ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ & ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Δομημένη αποθήκευση και ανάκτηση πληροφορίας με χρήση Τεχνητής Νοημοσύνης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χατζηλία Ηλιάνα

Επιβλέπων : Γρηγόρης Μέντζας
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβρης 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Δομημένη αποθήκευση και ανάκτηση πληροφορίας με χρήση Τεχνητής Νοημοσύνης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χατζηλία Ηλιάνα

Επιβλέπων : Γρηγόρης Μέντζας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 17η Οκτωβρίου 2025.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γρηγόρης Μέντζας	Δημήτριος Ασκούνης	Ευάγγελος Μαρινάκης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβρης 2025

(Υπογραφή)

.....

Χατζηλία Ηλιάνα

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) έχουν φέρει επανάσταση στην επεξεργασία φυσικής γλώσσας, αλλά παρουσιάζουν κρίσιμους περιορισμούς όπως παραισθήσεις, ξεπερασμένη γνώση και έλλειψη εξειδικευμένης πληροφορίας. Η τεχνική Retrieval-Augmented Generation (RAG) αντιμετωπίζει αυτές τις προκλήσεις ενισχύοντας τα LLMs με εξωτερικές πηγές γνώσης. Η παρούσα διπλωματική εργασία υλοποιεί και αξιολογεί συγκριτικά τρεις προσεγγίσεις RAG: Naive RAG (γραμμή βάσης), RAPTOR RAG (ιεραρχική δενδρική οργάνωση), και KG-RAG (ενίσχυση με Γράφους Γνώσης). Χρησιμοποιώντας το dataset HotpotQA για ερωτήσεις πολλαπλών βημάτων, αξιολογούμε κάθε μέθοδο με ολοκληρωμένες μετρικές συμπεριλαμβανομένων F1, precision, recall και πλαισίων αξιολόγησης βασισμένα σε LLM. Τα αποτελέσματα δείχνουν ότι το KG-RAG επιτυγχάνει ανώτερη απόδοση (F1=0.76) αξιοποιώντας δομημένη γνώση, με κόστος όμως αυξημένης υπολογιστικής πολυπλοκότητας. Η ιεραρχική προσέγγιση του RAPTOR αποδείχθηκε λιγότερο αποδοτική για σύντομες ερωτήσεις, ενώ το Naive RAG προσέφερε τους ταχύτερους χρόνους απόκρισης με μέτρια ακρίβεια. Η εργασία παρέχει πρακτικές γνώσεις για τα trade-offs μεταξύ ακρίβειας, ερμηνευσιμότητας και υπολογιστικής αποδοτικότητας στα συστήματα RAG.

Λέξεις Κλειδιά: <<Μεγάλα Γλωσσικά Μοντέλα, Τεχνητή Νοημοσύνη, RAG, Γράφοι Γνώσης, Ανάκτηση Πληροφοριών, Naive RAG, RAPTOR RAG, KG-RAG>>

Abstract

Large Language Models (LLMs) have revolutionized natural language processing but suffer from critical limitations including hallucinations, outdated knowledge, and lack of domain-specific information. Retrieval-Augmented Generation (RAG) addresses these challenges by augmenting LLMs with external knowledge sources. This thesis implements and comparatively evaluates three RAG approaches: Naive RAG (baseline), RAPTOR RAG (hierarchical tree-based organization), and KG-RAG (Knowledge Graph-enhanced). Using the HotpotQA dataset for multi-hop question answering, we assess each method through comprehensive metrics including F1 score, precision, recall, and LLM-based evaluation frameworks. Results demonstrate that KG-RAG achieves superior performance (F1=0.76) by leveraging structured knowledge, though at the cost of increased computational complexity. RAPTOR's hierarchical approach proved less effective for short, factoid-style questions, while Naive RAG offered the fastest response times with moderate accuracy. This work provides practical insights into the trade-offs between accuracy, explainability, and computational efficiency in RAG systems.

