολόγηση πραγματοποιήθηκαν δύο ξεχωριστοί κύκλοι, καθένας από τους οποίους χρησιμοποιεί διαφορετικό έλεγχο, καθώς η συλλογή είχε μεταβληθεί στο ενδιάμεσο. Για αυτόν τον λόγο, τα αποτελέσματα μεταξύ των κύκλων συγκρίνονται μόνο με το αντίστοιχο σημείο αναφοράς του ίδιου κύκλου.

Μία τεχνική αλλαγή που αποφασίστηκε στο στάδιο της υλοποίησης ήταν η διεύρυνση του παραθύρου σε περισσότερους χαρακτήρες. Αυτό αναμενόταν να βοηθήσει το μοντέλο, ώστε να ληφθεί υπόψη το ευρύτερο πλαίσιο αναφοράς. Αντίθετα, σε όλες τις δοκιμές που πραγματοποιήθηκαν παρατηρήθηκε υποβάθμιση των μετρικών και αύξηση του χρόνου επεξεργασίας. Μια πιθανή εξήγηση είναι ότι το περιορισμένο παράθυρο διατηρεί μεγαλύτερη έμφαση στο αρχικό και συχνά περισσότερο συγκεκριμένο τμήμα του αποσπάσματος, ενώ η προσθήκη επιπλέον κειμένου μπορεί να ενισχύσει μεγαλύτερα αλλά λιγότερο σχετικά αποσπάσματα.

Μια ακόμη σχεδιαστική αλλαγή ήταν η αφαίρεση της κεφαλίδας μεταδεδομένων από την είσοδο του επανακατατάκτη, με στόχο να αυξηθεί ο διαθέσιμος χώρος για το κυρίως περιεχόμενο. Σε μια πρώτη μεμονωμένη μέτρηση τα αποτελέσματα ήταν αρκετά καλά, καθώς σημείωσε 0,9872 έναντι 0,9745 με την κεφαλίδα. Όταν όμως δοκιμάστηκε σε μεγαλύτερο σύνολο αξιολόγησης με περισσότερα απαιτητικά δεδομένα, η αφαίρεση της κεφαλίδας προκάλεσε μείωση του MRR κατά 0,0917. Ως αποτέλεσμα, διατηρήθηκε η κεφαλίδα μεταδεδομένων, καθώς παρείχε χρήσιμη πληροφορία για κάθε αρχείο. Από τα παραπάνω προκύπτει το εξής συμπέρασμα: η αξία ενός χαρακτηριστικού δεν μπορεί πάντα να εκτιμηθεί εξετάζοντας έναν υποψήφιο απομονωμένα.

Τέλος, εξετάστηκε η αφαίρεση ιστορικού και ληγμένου περιεχομένου πριν από την επανακατάταξη. Αυτή η αλλαγή μείωσε αισθητά την παρουσία ιστορικού και ληγμένου περιεχομένου, αλλά οδήγησε στην απώλεια τριών προηγούμενων διαθέσιμων απαντήσεων. Για τον λόγο αυτό, αποφασίστηκε ότι δεν αποτελεί σημαντική διαφορά στην επίδοση του συστήματος και, εν τέλει, η παρέμβαση δεν διατηρήθηκε.

5.6.3 Οι κριτές ποιότητας απάντησης

Μια ακόμη απορριφθείσα επιλογή ήταν να ενταχθούν δύο κριτές για την αυτόματη ανίχνευση πιθανών λανθασμένων απαντήσεων μετά την παραγωγή τους. Και οι δύο απορρίφθηκαν, καθώς παρουσίασαν σημαντικό ποσοστό λανθασμένων ενεργοποιήσεων.

Πίνακας 5.23: Αποτελέσματα των δύο κριτών

Κριτής	Μέτρηση	Αποτέλεσμα
Κριτής συνεπαγωγής, γύρος 1	Ψευδώς θετικά	0,53
Κριτής συνεπαγωγής, γύρος 2	Ψευδώς θετικά (n=15) / Αληθώς θετικά (n=5)	0,200 / 0,800
Αξιολογητής επάρκειας πλαισίου	Εντοπισμός εκτός πεδίου	0,8182
Αξιολογητής επάρκειας πλαισίου	Ψευδής ενεργοποίηση σε ορθές απαντήσεις (n=114)	0,1842

Ο πρώτος κριτής ήταν υπεύθυνος για την εύρεση ασυμφωνιών της παραγόμενης απάντησης με τα ανακτημένα αποσπάσματα. Αυτή η προσθήκη σημείωσε ποσοστό ψευδώς θετικών 53%. Μετά από παραμετροποίηση του μηχανισμού, το ποσοστό αυτό μειώθηκε στο 20%, που ακόμη ήταν αρκετά υψηλό για τα δεδομένα του συστήματος. Η βασική αδυναμία του μηχανισμού εμφανιζόταν σε απαντήσεις που περιείχαν πολλαπλά στοιχεία, καθώς μία πρόταση μπορούσε να αξιολογηθεί σε σχέση με απόσπασμα που αφορούσε διαφορετικό στοιχείο της ίδιας απάντησης. Για τον λόγο αυτό, το πρόβλημα θεωρήθηκε δομικό και όχι απλώς ζήτημα ρύθμισης του κατωφλίου.

Ο δεύτερος κριτής ήταν το ίδιο γλωσσικό μοντέλο και εφαρμόστηκε για να αξιολογήσει αν το ανακτημένο πλαίσιο ήταν επαρκές για την παραγωγή απάντησης. Τα αποτελέσματα χρήσης του έδειξαν μεγάλη επιτυχία σε ερωτήματα που αφορούσαν περιεχόμενο εκτός πεδίου, καθώς εντόπισε το 81,8%. Ωστόσο, δεν απέδωσε το ίδιο καλά σε ορθές ερωτήσεις, καθώς ενεργοποιήθηκε λανθασμένα στο 18,4%. Αυτό σημαίνει ότι μία στις πέντε ερωτήσεις που έπρεπε να απαντήσει κανονικά τις απέρριπτε. Αυτή η συμπεριφορά μπορεί να οφείλεται στη χρήση του ίδιου μοντέλου για την παραγωγή και την αξιολόγηση. Αυτό οφείλεται στην περιορισμένη ανεξαρτησία ελέγχου, καθώς ένα θεματικά σχετικό αλλά ανεπαρκές απόσπασμα μπορεί να οδηγήσει σε παρόμοιο σφάλμα και στα δύο στάδια.

Συνολικά, οι δοκιμές δεν ανέδειξαν κάποιον απλό και αξιόπιστο μηχανισμό αυτόματου ελέγχου της ποιότητας των παραγόμενων απαντήσεων.

5.6.4 Υπόλοιπες απορρίψεις

Πέρα από τις παρεμβάσεις που εξετάστηκαν αναλυτικότερα, δοκιμάστηκαν και επιπλέον αλλαγές οι οποίες δεν διατηρήθηκαν. Εξετάστηκε η συνεχής αναδιατύπωση των ερωτήσεων ανεξάρτητα από την κατάστασή τους και οι μετρήσεις έδειξαν ότι μείωσε την ανάκληση του τότε συνόλου ελέγχου από 0,9643 σε 0,9286. Αυτό το αποτέλεσμα οδήγησε στην επιλογή της Υποενότητας 5.5.2, όπου ενεργοποιούνταν ο μηχανισμός μόνο όταν η αρχική ανάκτηση εμφάνιζε χαμηλή βαθμολογία.

Μια άλλη περίπτωση ήταν η γενική διόρθωση ορθογραφικών λαθών, που απορρίφθηκε σε δύο διαφορετικές δοκιμές. Οι δοκιμές κατέγραψαν αποτέλεσμα με μείωση του MRR κατά 0,036 και 0,018 αντίστοιχα. Ο συγκεκριμένος μηχανισμός μπορούσε να μετατρέψει άτονους τεχνικούς όρους σε διαφορετικές έγκυρες λέξεις και, επομένως, να μεταβάλει το αρχικό νόημα του ερωτήματος. Αντίστοιχα, η πλήρης αφαίρεση του αραιού σκέλους ανάκτησης απορρίφθηκε, καθώς προκάλεσε απώλεια σε ερωτήματα που περιείχαν μόνο συγκεκριμένους κωδικούς μαθημάτων, τους οποίους το πυκνό σκέλος δεν αναγνώριζε με την ίδια αξιοπιστία.

Αναλόγως, δοκιμάστηκε επίσης η αλλαγή του γλωσσικού μοντέλου καθώς και των αποκωδικοποιημένων εκδόσεών του. Η κερδοσκοπική αποκωδικοποίηση παρείχε επιτάχυνση 23–39% στις αρχικές δοκιμές, αλλά σε μεγαλύτερη εκτέλεση είκοσι παραγωγών προκάλεσε διακοπές του επιταχυντή. Επιπλέον, η έκδοση 6-bit του μοντέλου αξιολογήθηκε και διαπιστώθηκε ότι ήταν πιο αργή καθώς και περισσότερο επιρρεπής σε μη υποστηριζόμενες απαντήσεις. Ακόμη, ο εναλλακτικός επανακατατάκτης που εκτιμήθηκε παρουσίασε χειρότερη απόδοση σε όλες τις εξεταζόμενες διαιρέσεις, με μείωση του MRR του συνόλου ελέγχου κατά 0,169.

Οι παραπάνω απορρίψεις συνέβαλαν στον προσδιορισμό των ορίων της υφιστάμενης αρχιτεκτονικής και των παρεμβάσεων που δεν προσφέρουν μετρήσιμο όφελος στο συγκεκριμένο σύστημα.

5.7 Περιορισμοί της αξιολόγησης

Η αξιολόγηση του συγκεκριμένου συστήματος παρουσιάζει μερικούς περιορισμούς, οι οποίοι πρέπει να λαμβάνονται υπόψη κατά την ερμηνεία των αποτελεσμάτων. Οι σημαντικότεροι αφορούν την κάλυψη των συνόλων δοκιμών, την αναπαραγωγιμότητα ορισμένων πειραμάτων, τον τρόπο διεξαγωγής των μετρήσεων και ορισμένες διαστάσεις που δεν εξετάστηκαν.

5.7.1 Κάλυψη των συνόλων δοκιμών

Το σύνολο των δοκιμών που πραγματοποιήθηκε κατά το στάδιο της αξιολόγησης δεν καλύπτει όλες τις πιθανές μορφές ερωτημάτων που μπορεί να δεχθεί το σύστημα. Υπήρξαν περιπτώσεις που το διαθέσιμο σύνολο δεν επαρκούσε για την πραγματική αποτίμηση της επίδοσης του συστήματος, όπως στην περίπτωση των δύο πυλών ελέγχου στοιχείων επικοινωνίας της Υποενότητας 5.5.5.

Οι δοκιμές αυτές επιτρέπουν τον έλεγχο συμπεριφοράς του μηχανισμού σε συγκεκριμένες συνθήκες, αλλά δεν αποτελούν ισοδύναμο πραγματικής χρήσης. Για αυτό, τα παραγόμενα αποτελέσματα πρέπει να ερμηνεύονται ως ελεγχόμενες δοκιμές και όχι ως μέτρηση της συχνότητας ή της συμπεριφοράς τους σε πραγματικό πληθυσμό χρηστών.

5.7.2 Τεκμηρίωση και αναπαραγωγιμότητα

Παράλληλα, είναι σημαντική η καταγραφή των αποτελεσμάτων που παρήχθησαν για κάθε πειραματική αξιολόγηση. Ωστόσο, δεν διατηρήθηκε αποθηκευμένο αρχείο αποτελεσμάτων για κάθε περίπτωση που αξιολογήθηκε. Χαρακτηριστικό παράδειγμα αποτελεί η περίπτωση της Υποενότητας 5.6, όπου έγιναν 17 παρεμβάσεις και από αυτές μόνο οι πέντε διαθέτουν αποθηκευμένο αρχείο αποτελεσμάτων, ενώ οι υπόλοιπες τεκμηριώνονται κυρίως μέσω των καταγραφών της διαδικασίας ανάπτυξης.

Για τις υπόλοιπες, επομένως, το αρχικό αποτέλεσμα δεν μπορεί να επαληθευτεί με την ίδια ευκολία όπως στις παρεμβάσεις για τις οποίες έχει διατηρηθεί πλήρες αρχείο εξόδου. Για αυτό χρησιμοποιούνται ως συμπληρωματικά ευρήματα και όχι ως οι βασικοί δείκτες απόδοσης του συστήματος.

5.7.3 Περιορισμοί των μετρήσεων

Το σύνολο ελέγχου αξιοποιήθηκε για την αξιολόγηση του συστήματος και την αποτίμηση της αποδοτικότητας του μηχανισμού. Ορισμένα ερωτήματα χρησιμοποιήθηκαν για τη βελτίωση και τη διόρθωση ορισμένων αδυναμιών που παρατηρήθηκαν. Αντίστοιχα, μικρό μέρος του σφραγισμένου συνόλου εκτός πεδίου είχε χρησιμοποιηθεί σε προγενέστερους ελέγχους, με αποτέλεσμα μόνο ένα υποσύνολό του να παραμένει πραγματικά αθέατο μέχρι την τελική αξιολόγηση.

Η χειροκίνητη αξιολόγηση των 497 απαντήσεων πραγματοποιήθηκε από έναν μόνο αξιολογητή. Επομένως, παρότι είχαν οριστεί εκ των προτέρων συγκεκριμένα κριτήρια ταξινόμησης, δεν ήταν δυνατός ο υπολογισμός συμφωνίας μεταξύ ανεξάρτητων βαθμολογητών.

Συνοψίζοντας, ένα μέρος των μετρήσεων έγινε σε διαφορετική χρονική περίοδο, με αποτέλεσμα να υπάρξει αλλαγή στις εκδόσεις της βάσης γνώσης. Για αυτό, οι τιμές των μετρήσεων πρέπει να αντιστοιχούν στο συγκεκριμένο snapshot της συλλογής που υπήρχε τη στιγμή της μέτρησης.

5.7.4 Διαστάσεις που δεν αξιολογήθηκαν

Οι μετρήσεις που πραγματοποιήθηκαν κατά το στάδιο της αξιολόγησης δεν περιλαμβάνουν τη μελέτη πραγματικών χρηστών υπό ελεγχόμενες συνθήκες, επομένως δεν μετρήθηκαν παράγοντες όπως η χρηστικότητα, η ικανοποίηση των φοιτητών ή η επίδραση του συστήματος στον χρόνο αναζήτησης πληροφοριών. Επίσης, δεν υπάρχει συγκριτική μέτρηση με άλλα συστήματα ή εργαλεία παρόμοιου σκοπού. Ακόμη, είναι σημαντικό να αναφερθεί ότι οι μετρήσεις καθυστέρησης πραγματοποιήθηκαν σειριακά σε ένα μηχάνημα και δεν εξετάστηκε η συμπεριφορά του συστήματος υπό ταυτόχρονη χρήση από πολλούς χρήστες.

Κεφάλαιο 6ο: Συμπεράσματα και μελλοντικές επεκτάσεις

6.1 Σύνοψη και απαντήσεις στα ερευνητικά ερωτήματα

Η παρούσα εργασία ολοκληρώθηκε με την ανάπτυξη ενός λειτουργικού ελληνόγλωσσου συστήματος γραμματειακής υποστήριξης, το οποίο μπορεί να δεχτεί ερωτήματα μέσω διαδικτυακής διεπαφής και να εκτελείται εξ ολοκλήρου τοπικά.

Η απάντηση στο πρώτο ερευνητικό ερώτημα E1 είναι ότι ένα σύστημα RAG που απευθύνεται σε φοιτητές και λειτουργεί τοπικά μπορεί να υποστηρίξει αποτελεσματικά την ανάκτηση πληροφοριών από ετερογενείς θεσμικές πηγές. Τη μεγαλύτερη επιτυχία του συστήματος σημειώνει η ανάκτηση, ενώ μεγαλύτερο περιθώριο βελτίωσης παρατηρήθηκε στο στάδιο παραγωγής της τελικής απάντησης.

Ιδιαίτερο είναι το ενδιαφέρον όσον αφορά την απάντηση στο δεύτερο ερώτημα E2. Διαπιστώθηκε ότι η διατύπωση του ερωτήματος και οι διαφορετικές παραλλαγές του μπορούν να επηρεάσουν αρνητικά την απόδοση του συστήματος. Η ανάκτηση παρουσίασε υψηλή ευρωστία στις διαφορετικές μορφές διατύπωσης, ενώ μεγαλύτερη μείωση παρατηρήθηκε στην ποιότητα της τελικής απάντησης. Η μεγαλύτερη αδυναμία του συστήματος ήταν η καθομιλουμένη διατύπωση, ενώ τα greeklish και τα ορθογραφικά αποδείχτηκαν λιγότερο επιζήμια. Με βάση τα αποτελέσματα αυτά, συνάγεται ότι η διαφορά ανάμεσα στο λεξιλόγιο που χρησιμοποιεί ο φοιτητής και εκείνο των επίσημων εγγράφων μπορεί να επηρεάζει περισσότερο την απόδοση από τη μορφή γραφής των χαρακτήρων.

Παράλληλα, το E3 εξετάστηκε κυρίως μέσω της χειροκίνητης αξιολόγησης των 497 απαντήσεων. Η κυρίαρχη μορφή σφάλματος δεν ήταν η επινόηση περιεχομένου, η οποία περιορίστηκε περίπου στο 2% των απαντήσεων, αλλά η σύνδεση πραγματικής πληροφορίας με λανθασμένο αντικείμενο. Το μεγαλύτερο μέρος των σφαλμάτων προήλθε μετά το στάδιο της ανάκτησης, κατά την παραγωγή της απάντησης από το γλωσσικό μοντέλο, όταν επιλέγεται η κατάλληλη σύνδεση των διαθέσιμων στοιχείων. Αυτό αποδεικνύει το περιθώριο ανάπτυξης που έχει το στάδιο μετά την ανάκτηση και το στάδιο της παραγωγής της απάντησης. Παράλληλα, στο ίδιο εύρος του ερωτήματος εντάσσεται και η συμπεριφορά του συστήματος όταν η διαθέσιμη πληροφορία δεν επαρκεί. Σε αυτή την περίπτωση οι ντετερμινιστικοί μηχανισμοί παρουσίασαν υψηλή επιτυχία στην άρνηση ερωτήσεων εκτός πεδίου. Ωστόσο, δεν σημειώθηκε η ίδια επιτυχία σε ερωτήματα που απαιτούσαν πιο σύνθετους τρόπους απόκρισης, όπως η μερική απάντηση, το αίτημα διευκρίνισης και η διόρθωση λανθασμένης προϋπόθεσης.

Ως προς το ερώτημα E4, η αξιολόγηση απέδειξε ότι δεν απέφεραν όλες οι αλλαγές την ίδια βελτίωση στην τελική απόδοση του συστήματος. Ορισμένες παρεμβάσεις απέδωσαν σαφές όφελος, ενώ άλλες βελτίωσαν μόνο μεμονωμένα στάδια χωρίς αντίστοιχη επίδραση στο τελικό αποτέλεσμα. Το εύρημα αυτό αναδεικνύει τη σημασία της αξιολόγησης των αλλαγών με βάση τη συνολική λειτουργία του συστήματος και όχι μόνο την απόδοση των επιμέρους συστατικών του.

6.2 Συνεισφορά

Η συνεισφορά της εργασίας μπορεί να διακριθεί σε τρία επίπεδα.

Σε επίπεδο υλοποίησης, αναπτύχθηκε ένα ολοκληρωμένο σύστημα γραμματειακής υποστήριξης στα ελληνικά που λειτουργεί εξ ολοκλήρου τοπικά, χωρίς εξάρτηση από εξωτερικές υπηρεσίες κατά την εκτέλεσή του και εντός περιορισμένων υπολογιστικών πόρων. Η υλοποίηση συνδυάζει ντετερμινιστικές διαδρομές με σημασιολογική ανάκτηση και παραγωγή φυσικής γλώσσας, επιτρέποντας την επιλογή διαφορετικού μηχανισμού ανάλογα με το είδος του ερωτήματος.

Σε επίπεδο μεθοδολογίας, η αξιολόγηση χωρίστηκε σε τέσσερα είδη ελέγχου, στα οποία εξετάζονται ξεχωριστά η ανάκτηση, η ευρωστία απέναντι σε διαφορετικές διατυπώσεις, η συμπεριφορά σε ερωτήματα εκτός πεδίου καθώς και η ποιότητα της τελικής απάντησης. Η προσέγγιση αυτή, σε συνδυασμό με τον χωρισμό του συνόλου εγγράφων σε σύνολο ανάπτυξης και σε σφραγισμένο, επέτρεψε τον λεπτομερέστερο εντοπισμό του σταδίου στο οποίο εμφανίζονται οι αστοχίες του συστήματος.