Keywords: <<Large Language Models, Artificial Intelligence, Knowledge Graphs, Retrieval-Augmented Generation, Information extraction, Naive RAG, RAPTOR RAG, KG-RAG >>

Ευχαριστίες

Θα ήθελα να εκφράσω τις πιο θερμές μου ευχαριστίες προς τον επιβλέποντα καθηγητή μου, κ. **Γρηγόρη Μέντζα**, για την άψογη συνεργασία, την εμπιστοσύνη που μου έδειξε και την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον και απαιτητικό θέμα. Ιδιαίτερες ευχαριστίες οφείλω επίσης στον διδακτορικό ερευνητή **Ματθαίο Φικάρδο**, για τη συνεχή καθοδήγηση, τη διαθεσιμότητα και την πολύτιμη υποστήριξή του καθ' όλη τη διάρκεια της εκπόνησης της Διπλωματικής μου Εργασίας.

Θα ήθελα να ευχαριστήσω επίσης την οικογένεια μου και τους φίλους μου για όλη την υπομονή που έδειξαν κατά τη διάρκεια εκπόνησης αυτής της εργασίας.

Πίνακας περιεχομένων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1 Εισαγωγή

1.1 Μεγάλα Γλωσσικά Μοντέλα και η Ανάγκη για Ενίσχυση με Εξωτερική Γνώση

Τα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models - LLMs), όπως το GPT-4 και το PaLM2, έχουν φέρει επανάσταση στον χώρο της Τεχνητής Νοημοσύνης, επιδεικνύοντας πρωτοφανείς δυνατότητες στην κατανόηση και παραγωγή φυσικής γλώσσας. Η ικανότητά τους να εκτελούν ποικίλες εργασίες, από απλή ερώτηση-απάντηση μέχρι σύνθετη λογική συλλογιστική, τα έχει καταστήσει θεμελιώδη εργαλεία για πλήθος εφαρμογών σε τομείς όπως η εκπαίδευση, η επιστημονική έρευνα και η επιχειρηματική ευφυΐα [1]. Ωστόσο, αντιμετωπίζουν σημαντικούς περιορισμούς: η γνώση τους περιορίζεται χρονικά στις πηγές εκπαίδευσής τους, παράγουν συχνά πλασματικές απαντήσεις («παραισθήσεις»), δεν παρέχουν διαφανή εξήγηση για τις αποφάσεις τους, και υστερούν σε εξειδικευμένα πεδία λόγω έλλειψης domain-specific γνώσης [1], [2].

Για την αντιμετώπιση αυτών των προκλήσεων, έχει αναπτυχθεί η τεχνική Retrieval-Augmented Generation (RAG), η οποία συνδυάζει τη γενετική ισχύ των LLMs με τη δυνατότητα δυναμικής ανάκτησης πληροφορίας από εξωτερικές βάσεις γνώσης. Το RAG επιτρέπει στα μοντέλα να αξιοποιούν επικαιροποιημένα, τεκμηριωμένα δεδομένα κατά τη στιγμή της παραγωγής απαντήσεων, μειώνοντας δραστικά τις παραισθήσεις και βελτιώνοντας την πραγματολογική ακρίβεια. Η αποτελεσματικότητα ενός συστήματος RAG εξαρτάται όμως κρίσιμα από την ποιότητα της ανάκτησης και τον τρόπο οργάνωσης της γνώσης, ενώ υπάρχουν σημαντικά ερωτήματα σχετικά με την επεκτασιμότητα, την υπολογιστική αποδοτικότητα και την προσαρμοστικότητα διαφορετικών προσεγγίσεων RAG σε ποικίλα είδη εργασιών.