Τέλος, σε επίπεδο ευρημάτων, καταγράφηκε η κατανομή των σφαλμάτων ενός τοπικού συστήματος RAG πάνω σε θεσμικό σώμα κειμένων, με σημαντικό εύρημα την εσφαλμένη σύνδεση πραγματικών πληροφοριών με λανθασμένα αντικείμενα. Παράλληλα, οι αλλαγές που ενδεχομένως είχαν σκοπό τη βελτίωση της ανάκτησης δεν μεταφέρονται απαραίτητα στην τελική έξοδο μετά την επανακατάταξη. Αυτό το εύρημα αποδεικνύει ότι μια αλλαγή σε ένα στάδιο του συστήματος που βελτιώνει το συγκεκριμένο τμήμα δεν συνεπάγεται τη συνολική βελτιστοποίησή του.

6.3 Μελλοντικές επεκτάσεις

Βασική αδυναμία του συστήματος αποτελεί η λάθος σύνδεση των πληροφοριών για τη σύνθεση μιας απάντησης, καθώς πολλές φορές μία σωστή πληροφορία αντιστοιχίζεται με λάθος άτομο, μάθημα ή άλλο αντικείμενο. Ειδικότερα, μία μελλοντική βελτίωση θα αποτελούσε την προσθήκη ενός μηχανισμού που θα ελέγχει αν κάθε βασική πληροφορία της απάντησης αντιστοιχεί πράγματι στη σωστή πηγή και στο σωστό αντικείμενο. Μια ακόμη κατεύθυνση είναι η υποστήριξη πολλαπλών τρόπων απόκρισης. Πιο συγκεκριμένα, το σύστημα θα μπορούσε να ζητάει διευκρίνιση όταν μια ερώτηση είναι ασαφής, να δίνει μερική απάντηση όταν διαθέτει μόνο μέρος της απαιτούμενης πληροφορίας και να διορθώνει μια λανθασμένη υπόθεση που περιέχεται στην ερώτηση του χρήστη.

Παράλληλα, μια ιδιαίτερα σημαντική λειτουργία που θα μπορούσε να ενσωματωθεί είναι η υποστήριξη φωνητικών εντολών με τη μετατροπή από φωνή σε κείμενο, καθώς και η επισύναψη αρχείων από τον χρήστη, διατηρώντας παράλληλα την τοπική επεξεργασία των δεδομένων.

Τέλος, η αξιολόγηση του συστήματος θα μπορούσε να επεκταθεί με τη συμμετοχή πραγματικών χρηστών, τη συστηματική εξέταση της ορθότητας των παραπομπών, τη σύγκριση με άλλες προσεγγίσεις και τη δοκιμή του συστήματος με πολλούς ταυτόχρονους χρήστες. Με αυτόν τον τρόπο θα μπορούσε να αξιολογηθεί καλύτερα όχι μόνο η τεχνική του απόδοση, αλλά και η πραγματική χρησιμότητά του σε συνθήκες καθημερινής λειτουργίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 9459–9474.
- [2] Y. Gao et al., “Retrieval-augmented generation for large language models: A survey,” arXiv:2312.10997, 2023.
- [3] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge University Press, 2008.
- [4] E. Adamopoulou and L. Moussiades, “Chatbots: History, technology, and applications,” Machine Learning with Applications, vol. 2, art. 100006, 2020.
- [5] R. Winkler and M. Söllner, “Unleashing the potential of chatbots in education: A state-of-the-art analysis,” in Academy of Management Annual Meeting Proceedings, Chicago, IL, USA, 2018.
- [6] Ευρωπαϊκό Κοινοβούλιο και Συμβούλιο, «Κανονισμός (ΕΕ) 2016/679 για την προστασία των φυσικών προσώπων έναντι της επεξεργασίας των δεδομένων προσωπικού χαρακτήρα (Γενικός Κανονισμός για την Προστασία Δεδομένων),» Επίσημη Εφημερίδα της Ευρωπαϊκής Ένωσης, L 119, σσ. 1–88, Μάιος 2016.
- [7] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL), Berlin, Germany, 2016, pp. 1715–1725.
- [8] M. Schuster and K. Nakajima, “Japanese and Korean voice search,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Kyoto, Japan, 2012, pp. 5149–5152.
- [9] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP): System Demonstrations, Brussels, Belgium, 2018, pp. 66–71.
- [10] A. Conneau et al., “Unsupervised cross-lingual representation learning at scale,” in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2020, pp. 8440–8451.
- [11] The Unicode Consortium, The Unicode Standard, Version 15.1.0. Mountain View, CA, USA: Unicode, Inc., 2023.
- [12] M. Davis and K. Whistler, “Unicode Standard Annex #15: Unicode Normalization Forms,” The Unicode Consortium, 2023. [Online]. Available: <https://www.unicode.org/reports/tr15/>
- [13] M. F. Porter, “An algorithm for suffix stripping,” Program, vol. 14, no. 3, pp. 130–137, 1980.
- [14] M. F. Porter, “Snowball: A language for stemming algorithms,” 2001. [Online]. Available: <https://snowballstem.org/texts/introduction.html>
- [15] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, “GREEK-BERT: The Greeks visiting Sesame Street,” in Proc. 11th Hellenic Conf. Artificial Intelligence (SETN), Athens, Greece, 2020, pp. 110–117.
- [16] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.
- [17] S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.
- [18] Z. S. Harris, “Distributional structure,” Word, vol. 10, no. 2–3, pp. 146–162, 1954.
- [19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proc. Int. Conf. Learning Representations (ICLR) Workshop Track, Scottsdale, AZ, USA, 2013.
- [20] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.

- [21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.
- [22] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in Proc. EMNLP-IJCNLP, Hong Kong, China, 2019, pp. 3982–3992.
- [23] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, “BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation,” arXiv:2402.03216, 2024.
- [24] A. Vaswani et al., “Attention is all you need,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 5998–6008.
- [25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI Technical Report, 2019.
- [26] T. B. Brown et al., “Language models are few-shot learners,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877–1901.
- [27] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 2022, pp. 27730–27744.
- [28] R. Pope et al., “Efficiently scaling transformer inference,” in Proc. Machine Learning and Systems (MLSys), Miami, FL, USA, 2023.
- [29] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” in Proc. Int. Conf. Learning Representations (ICLR), 2020.
- [30] A. Fan, M. Lewis, and Y. Dauphin, “Hierarchical neural story generation,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL), Melbourne, Australia, 2018, pp. 889–898.
- [31] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, “CTRL: A conditional transformer language model for controllable generation,” arXiv:1909.05858, 2019.
- [32] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, “LLM.int8(): 8-bit matrix multiplication for transformers at scale,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2022.
- [33] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, “GPTQ: Accurate post-training quantization for generative pre-trained transformers,” in Proc. Int. Conf. Learning Representations (ICLR), 2023.
- [34] J. Lin et al., “AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration,” in Proc. Machine Learning and Systems (MLSys), Santa Clara, CA, USA, 2024.
- [35] Z. Ji et al., “Survey of hallucination in natural language generation,” ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.
- [36] L. Huang et al., “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions,” arXiv:2311.05232, 2023.
- [37] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 6769–6781.
- [38] P. Indyk and R. Motwani, “Approximate nearest neighbors: Towards removing the curse of dimensionality,” in Proc. 30th Annu. ACM Symp. Theory of Computing (STOC), Dallas, TX, USA, 1998, pp. 604–613.
- [39] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, no. 1, pp. 117–128, 2011.
- [40] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824–836, 2020.
- [41] Qdrant, “Qdrant documentation.” [Online]. Available: <https://qdrant.tech/documentation/>

- [42] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, “Reciprocal rank fusion outperforms Condorcet and individual rank learning methods,” in Proc. 32nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval, Boston, MA, USA, 2009, pp. 758–759.
- [43] R. Nogueira and K. Cho, “Passage re-ranking with BERT,” arXiv:1901.04085, 2019.
- [44] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, “C-Pack: Packed resources for general Chinese embeddings,” in Proc. 47th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, Washington, DC, USA, 2024, pp. 641–649.
- [45] E. M. Voorhees, “The TREC-8 question answering track report,” in Proc. 8th Text REtrieval Conference (TREC-8), Gaithersburg, MD, USA, 1999, pp. 77–82.
- [46] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” ACM Trans. Information Systems, vol. 20, no. 4, pp. 422–446, 2002.
- [47] N. F. Liu et al., “Lost in the middle: How language models use long contexts,” Trans. Assoc. Computational Linguistics, vol. 12, pp. 157–173, 2024.
- [48] T. Gao, H. Yen, J. Yu, and D. Chen, “Enabling large language models to generate text with citations,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Singapore, 2023, pp. 6465–6488.
- [49] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” in Proc. Int. Conf. Learning Representations (ICLR), Vienna, Austria, 2024.
- [50] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, “RAGAS: Automated evaluation of retrieval augmented generation,” in Proc. 18th Conf. European Chapter Assoc. Comput. Linguistics (EACL): System Demonstrations, St. Julians, Malta, 2024, pp. 150–158.
- [51] L. Zheng et al., “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena,” in Proc. Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track, New Orleans, LA, USA, 2023.
- [52] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: The muppets straight out of law school,” in Findings of the Assoc. Comput. Linguistics: EMNLP 2020, 2020, pp. 2898–2904.
- [53] G. van Rossum and F. L. Drake, The Python Language Reference, version 3.12. Python Software Foundation, 2023. [Online]. Available: <https://docs.python.org/3/reference/>
- [54] Microsoft, “TypeScript documentation.” [Online]. Available: <https://www.typescriptlang.org/docs/>
- [55] E. F. Codd, “A relational model of data for large shared data banks,” Communications of the ACM, vol. 13, no. 6, pp. 377–387, 1970.
- [56] Free Software Foundation, “GNU Bash reference manual,” 2022. [Online]. Available: <https://www.gnu.org/software/bash/manual/>
- [57] K. Reitz, “Requests: HTTP for humans.” [Online]. Available: <https://requests.readthedocs.io/>
- [58] L. Richardson, “Beautiful Soup documentation.” [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [59] M. Dombrowski, “markdownify: Convert HTML to Markdown.” [Online]. Available: <https://pypi.org/project/markdownify/>
- [60] Microsoft, “Playwright for Python documentation.” [Online]. Available: <https://playwright.dev/python/>
- [61] K. Reitz, “certifi: Python SSL certificates.” [Online]. Available: <https://pypi.org/project/certifi/>
- [62] C. Auer et al., “Docling technical report,” arXiv:2408.09869, 2024.

- [63] Blaizzy, “mlx-vlm: Vision language models on Apple silicon with MLX.” [Online]. Available: <https://github.com/Blaizzy/mlx-vlm>
- [64] J. Singer-Vine, “pdfplumber: Plumb a PDF for detailed information.” [Online]. Available: <https://github.com/jsvine/pdfplumber>
- [65] Artifex Software, “PyMuPDF documentation.” [Online]. Available: <https://pymupdf.readthedocs.io/>
- [66] S. Canny, “python-docx documentation.” [Online]. Available: <https://python-docx.readthedocs.io/>
- [67] W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python in Science Conf. (SciPy), Austin, TX, USA, 2010, pp. 56–61.
- [68] A. Hannun, J. Digani, A. Katharopoulos, and R. Collobert, “MLX: Efficient and flexible machine learning on Apple silicon,” 2023. [Online]. Available: <https://github.com/ml-explore/mlx>
- [69] A. Yang et al., “Qwen3 technical report,” arXiv:2505.09388, 2025.
- [70] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 2019, pp. 8024–8035.
- [71] C. R. Harris et al., “Array programming with NumPy,” Nature, vol. 585, pp. 357–362, 2020.
- [72] LangChain, “LangChain documentation.” [Online]. Available: <https://python.langchain.com/docs/>
- [73] Snowball, “snowballstemmer: Snowball stemming algorithms for Python.” [Online]. Available: <https://pypi.org/project/snowballstemmer/>
- [74] S. Ramírez, “FastAPI documentation.” [Online]. Available: <https://fastapi.tiangolo.com/>
- [75] WHATWG, “HTML Living Standard — Server-sent events.” [Online]. Available: <https://html.spec.whatwg.org/multipage/server-sent-events.html>
- [76] S. Colvin et al., “Pydantic documentation.” [Online]. Available: <https://docs.pydantic.dev/>
- [77] Vercel, “Next.js documentation.” [Online]. Available: <https://nextjs.org/docs>
- [78] Meta Open Source, “React documentation.” [Online]. Available: <https://react.dev/>
- [79] World Wide Web Consortium, “HTML Standard and Cascading Style Sheets (CSS) specifications.” [Online]. Available: <https://www.w3.org/standards/>
- [80] The PostgreSQL Global Development Group, “PostgreSQL 16 documentation,” 2023. [Online]. Available: <https://www.postgresql.org/docs/16/>
- [81] M. Bayer, “SQLAlchemy documentation.” [Online]. Available: <https://docs.sqlalchemy.org/>
- [82] SQLite Consortium, “Write-ahead logging (WAL).” [Online]. Available: <https://www.sqlite.org/wal.html>
- [83] D. Merkel, “Docker: Lightweight Linux containers for consistent development and deployment,” Linux Journal, vol. 2014, no. 239, art. 2, 2014.
- [84] Apple Inc., “Daemons and services programming guide.” [Online]. Available: <https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPSystemStartup/>
- [85] H. Krekel et al., “pytest documentation.” [Online]. Available: <https://docs.pytest.org/>
- [86] S. Chacon and B. Straub, Pro Git, 2nd ed. New York, NY, USA: Apress, 2014.
- [87] D. Delbecque, “Loguru: Python logging made simple.” [Online]. Available: <https://loguru.readthedocs.io/>

ΠΑΡΑΡΤΗΜΑΤΑ

Παράρτημα Α: Ρυθμίσεις, παράμετροι και εκδόσεις

Στο παράρτημα αυτό συγκεντρώνονται οι παράμετροι λειτουργίας του συστήματος, οι εκδόσεις των βιβλιοθηκών που χρησιμοποιήθηκαν και η μεταβολή του μεγέθους της συλλογής κατά την περίοδο των μετρήσεων. Οι τιμές αντιστοιχούν στην κατάσταση του αποθετηρίου κατά την ολοκλήρωση της εργασίας.

Πίνακας Α.1: Σταθερές που κατονομάζονται στα Κεφάλαια 4 και 5

Σταθερά	Ενεργή τιμή	Αρχείο:γραμμή	Ρόλος
RETRIEVAL_MIN_TOP_SCORE	0,25	config.py:143	Πύλη απάντησης· κάτω από την τιμή αυτή επιστρέφεται άρνηση χωρίς κλήση του γλωσσικού μοντέλου
REWRITE_TRIGGER_HI	0,28	config.py:704	Κατώφλι ενεργοποίησης της αναδιατύπωσης σε ακαδημαϊκό ύφος
QDRANT_HNSW_EF	512	config.py:84 και .env	Εύρος αναζήτησης HNSW· η προεπιλογή 0 συνεπάγεται την τιμή 128 του διακομιστή
QDRANT_INDEXING_THRESHOLD	1000	config.py:93	Χαμηλότερο από την προεπιλογή 20000, ώστε να κατασκευάζεται ευρετήριο HNSW στο μέγεθος της συλλογής
AGENTIC_ADAPTIVE_GATE	True	config.py:747	Προσαρμοστική πύλη της πρακτορικής διαδρομής
ENABLE_AGENTIC_RAG	True	config.py:736	Αποσύνθεση πολυμερούς ερωτήματος
AGGREGATION_MAX_CHUNKS	24	config.py:310	Ανώτατο πλήθος τμημάτων στη συνάθροιση ωρολογίου
THRESHOLD_NAMES	0,20	src/utis/fuzzy_snap.py:48	Σχετική ως προς το μήκος απόσταση επεξεργασίας για ονόματα· αυστηρότερο από το THRESHOLD_WORDS = 0,30

Πίνακας Α.2: Πλήθος αποτελεσμάτων ανά στάδιο της ανάκτησης

Στάδιο	Τιμή	Αρχείο:γραμμή
Μέγεθος δεξαμενής (candidate_k)	40· διπλασιάζεται σε χαλαρή επανάληψη	retriever.py:831–833
Πυκνό σκέλος (Qdrant)	40	retriever.py:467

Αραιό σκέλος (BM25)	40	retriever.py:496
Μετά τη σύντηξη RRF	40	retriever.py:925
Είσοδος επανακατατάκτη	Όλοι οι υποψήφιοι	retriever.py:1043
Μετά την επανακατάταξη	6 (παραγωγή) / 8 (αξιολόγηση)	config.py:278 / config.py:132
Βάρη σύντηξης	Πυκνό 0,6 · αραιό 0,4 · K = 60	retriever.py:74–78
Ελάχιστη πυκνή ομοιότητα	0,15	config.py:133
Κατώφλι εμπιστοσύνης επανακατατάκτη	0,15	config.py:273

Πίνακας Α.3: Στρατηγική τεμαχισμού και επικάλυψη ανά τύπο εγγράφου

Τύπος εγγράφου	Στρατηγική	Μέγεθος / επικάλυψη	Αρχείο:γραμμή
announcement	Ολόκληρο, εφόσον χωρά	≤ 2000 χαρ., αλλιώς αναδρομικός χωρισμός	qdrant_indexer.py:66, 479
regulation, curriculum, page, calendar	MarkdownHeader και έπειτα αναδρομικός	1500 / 200	qdrant_indexer.py:625–626
Γενική διαδρομή	Αναδρομικός	1200 / 200	qdrant_indexer.py:62–63
law	Χωρισμός ανά άρθρο, έπειτα αναδρομικός	900 / 100	ministry_crawler.py:586
Ιστοσελίδες υπουργείου και ιδρύματος	Αναδρομικός ανά πηγή	700 / 80 ή 600 / 60	ministry_crawler.py:137–266
Έγγραφα-κατάλογοι	Ένα τμήμα ανά εγγραφή	Όριο η ίδια η εγγραφή	qdrant_indexer.py:544–553
Ωρολόγιο (JSON)	Ένα τμήμα ανά μάθημα και ομάδα	Χωρίς συγχώνευση ομάδων	qdrant_indexer.py:1003
Όριο ασφαλείας	Επανα-χωρισμός τμημάτων άνω των 2250 χαρ.	1500 / 200	qdrant_indexer.py:752

Πίνακας Α.4: Παράμετροι αποκωδικοποίησης του γλωσσικού μοντέλου

Παράμετρος	Ενεργή τιμή	Αρχείο:γραμμή
LLM_MODEL	mlx-community/Qwen3-8B-4bit	config.py:313
LLM_TEMPERATURE	0,0 (άπληστη αποκωδικοποίηση)	config.py:339
LLM_REPETITION_PENALTY	1,05	config.py:344
LLM_MAX_TOKENS	2048	config.py:334
LLM_CONTEXT_WINDOW	32768	config.py:345
LLM_THINKING_MODE	False	config.py:333
LLM_DRAFT_MODEL	Κενό· η κερδοσκοπική αποκωδικοποίηση είναι ανενεργή	config.py:331
LLM_HISTORY_TURNS / LLM_MAX_HISTORY_CHARS	4 γύροι / 1000 χαρ. ανά μήνυμα	config.py:350–351
USE_COMPACT_TEMPLATE / ENABLE_PREFIX_CACHE	True / True	config.py:359, 378

Πίνακας Α.5: Εκδόσεις των βιβλιοθηκών που αναφέρονται στο Κεφάλαιο 3

Βιβλιοθήκη	Έκδοση	Βιβλιοθήκη	Έκδοση
requests	2.32.5	pdfplumber	0.11.9
beautifulsoup4	4.14.3	PyMuPDF	1.27.1
markdownify	1.2.2	python-docx	1.2.0
playwright	1.58.0	sentence-transformers	5.2.3
certifi	2026.1.4	qdrant-client	1.17.0
docling	2.96.0	snowballstemmer	3.1.1
mlx	0.30.6	fastapi	0.135.1
mlx-vlm	0.4.1	uvicorn	0.41.0
mlx-lm	0.31.1	pydantic	2.12.5
torch	2.11.0	SQLAlchemy	2.0.46
numpy	2.4.2	psycpg	3.3.4
pandas	3.0.5	pytest	9.0.3
langchain-text-splitters	1.1.1	loguru	0.7.3
next	16.2.10	react / react-dom	19.2.4
typescript	^5		