1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία έχει ως κεντρικό στόχο την υλοποίηση, αξιολόγηση και συγκριτική μελέτη τριών διαφορετικών προσεγγίσεων Retrieval-Augmented Generation: (1) **Naive RAG**, η κλασική προσέγγιση που χρησιμοποιεί απλή σημασιολογική ανάκτηση από διανυσματικό χώρο, (2) **RAPTOR RAG**, που εισάγει ιεραρχική δενδρική δομή μέσω clustering και αφαιρετικής περίληψης για οργάνωση της γνώσης σε πολλαπλά επίπεδα, και (3) **KG-RAG**, που ενσωματώνει δομημένη γνώση υπό μορφή Γράφων Γνώσης με εξαγωγή τριπλετών και αξιοποίηση σημασιολογικών σχέσεων. Κάθε προσέγγιση αντιμετωπίζει συγκεκριμένα προβλήματα: η Naive RAG παρέχει τεκμηριωμένο πλαίσιο αλλά αδυνατεί σε σύνθετα ερωτήματα, η RAPTOR επιλύει προβλήματα πληρότητας και οργάνωσης μέσω ιεραρχίας, ενώ η KG-RAG αντιμετωπίζει θέματα σχετικότητας, θορύβου και πολυβηματικού συλλογισμού.

Για την αξιολόγηση χρησιμοποιείται το HotpotQA dataset με πολυδιάστατες μετρικές (F1, Precision, Recall, DeepEval), στοχεύοντας να απαντηθούν ερωτήματα σχετικά με την επίδραση της οργάνωσης γνώσης στην ποιότητα απαντήσεων, τη συμβολή ιεραρχικών και γραφο-δομημένων προσεγγίσεων, και τα trade-offs μεταξύ ακρίβειας και υπολογιστικού κόστους. Τα αποτελέσματα δείχνουν ότι η KG-RAG επιτυγχάνει την υψηλότερη απόδοση (F1=0.76) αλλά με σημαντικά αυξημένο υπολογιστικό κόστος, η RAPTOR δεν αποδίδει ικανοποιητικά σε σύντομα αποσπάσματα αλλά υπόσχεται για εκτενή κείμενα, ενώ η Naive RAG προσφέρει την καλύτερη ισορροπία ταχύτητας-απόδοσης.

1.3 Οργάνωση κειμένου

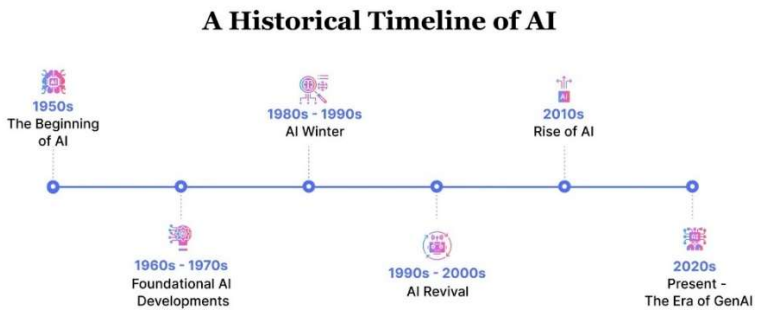
Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο και οι σχετικές εργασίες, συμπεριλαμβανομένης της εξέλιξης της Τεχνητής Νοημοσύνης, των Μεγάλων Γλωσσικών Μοντέλων και των τεχνικών RAG και Knowledge Graphs. Στο Κεφάλαιο 3 αναλύεται η προτεινόμενη προσέγγιση με τα ερευνητικά ερωτήματα και την περιγραφή των τριών μεθόδων RAG. Στο Κεφάλαιο 4 παρουσιάζεται η πρακτική υλοποίηση με τεχνικές λεπτομέρειες προ-επεξεργασίας, ανάκτησης και παραγωγής απαντήσεων. Στο Κεφάλαιο 5 αναλύεται η μεθοδολογία αξιολόγησης, το dataset HotpotQA, οι μετρικές και τα πειραματικά αποτελέσματα. Τέλος, στο Κεφάλαιο 6 συνοψίζονται τα συμπεράσματα, συζητούνται οι περιορισμοί και προτείνονται κατευθύνσεις μελλοντικής έρευνας.