Πίνακας Α.6: Πλατφόρμα και εργαλεία υποδομής

Στοιχείο	Έκδοση	Στοιχείο	Έκδοση
Python	3.12.4	Docker	29.6.2
PostgreSQL	16.14	Docker Compose	5.3.1
SQLite	3.45.3	Git	2.53.0
Διακομιστής Qdrant	1.17.1		

Πίνακας Α.7: Μεταβολή του μεγέθους της συλλογής κατά την περίοδο των μετρήσεων

Ημερομηνία	Σημεία	Γεγονός
29/07/2026	8.524 → 8.599	Νυχτερινή συλλογή
10/08 – 14/08/2026	8.677	Σταθερό· μέγεθος κατά τη μέτρηση του τεμαχισμού ανά εγγραφή
16/08 – 21/08/2026	8.697	Αύξηση κατά 20 από την ίδια την παρέμβαση
22/08/2026	8.697 → 11.090	Ευρετηρίαση 3.032 τμημάτων από 2.054 αρχεία, εκ των οποίων 1.568 ιστορικά
23/08/2026	11.090	Μέγεθος κατά την επανεξέταση της πόλης ανάκτησης
24/08/2026	11.090 → 11.217	Νυχτερινή συλλογή
25/08/2026	11.217	Τρέχον μέγεθος του ευρετηρίου

Παράρτημα Β: Το ντετερμινιστικό επίπεδο και η πύλη εισόδου

Παρατίθεται ο κώδικας του ντετερμινιστικού επιπέδου γεγονότων και της πύλης εισόδου, που περιγράφονται στις Υποενότητες 4.3.2 και 4.3.3.

Παράθεμα Β.1: Ο πίνακας χειριστών του ντετερμινιστικού επιπέδου και η λογική πρώτης επιτυχίας.

Πηγή: src/chatbot/fact_layer.py, γραμμές 70–106

```
_HANDLERS: list[tuple[Callable[[], bool], FactHandler]] = [
    (_scope_gate_enabled,    try_out_of_scope_answer),
    (_staff_status_enabled,  try_staff_status_answer),
    (_staff_contact_enabled, try_staff_contact_answer),
    (_structured_enabled,    try_structured_answer),
]

def try_fact_answer(query: str) -> StructuredOutcome | None:
    for is_enabled, handler in _HANDLERS:
        if not is_enabled():
            continue
        outcome = handler(query)
        if outcome is not None:
            return outcome
    return None
```

Παράθεμα Β.2: Ο μηχανικά αναγνώσιμος λόγος άρνησης για ερωτήματα εκτός εμβέλειας.

Πηγή: src/chatbot/structured_path.py, γραμμές 777–796

```
def out_of_scope_reason(query: str) -> str | None:
    if not query:
        return None
    folded = strip_accents(query)
    if any(m in folded for m in _OURS_MARKERS):
        return None
    for name, pattern in _FOREIGN_DEPT_PATTERNS:
        if pattern.search(folded):
            return f"foreign_department:{name}"
    if any(n in folded for n in _FACILITY_NOUNS):
        for name, pattern in _FOREIGN_CAMPUS_PATTERNS:
            if pattern.search(folded):
                return f"foreign_campus:{name}"
    return None
```

Παράρτημα Γ: Ανάκτηση, σύντηξη και επανακατάταξη

Παρατίθεται ο κώδικας της δρομολόγησης πρόθεσης, της σύντηξης των δύο σκελών ανάκτησης και της επανακατάταξης, που περιγράφονται στις Υποενότητες 4.3.6 και 4.3.7.

Παράθεμα Γ.1: Η δρομολόγηση πρόθεσης δύο επιπέδων: υπερισχύον κανονικό πρότυπο με βέτο και, εφόσον δεν ενεργοποιηθεί, ομοιότητα συνημιτόνου με τα πρότυπα διανύσματα.

Πηγή: src/chatbot/intent_router.py, γραμμές 353–421 (αποσπασματικά)

```
def classify_intent(
    query: str,
    query_vector: list[float] | np.ndarray | None = None,
    embedder: "Embedder | None" = None,
) -> IntentResult:
    result = IntentResult()
    q = query.strip()
    q_fold = strip_accents(q)

    for rule_name, pattern, folded, guard in _REGEX_OVERRIDES:
        subject = q_fold if folded else q
        if pattern.search(subject):
            if guard is not None and guard.search(subject):
                continue
            rule = _RULE_BY_NAME[rule_name]
            result.matched_intent = rule_name
            result.doc_types_exclusive = list(rule["exclusive"])
            result.schedule_kind_exclusive =
rule.get("exclusive_schedule_kind")
            return result

    if query_vector is None and embedder is None:
        return result
    scores = {name: _cosine(qv, p) for name, p in protos.items()}
    best = max(scores, key=scores.get)
    result.matched_intent = best
    result.schedule_kind_soft =
_RULE_BY_NAME[best].get("exclusive_schedule_kind")
    return result
```

Παράθεμα Γ.2: Η σύντηξη αντίστροφης κατάταξης πάνω σε σταθμισμένες λίστες, με κανόνα ισοπαλίας υπέρ της λίστας που εισήγαγε πρώτη το έγγραφο.

Πηγή: src/chatbot/retriever.py, γραμμές 72–78 και 529–557

```
_DENSE_WEIGHT = 0.6
_SPARSE_WEIGHT = 0.4
_RRF_K = 60

def _rrf_fuse(
    self,
    weighted_lists: list[tuple[list[dict], float]],
    top_k: int,
) -> list[dict]:
    rrf_scores: dict[str, float] = {}
    by_qid: dict[str, dict] = {}

    for docs, weight in weighted_lists:
        for rank, doc in enumerate(docs):
            qid = doc["_qid"]
```

```

        rrf_scores[qid] = rrf_scores.get(qid, 0.0) + weight /
(_RRF_K + rank + 1)
        by_qid.setdefault(qid, doc)

    ranked = sorted(rrf_scores.items(), key=lambda x: x[1],
reverse=True)[:top_k]
    return [
        {**by_qid[qid], "score": round(rrf_score, 6)}
        for qid, rrf_score in ranked
    ]

```

Παράθεμα Γ.3: Η πολιτική εγκυρότητας και το σημείο εφαρμογής του επανακατατάκτη: η ληγμένη προθεσμία αφαιρείται πριν την επανακατάταξη, ενώ οι ποινές εφαρμόζονται μετά.

Πηγή: src/chatbot/retriever.py, μέθοδος search(), γραμμές 954–1072 (αποσπασματικά)

```

_deadline_query = asks_deadline(query)

def _is_expired_deadline(c: dict) -> bool:
    if not (settings.USE_EVENT_VALIDITY and _deadline_query):
        return False
    meta = c.get("metadata", {})
    if meta.get("doc_type") != "announcement":
        return False
    return _valid_to_expired(meta, date.today())

if settings.EVENT_VALIDITY_DROP:
    candidates = [c for c in candidates if not
_is_expired_deadline(c)]

from src.chatbot.reranker import BGEReranker
reranker = BGEReranker.get()
if reranker is not None:
    reranked, failed = reranker.rerank(query, candidates,
top_k=len(candidates))
    results = [] if failed else reranked
else:
    results = candidates

if results:
    prefer = (set(intent.boost_doc_types)
              if not intent.doc_types_exclusive else set())
    for c in results:
        base = c.get("rerank_score", c.get("score", 0.0))
        if _is_hist_noise(c) or _is_expired_deadline(c):
            base += settings.HISTORICAL_PENALTY
        elif reranker is not None and c.get("metadata",
{}).get("doc_type") in prefer:
            base += settings.INTENT_MATCH_BOOST
        c["rerank_score"] = base
    results.sort(key=lambda c: c.get("rerank_score", 0.0),
reverse=True)
    results = results[:top_k]

```

Παράρτημα Δ: Παραγωγή της απάντησης και έλεγχοι

Παρατίθεται ο κώδικας που αφορά τη διαμόρφωση της προτροπής, την πύλη απάντησης και τους ελέγχους που εκτελούνται μετά την παραγωγή, όπως περιγράφονται στην Ενότητα 4.5.

Παράθεμα Δ.1: Ο σκελετός της προτροπής: σταθερό πρόθεμα, μεταβλητό μέρος με ημερομηνία και συμφραζόμενα, και η συναρμολόγηση των μηνυμάτων.

Πηγή: chatbot_mlx.py, γραμμές 205–214 και 566–603 (αποσπασματικά)

```
def _build_prompt(
    self,
    query: str,
    chunks: list[dict],
    history: list[Message],
    schedule_prefix: str = "",
) -> str:
    context = self._format_context(chunks)
    if schedule_prefix:
        context = schedule_prefix + context
    template = (IEE_SYSTEM_TEMPLATE_COMPACT
                 if settings.USE_COMPACT_TEMPLATE else
                 IEE_SYSTEM_TEMPLATE)
    system_block = template.format(
        context=context,
        current_time=_format_greek_datetime(datetime.now()),
    )

    messages = [{"role": "system", "content": system_block}]
    for msg in history:
        messages.append({"role": msg.role, "content": msg.content})
    messages.append({"role": "user", "content": query})

    return self.tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=settings.LLM_THINKING_MODE,
    )
```

Παράθεμα Δ.2: Ο έλεγχος παραπομπών: κάθε σύνδεσμος της απάντησης πρέπει να εμφανίζεται αυτούσιος σε ανακτημένο τμήμα, αλλιώς αντικαθίσταται και καταγράφεται ως επινοημένος.

Πηγή: src/chatbot/citation_validator.py, γραμμές 316–353

```
def _normalize(url: str) -> str:
    return url.strip().rstrip("/")

def validate_citations(answer: str, chunks: list[dict]) -> tuple[str,
list[str]]:
    valid_urls: set[str] = set()
    for chunk in chunks:
        meta = chunk.get("metadata", {}) if isinstance(chunk, dict) else
        chunk.payload
        if url := meta.get("url", ""):
            valid_urls.add(_normalize(url))
        text = chunk.get("text", "") if isinstance(chunk, dict) else
        chunk.payload.get("text", "")
```

```

for u in _URL_RE.findall(text):
    valid_urls.add(_normalize(u))

hallucinated: list[str] = []

def _check_url(match: re.Match) -> str:
    url = match.group(0).rstrip(".,;:!?")
    if _normalize(url) in valid_urls:
        return url
    hallucinated.append(url)
    return "[πηγή μη επαληθεύσιμη]"

return _URL_RE.sub(_check_url, answer), hallucinated

```

Παράθεμα Δ.3: Η πύλη απάντησης: το γλωσσικό μοντέλο δεν καλείται όταν η δεξαμενή είναι κενή ή η κορυφαία βαθμολογία υπολείπεται του κατωφλίου.

Πηγή: chatbot_mlx.py, γραμμές 1357–1374

```

_top_score: float | None = None
if retrieval.context_chunks:
    _top = retrieval.context_chunks[0]
    _top_score = _top.get("rerank_score", _top.get("score"))

if (not retrieval.context_chunks
    or (_top_score is not None
        and _top_score < settings.RETRIEVAL_MIN_TOP_SCORE)):
    refusal = ("Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να
"
              "απαντήσω με σιγουριά.")
    _refusal_reason = (
        "empty_pool" if not retrieval.context_chunks else
"min_top_score"
    )

```

Παράρτημα Ε: Ευρετηρίαση

Παρατίθεται ο ορισμός των τύπων εγγράφου και η δρομολόγηση του τεμαχισμού, που περιγράφονται στις Υποενότητες 4.2.3 και 4.2.4.

Παράθεμα Ε.1: Το κλειστό σύνολο τύπων εγγράφου και το υποπεδίο που διακρίνει την εξεταστική από τη διδασκαλία.

Πηγή: src/ingestion/schema.py, γραμμές 28–51

```
DOC_TYPES: frozenset[str] = frozenset({
    "schedule",
    "curriculum",
    "announcement",
    "announcement_attachment",
    "regulation",
    "law",
    "guide",
    "staff",
    "official_document",
    "contact",
    "infrastructure",
    "lab",
    "general",
    "espa",
    "iky",
})

SCHEDULE_KINDS: frozenset[str] = frozenset({"exam", "teaching"})
```

Παράθεμα Ε.2: Η δρομολόγηση της στρατηγικής τεμαχισμού με βάση τον τύπο του εγγράφου και, δευτερευόντως, την πηγή προέλευσης.

Πηγή: qdrant_indexer.py, γραμμές 766–815 (αποσπασματικά)

```
def chunks_from_txt(
    path: Path,
    source_name: str,
    done_dir: Path,
) -> list[Chunk]:
    meta, body = parse_txt_file(path)
    if meta is None or body is None:
        return []

    doc_type = meta.get("DOC_TYPE", "").lower()

    if source_name == "announcements" or doc_type == "announcement":
        return _chunks_from_announcement(path, meta, body, source_name,
done_dir)

    if source_name in ("pages", "calendar", "curated") or doc_type in (
        "regulation", "curriculum", "page", "calendar"):
        return _chunks_from_markdown_doc(path, meta, body, source_name,
done_dir)

    cleaned = _clean_body(body)
    segments = [_build_enriched_text(meta, seg) for seg in
_split_with_recursive(cleaned)]
    return _make_chunks(segments, meta, path, source_name, done_dir)
```

Παράρτημα ΣΤ: Ελληνικός λεξικός ανακτητής

Παρατίθεται η αλυσίδα κανονικοποίησης του αραιού σκέλους και ο τρόπος με τον οποίο δίνεται στον λεξικό ανακτητή, καθώς και η βαθμολόγηση του σκέλους αυτού.

Παράθεμα ΣΤ.1: Η αλυσίδα κανονικοποίησης του αραιού σκέλους και το σημείο εφαρμογής της.

Πηγή: src/chatbot/greek_bm25.py και src/chatbot/retriever.py

```
_STEMMER = snowballstemmer.stemmer("greek")
_TOKEN_RE = re.compile(r"\w+", re.UNICODE)

def _stem(tok: str) -> str:
    return _STEMMER.stemWord(tok).replace("ς", "σ")

def greek_bm25_preprocess(text: str, *, use_stopwords: bool = False) -> list[str]:
    if not text:
        return []
    lowered = normalize_greek(text).lower()
    out: list[str] = []
    for tok in _TOKEN_RE.findall(lowered):
        if use_stopwords and tok in _GREEK_STOPWORDS:
            continue
        out.append(_stem(tok))
    return out

@staticmethod
def _build_bm25(docs: list[dict]):
    from langchain_community.retrievers import BM25Retriever
    from langchain_core.documents import Document
    if not docs:
        return None
    lc_docs = [
        Document(
            page_content=normalize_greek(d["text"]),
            metadata={"_qid": d["_qid"], **d["payload"]},
        )
        for d in docs
    ]
    if settings.BM25_GREEK_PREPROCESS:
        from src.chatbot.greek_bm25 import make_preprocess
        return BM25Retriever.from_documents(
            lc_docs,
            preprocess_func=make_preprocess(settings.BM25_GREEK_STOPWORDS),
        )
    return BM25Retriever.from_documents(lc_docs)
```

Παράθεμα ΣΤ.2: Η βαθμολόγηση του αραιού σκέλους, το οποίο συνεισφέρει στη σύντηξη με την κατάταξή του.

Πηγή: src/chatbot/retriever.py, γραμμές 485-507

```
def _sparse_search(
    self,
    query: str,
    academic_year: str | None,
    candidate_k: int,
    bundle: CorpusBundle,
) -> list[dict]:
    bm25 = self._get_bm25(academic_year, bundle)
    if bm25 is None:
```

```

        return []
bm25.k = candidate_k
lc_docs = bm25.invoke(query)
return [
    {
        "_qid":      doc.metadata.get("_qid", ""),
        "text":      doc.page_content,
        "score":     0.0,
        "metadata": {k: v for k, v in doc.metadata.items()
                     if k not in ("_qid", "text")},
    }
    for doc in lc_docs
]

```

Παράρτημα Z: Κανονικοποίηση ελληνικού ερωτήματος

Παρατίθεται η κατασκευή του λεξικού φωνηεντικού σκελετού από το σώμα κειμένων και η διαδικασία προσκόλλησης λεκτικών μονάδων.

Παράθεμα Z.1: Το λεξικό φωνηεντικού σκελετού και η προσκόλληση διακριτικού.

Πηγή: src/utils/text.py και src/utils/fuzzy_snap.py

```
def _greek_skeleton(word: str) -> str:
    w = strip_accents(word)
    for a, b in (("ou", "u"), ("ει", "i"), ("οι", "i"),
                  ("υι", "i"), ("αι", "e")):
        w = w.replace(a, b)
    return "".join(out)

def build_greeklsh_lexicon(texts) -> dict[str, str]:
    from collections import Counter
    freq: dict[str, Counter] = {}
    for t in texts:
        if not t:
            continue
        for m in _GREEK_WORD_RE.findall(t):
            surface = normalize_greek(m).lower()
            sk = _greek_skeleton(surface)
            freq.setdefault(sk, Counter())[surface] += 1
    return {sk: c.most_common(1)[0][0] for sk, c in freq.items()}

def snap_token(self, token: str) -> tuple[str, SnapInfo | None]:
    if len(token) < MIN_LEN:
        return token, None
    if token.lower() in self._vocab_lower:
        return token, None
    folded = _fold(token)
    if len(folded) < MIN_LEN:
        return token, None
    if folded in self._gen_surface:
        target = self._gen_surface[folded].most_common(1)[0][0]
        if _fold(target) == folded and target.lower() != token.lower():
            info = SnapInfo(original=token, target=target, distance=0,
                             tree="fold", relative_distance=0.0)
            return target, info
    return token, None
hit = self._best(self.names_tree, self._name_freq,
                  self._name_surface, folded, THRESHOLD_NAMES)
if hit is None:
    if settings.FUZZY_SNAP_NAMES_ONLY:
        return token, None
if hit is None:
    return token, None
target, dist, rel = hit
info = SnapInfo(original=token, target=target, distance=dist,
                  tree="names", relative_distance=round(rel, 4))
return target, info
```

Παράρτημα Η: Το σύνολο ερωτημάτων αξιολόγησης

Παρατίθεται το σύνολο των βασικών ερωτημάτων με τα οποία αξιολογήθηκε το σύστημα, μαζί με την αναμενόμενη απάντηση, την απάντηση που παρήγαγε πράγματι το σύστημα και το έγγραφο αναφοράς. Η παράθεση γίνεται ώστε η αξιολόγηση να είναι αναπαραγώγιμη και ελέγξιμη. Σε δεκαοκτώ ερωτήματα η αναμενόμενη συμπεριφορά είναι η άρνηση, καθώς η ζητούμενη πληροφορία δεν υπάρχει στο σώμα κειμένων. Οι παραλλαγές διατύπωσης των οργάνων Β και Δ, οι οποίες παράγονται από τα ερωτήματα αυτά, δεν παρατίθενται λόγω έκτασης.

Οι απαντήσεις του συστήματος προέρχονται από την εκτέλεση της 6ης Ιουλίου 2026, σε προγενέστερο στιγμιότυπο της συλλογής και πριν από τις παρεμβάσεις της Ενότητας 5.5. Για τα επτά ερωτήματα εξεταστικής, τα οποία προστέθηκαν μεταγενέστερα, δεν υπάρχει καταγεγραμμένη απάντηση.

Πίνακας Η.1: Τα βασικά ερωτήματα αξιολόγησης, η αναμενόμενη και η πραγματική απάντηση

Κωδικός	Ερώτημα	Αναμενόμενη απάντηση	Απάντηση συστήματος	Έγγραφο
sch_001	Πότε έχει θεωρία τα Μαθηματικά ΙΙ για την Ομάδα 1η (Β1);	Τετάρτη 14-16 (ΑμφΠ) και Παρασκευή 9-11 (ΑμφΗ).	Η θεωρία των Μαθηματικών ΙΙ για την Ομάδα 1η (Β1): Τετάρτη 14-16, αίθουσα ΑμφΠ· Παρασκευή 9-11, αίθουσα ΑμφΗ. [SOURCE: schedule]	schedule_output.json
sch_002	Ποιες μέρες και ώρες έχει θεωρία η Σχεδίαση Ψηφιακών Συστημάτων για την Ομάδα 2η;	Δευτέρα 9-11 και Τετάρτη 9-11.	Δευτέρα 9-11· Τετάρτη 9-11· Τετάρτη 14-16 (Εργ.)· Πέμπτη 11-13 (Εργ.)· Παρασκευή 14-16 (Εργ.). Η θεωρία διεξάγεται Δευτέρα και Τετάρτη 9-11.	schedule_output.json
sch_003	Πότε είναι το εργαστήριο Αντικειμενοστρεφής Προγραμματισμός για ομάδα τ5/τ6 Ομάδα 1η;	Δευτέρα 14-16, αίθουσα 210.	Τρίτη 11-13, αίθουσα 210· Πέμπτη 9-11, αίθουσα Δ1.	schedule_output.json
sch_004	Σε ποια αίθουσα γίνεται η θεωρία των Μαθηματικών ΙΙ για Ομάδα 2η την Παρασκευή;	Αίθουσα ΑμφΗ, ώρα 11-13.	Η θεωρία των Μαθηματικών ΙΙ για την Ομάδα 2η διεξάγεται στην αίθουσα ΑμφΗ την Παρασκευή.	schedule_output.json
sch_005	Ποιος διδάσκει τα Μαθηματικά ΙΙ;	Αντωνίου/Τζέκης.	Ευστάθιος Αντωνίου, συντονιστής του μαθήματος «Μαθηματικά ΙΙ» (κωδ. 1201).	schedule_output.json
sch_006	Πότε έχω εργαστήριο Μετρήσεις και Κυκλώματα Εναλλασσόμενου Ρεύματος για υποομάδα τ5 Ομάδα 1η;	Τρίτη 11-13, αίθουσα Γ1.	Τρίτη 11-13, αίθουσα Γ1.	schedule_output.json
sch_007	Πότε έχει θεωρία η Τεχνητή Νοημοσύνη;	Δευτέρα 11-13 και Τετάρτη 11-13 (κωδ. 1601).	Δευτέρα 11-13 και Τετάρτη 11-13· αίθουσα Β2/Α5·	schedule_output.json

			διδάσκοντες Κράβ./Μπράτ.	
sch_008	Πότε γίνεται το εργαστήριο Ενσωματωμένα Συστήματα για υποομάδα τ3;	Πέμπτη 9-11, αίθουσα Γ4.	Τρίτη 9-11, αίθουσα Γ4, διδάσκοντες Γιακουμής Α., Χαραλαμπίδης Χ.	schedule_output.json
sch_009	Ποιες ώρες έχει θεωρία η Ασφάλεια Πληροφοριακών Συστημάτων;	Δευτέρα 16-18 και Πέμπτη 16-18 (κωδ. 1801).	Δευτέρα 16-18 και Πέμπτη 16-18 διδασκων Ρέκκας Β.: αίθουσα 101.	schedule_output.json
sch_010	Πότε έχει μάθημα ο Αντικειμενοστρεφής Προγραμματισμός για Ομάδα 1η Τρίτη;	Τρίτη 9-11 στο ΑμφΗ.	Τρίτη 9-11 θεωρία (ΑμφΗ). Τρίτη 11-13 και 14-16 εργαστήριο (210).	schedule_output.json
sch_011	Πότε γίνεται η θεωρία Διοίκηση Έργων;	Πέμπτη 14-16 και Παρασκευή 14-16.	Πέμπτη 14-16 και Παρασκευή 14-16, αίθουσα Α. 101, διδασκων Καραμητόπουλος.	schedule_output.json
sch_012	Πότε έχουμε Αριθμητικές Μέθοδοι;	Δευτέρα 16-18 και Πέμπτη 9-11 (κωδ. 1641).	Δευτέρα 16-18 και Πέμπτη 9-11, αίθουσα Β2, διδασκων Τζέκης Π.	schedule_output.json
sch_013	Πότε διεξάγεται το εργαστήριο Μετατροπείς Ισχύος;	Τρίτη 11-13, αίθουσα Α1.	Τρίτη 11-13, αίθουσα Α1.	schedule_output.json
sch_014	Πότε έχω μάθημα Αρχές και Μέθοδοι Μηχανικής Μάθησης;	Δευτέρα 14-16 και Τρίτη 14-16 (κωδ. 1802).	Δευτέρα 14-16 και Τρίτη 14-16, αίθουσα Β4, διδασκων Τσιντώτας Κ.	schedule_output.json
sch_015	Σε ποια αίθουσα γίνεται η θεωρία Τεχνική Συγγραφή Ομάδα 1η Τετάρτη;	Τετάρτη 16-18 στο ΑμφΗ.	Στην αίθουσα ΑμφΗ, Τετάρτη.	schedule_output.json
page_001	Τι διδάσκει ο καθηγητής Δεληγιάννης Ιγνάτιος;	Αναμενόμενη άρνηση.	Ο καθηγητής Δεληγιάννης Ιγνάτιος διδάσκει «Μηχανική Λογισμικού» και «Αντικειμενοστρεφή Προγραμματισμό».	—
page_002	Τι αφορά η Αλληλεπίδραση Ανθρώπου-Μηχανής;	Σχεδιασμός διεπαφών και επικοινωνία ανθρώπου-υπολογιστή.	Αφορά τη σχεδίαση, υλοποίηση και αξιολόγηση λειτουργικών και εύχρηστων διαδραστικών συστημάτων υπολογιστών.	Αλληλεπίδραση_Ανθρώπου-Μηχανής.txt
page_003	Τι πληροφορίες υπάρχουν για το Erasmus στο ΙΕΕ;	Συμμετοχή στο Erasmus+ για φοιτητικές ανταλλαγές.	Το Τμήμα έχει υπογράψει διμερείς συμφωνίες συνεργασίας με πανεπιστήμια σε όλη την Ευρώπη και άλλες περιοχές.	Erasmus.txt
page_004	Ποια χρήσιμα έντυπα υπάρχουν για τους φοιτητές;	Αιτήσεις, δηλώσεις μαθημάτων, αιτήσεις βεβαιώσεων.	Παρατίθεται η λίστα των εντύπων: αίτηση εγγραφής πρωτοετών, υπεύθυνη δήλωση κ.ά., με συνδέσμους.	Χρήσιμα_Έντυπα_για_φοιτητές.txt
page_005	Τι είναι το Ενσωματωμένα Συστήματα ως αντικείμενο έρευνας;	Σχεδίαση υπολογιστικών συστημάτων	Ανάπτυξη εφαρμογών με χρήση του μικροελεγκτή PIC 18F4550, με	Ενσωματωμένα Συστήματα.txt, Ερευνητικά_Εργαστήρια.txt

		ενσωματωμένων σε συσκευές.	προγραμματισμό σε γλώσσα C.	
page_006	Ποιες υπηρεσίες υπάρχουν για τους φοιτητές στο τμήμα;	Υπηρεσίες μέσω γραμματείας και ιστοσελίδας.	Διαδικτυακός τόπος τμήματος, ηλεκτρονικές ανακοινώσεις και λοιπές υπηρεσίες προς τους φοιτητές.	Υπηρεσίες_Τμήματος_οδηγίες.txt
page_007	Τι αφορά το μάθημα Επεξεργασία Σήματος;	Ανάλυση και επεξεργασία σημάτων, φίλτρα, μετασχηματισμοί.	Εισαγωγή στις έννοιες και την ανάλυση αναλογικών και διακριτών σημάτων και συστημάτων.	Επεξεργασία_Σήματος.txt
page_008	Τι είναι το Διαδίκτυο των Πραγμάτων;	Διασύνδεση φυσικών συσκευών μέσω διαδικτύου.	Τεχνολογία που συνδέει φυσικά αντικείμενα μέσω διαδικτύου για επικοινωνία και ανταλλαγή δεδομένων.	Διαδίκτυο_των_Πραγμάτων.txt
page_009	Ποιες υποδομές και υπηρεσίες διαθέτει το τμήμα;	Εργαστηριακές υποδομές και ψηφιακές υπηρεσίες.	Διαδικτυακός τόπος, ανακοινώσεις, εργαστηριακές υποδομές και υπηρεσίες υποστήριξης.	Υποδομές_Υπηρεσίες.txt
page_010	Τι αφορά ο σκοπός του τμήματος ΙΕΕ;	Εκπαίδευση μηχανικών σε πληροφορική και ηλεκτρονικά συστήματα.	Παροχή θεωρητικών και πρακτικών γνώσεων και εκπαίδευση μηχανικών στην Πληροφορική και τα Ηλεκτρονικά Συστήματα.	Σκοπός.txt
reg_001	Πόσα ECTS χρειάζονται για την απόκτηση πτυχίου;	300 ECTS.	Απαιτούνται συνολικά 300 πιστωτικές μονάδες (ECTS), συμπεριλαμβανομένης της διπλωματικής εργασίας.	Κανονισμός σπουδών v3, Συνήθεις Ερωτήσεις, EN-Study-Guide
reg_002	Ποιες είναι οι προϋποθέσεις για συμμετοχή εξετάσεις;	Εγγραφή στο μάθημα και ελάχιστη παρακολούθηση.	Ηλεκτρονική δήλωση του μαθήματος στη γραμματεία και, όπου απαιτείται, δήλωση στον διδάσκοντα.	Κανονισμός-εξετάσεων_ΤΜΠΗΣ_v2.pdf
reg_003	Τι ισχύει για τις διπλωματικές εργασίες;	Διαδικασίες ανάθεσης, εκπόνησης και εξέτασης.	Ελάχιστη διάρκεια ένα ακαδημαϊκό εξάμηνο, μέγιστη δύο έτη από την ανάθεση· μετά τη διετία ακυρώνεται.	Κανονισμός-Διπλωματικών-Εργασιών-R1.pdf
reg_004	Ποιος είναι ο κανονισμός Erasmus για κινητικότητα φοιτητών;	Διαδικασίες αίτησης, επιλογής και αναγνώρισης σπουδών.	Ο Κανονισμός Κινητικότητας Erasmus διέπει διαδικασίες, προϋποθέσεις, επιλογή, αναγνώριση και επιχορήγηση.	Κανονισμός-Κινητικότητας-Erasmus.pdf
reg_005	Ποιες είναι οι απαιτήσεις για πρακτική άσκηση;	Διάρκεια, αποδεκτοί φορείς, διαδικασία αξιολόγησης.	Ο φοιτητής πρέπει να βρίσκεται στο 8ο ή μεγαλύτερο εξάμηνο και να έχει συγκεντρώσει τουλάχιστον 210 ECTS.	Εσωτερικός-Κανονισμός-Πρακτικής-Άσκησης-v5.pdf
reg_006	Τι ισχύει για κάρτα σίτισης;	Εισοδηματικά και λοιπά κριτήρια, ακαδ. έτος 2025-2026.	Απάντηση για την Ευρωπαϊκή Κάρτα Ασφάλισης Ασθένειας και την ιατρική περίθαλψη στο εξωτερικό.	ΠΡΟΫΠΟΘΕΣΕΙΣ_ΔΩΡΕΑΝ_ΣΙΤΙΣΗΣ_2025-2026.pdf

reg_007	Πόσα ECTS χρειάζομαι για να πάρω πτυχίο;	300 ECTS.	Απαιτούνται συνολικά 300 πιστωτικές μονάδες (ECTS), συμπεριλαμβανομένης της διπλωματικής εργασίας.	Συνήθεις_Ερωτήσεις_και_Απαντήσεις.txt
ann_001	Ακυρώθηκε κάποιο μάθημα αυτή την εβδομάδα;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
ann_002	Υπάρχει ανακοίνωση για εξετάσεις;	Ανακοινώσεις «Πρόγραμμα Εξεταστικής».	Ναι· το πρόγραμμα των εξετάσεων αναρτάται στη σελίδα εξετάσεων του τμήματος.	64190.txt, 63600.txt
ann_003	Πότε είναι η τελευταία ημερομηνία κατάθεσης αίτησης για πρακτική άσκηση;	Αναμενόμενη άρνηση (βλ. σημείωση).	Η τελευταία ημερομηνία κατάθεσης αίτησης είναι 29 Φεβρουαρίου 2024.	—
ann_004	Υπάρχει ανακοίνωση για αλλαγή αίθουσας;	Αναμενόμενη άρνηση.	Δεν υπάρχουν πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
ann_005	Τι ανακοινώσεις υπάρχουν για το εαρινό εξάμηνο;	«Έναρξη Εαρινού εξαμήνου 2025-26».	Έναρξη εαρινού εξαμήνου 2025-26: τα μαθήματα ξεκινούν στις 16 Φεβρουαρίου 2026.	65305.txt
ann_006	Υπάρχει ανακοίνωση για αναβολή παραδόσεων;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
ann_007	Πότε ανακοινώνονται οι βαθμολογίες εξετάσεων;	Καθορίζεται από τον Κανονισμό Εξετάσεων.	Ανακοινώνονται με ευθύνη του διδάσκοντα στο ηλεκτρονικό βαθμολόγιο του πληροφοριακού συστήματος.	Κανονισμός-εξετάσεων_TΜΠΗΣ.pdf
ann_008	Υπάρχει ανακοίνωση για υποτροφίες;	«Νέα Πρόσκληση για υποτροφίες - IKY».	Ναι· πρόσκληση Fulbright και προκήρυξη υποτρόφων για σπουδές δεύτερου κύκλου.	64158.txt
ann_009	Τι ανακοινώσεις υπάρχουν για εκδηλώσεις του τμήματος;	Αναμενόμενη άρνηση.	Εκδήλωση ενδιαφέροντος για την ομάδα κυβερνοασφάλειας και τη δράση Hellenic University Hack.	—
ann_010	Υπάρχουν ανακοινώσεις για τη διπλωματική εργασία;	«Διαθέσιμα θέματα διπλωματικών εργασιών».	Ναι· ανακοινώθηκαν θέματα διπλωματικών με διαδικτυακές συναντήσεις τον Μάρτιο 2026.	65412.txt
multi_001	Πότε έχω μάθημα Τεχνητή Νοημοσύνη και ποιο είναι το περιεχόμενό του;	Δευτέρα 11-13 και Τετάρτη 11-13· αντικείμενο τεχνικές ΤΝ.	Διδάσκεται στο 6ο εξάμηνο ως υποχρεωτικό· ακολουθεί περιγραφή περιεχομένου και ωρών.	schedule_output.json
multi_002	Ποια μαθήματα έχω Δευτέρα πρωί και τι γνωρίζεις για το αντικείμενό τους;	Μαθηματικά II, Μετρήσεις, Σχεδίαση Ψηφιακών Συστημάτων κ.ά.	Σχεδίαση Ψηφιακών Συστημάτων 9-11 (ΑμφΗ) και λοιπά μαθήματα, με περιγραφή αντικειμένου.	schedule_output.json

multi_003	Ποια είναι τα προαπαιτούμενα για τη Διπλωματική Εργασία και πότε μπορώ να αρχίσω;	Ορίζονται στον κανονισμό σπουδών.	Τουλάχιστον 210 πιστωτικές μονάδες και επιτυχής εξέταση στα προαπαιτούμενα μαθήματα.	Κανονισμός Διπλωματικών R1, Κανονισμός σπουδών v3
multi_004	Πώς συνδυάζεται το Erasmus με το πρόγραμμα σπουδών μου;	Αναγνώριση μαθημάτων βάσει κανονισμού κινητικότητας.	Επιτρέπει μετακίνηση για σπουδές ή πρακτική σε άλλο ίδρυμα, με αναγνώριση μαθημάτων.	Erasmus.txt, Κανονισμός-Κινητικότητας-Erasmus.pdf
multi_005	Τι ώρα έχω εργαστήριο Σχεδίαση Ψηφιακών Συστημάτων Τετάρτη και υπάρχει σχετική ανακοίνωση;	Τετάρτη 16-18 στο Δ1.	Εργαστήριο Τετάρτη 14-16, υποομάδα τ1/τ2, αίθουσα Δ1· δεν εντοπίστηκε σχετική ανακοίνωση.	schedule_output.json
hist_001	Ποιο ήταν το πρόγραμμα μαθημάτων το ακαδημαϊκό έτος 2023-2024;	Αναμενόμενη άρνηση.	Τα μαθήματα του χειμερινού ξεκίνησαν Δευτέρα 30 Σεπτεμβρίου 2024, σύμφωνα με το ακαδημαϊκό ημερολόγιο.	—
hist_002	Τι ανακοινώσεις είχαν βγει το 2022-2023;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
hist_003	Ποιο ήταν το χρονοδιάγραμμα εξετάσεων για το 2024-2025;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
oos_001	Ποια είναι η καλύτερη συνταγή για μουςακά;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
oos_002	Ποιο είναι το ΑΕΠ της Ελλάδας το 2025;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
oos_003	Πώς λειτουργεί ένας πυρηνικός αντιδραστήρας;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
oos_004	Ποια είναι η πρωτεύουσα της Αυστραλίας;	Αναμενόμενη άρνηση.	Η πρωτεύουσα της Αυστραλίας είναι η Κάνπερα, με περαιτέρω σχολιασμό.	—
oos_005	Γράψε μου κώδικα Python για να ταξινομήσω μια λίστα.	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
oos_006	Τι καιρό κάνει σήμερα στη Θεσσαλονίκη;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
oos_007	Ποιος κέρδισε το Champions League φέτος;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
iky_001	Πώς μπορώ να πάρω υποτροφία IKY;	Αναμενόμενη άρνηση.	Παρατίθενται βήματα και απαιτήσεις για τα προγράμματα υποτροφιών του IKY.	—

iky_002	Ποια είναι τα κριτήρια για υποτροφία IKY 2025-26;	Αναμενόμενη άρνηση.	Δεν έχω επαρκείς πληροφορίες στα έγγραφά μου για να απαντήσω με σιγουριά.	—
scholar_001	Ποιες είναι οι προϋποθέσεις για Erasmus;	Ορίζονται στον Κανονισμό Κινητικότητας Erasmus.	Εγγραφή σε επίσημο πρόγραμμα σπουδών του ΔΙΠΑΕ και λοιπές προϋποθέσεις του προγράμματος.	Κανονισμός-Κινητικότητας-Erasmus.pdf
law_001	Τι προβλέπει ο νόμος 4957 για τη διαγραφή φοιτητή;	Κανόνες διαγραφής από τον Εσωτερικό Κανονισμό· ανώτατη διάρκεια φοίτησης.	Η οριστική διαγραφή προβλέπεται ως πειθαρχική ποινή για συγκεκριμένα σοβαρά παραπτώματα.	ν. 4957/2022
espa_001	Πώς χρηματοδοτείται η πρακτική άσκηση από ΕΣΠΑ;	Χρηματοδότηση από επιχειρησιακό πρόγραμμα, με αποζημίωση και ασφάλιση.	Περιγραφή της χρηματοδότησης πριν και μετά τον Οκτώβριο 2022, με αναφορά σε πρώην ΤΕΙ.	—
staff_001	Ποιος είναι ο Πρόεδρος του τμήματος;	Ορίζεται βάσει του Εσωτερικού Κανονισμού Λειτουργίας.	Ο Πρόεδρος του τμήματος είναι ο Αντωνίου Ευστάθιος.	Εσωτερικός-κανονισμός-λειτουργίας-Τμήματος.pdf
exam_001	Πότε είναι η εξέταση στην Τεχνητή Νοημοσύνη;	Δευτέρα 19-01-2026, 16:30–18:30, αίθουσα Β1.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26_Χειμερινό.txt
exam_002	Σε ποια αίθουσα εξετάζομαι στα Τηλεπικοινωνιακά Συστήματα;	Αίθουσα Β1, Δευτέρα 19-01-2026, 11:30–13:30.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26_Χειμερινό.txt
exam_003	Ημερομηνία εξέτασης Επεξεργασία Σήματος.	Δευτέρα 19-01-2026, 09:00–12:30, αίθουσες 201, 208, 210, 211.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26_Χειμερινό.txt
exam_004	Πότε εξετάζομαι στα Κυκλώματα Συνεχούς Ρεύματος;	Δευτέρα 08-06-2026, 11:30–13:30, αίθουσα Β1.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26_Εαρινό.txt
exam_005	Ποιο είναι το πρόγραμμα εξετάσεων για την Οπτοηλεκτρονική;	Δευτέρα 08-06-2026, 09:00–11:00, αίθουσα 210.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26_Εαρινό.txt
exam_006	Πώς μπορώ να δω το πρόγραμμα εξετάσεων;	Ανά μάθημα ημερομηνία, ώρα και αίθουσα εξέτασης.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26 (και τα δύο)
exam_007	Πότε είναι η εξεταστική;	Χειμερινό 19-01-2026, εαρινό 08-06-2026.	— (χωρίς καταγεγραμμένη εκτέλεση)	exams_2025-26 (και τα δύο)