2 Σχετικές εργασίες

2.1 Τεχνητή Νοημοσύνη και Εξέλιξή της

Η πορεία της Τεχνητής Νοημοσύνης (ΤΝ), όπως φαίνεται και στην *Εικόνα 1*, μπορεί να ιδωθεί ως μια συνεχής μετάβαση από συστήματα βασισμένα σε κανόνες προς μοντέλα που μαθαίνουν αυτόματα από δεδομένα. Η πρώτη εποχή, γνωστή ως Συμβολική ΤΝ (Symbolic AI) ή *Good Old-Fashioned AI (GOFAI)*, κυριάρχησε από τη δεκαετία του 1960 έως το 1980. Στηρίχθηκε στην υπόθεση ότι η νοημοσύνη μπορεί να περιγραφεί με λογικούς κανόνες και συμβολικές αναπαραστάσεις. Τα έμπειρα συστήματα της περιόδου, όπως το DENDRAL[3] και το MYCIN[4], έδειξαν την αξία αυτής της προσέγγισης αλλά και τους περιορισμούς της: έλλειψη προσαρμοστικότητας και δυσκολία στην ενσωμάτωση νέας γνώσης [5] .

Στη συνέχεια, η έρευνα στράφηκε σε στατιστικές και μαθησιακές μεθόδους, σηματοδοτώντας τη μετάβαση στη Μηχανική Μάθηση (Machine Learning). Η εισαγωγή πιθανοτικών μοντέλων, όπως τα Bayesian networks [6], και η ανάπτυξη αλγορίθμων όπως οι Support Vector Machines (SVMs) και τα decision tress, έδειξαν ότι η γνώση μπορεί να προκύψει από τα δεδομένα και όχι μόνο από κανόνες [7], [8].



Εικόνα 1 - Η εξέλιξη της τεχνητής νοημοσύνης ανά τα χρόνια

Η Βαθιά Μάθηση (Deep Learning), με αρχιτεκτονικές όπως τα Convolutional Neural Networks (CNNs) και τα Recurrent Neural Networks (RNNs), έφερε δραματικές βελτιώσεις σε προβλήματα όρασης και φυσικής γλώσσας [9], [10]. Η καθιέρωση της αρχιτεκτονικής

Transformer (Vaswani et al., 2017) αποτέλεσε σημείο καμπής, προσφέροντας πιο αποδοτική εκπαίδευση και ικανότητα κατανόησης μακροχρόνιων εξαρτήσεων [11], [12].

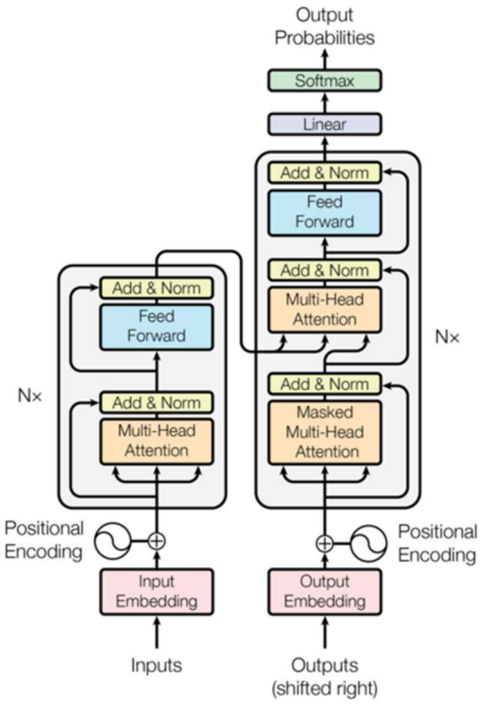
Η μετάβαση αυτή – από τη συμβολική ΤΝ, στατιστικά μοντέλα και βαθιά μάθηση, έως τους Transformers – έθεσε τα θεμέλια για την ανάπτυξη των σημερινών Μεγάλων Γλωσσικών Μοντέλων (LLMs), τα οποία αποτελούν το επόμενο βήμα στην ιστορική εξέλιξη της ΤΝ [13], [14].

2.2 Μεγάλα Γλωσσικά Μοντέλα (LLMs)

2.2.1 Αρχιτεκτονική Transformer

Η ραγδαία ανάπτυξη των LLMs κατέστη δυνατή χάρη στην αρχιτεκτονική Transformer, η οποία φαίνεται στην *Εικόνα 2*, που εισήχθη από τους Vaswani et al. (2017) [11]. Ο Transformer σηματοδότησε ένα σημείο καμπής, καθώς εγκατέλειψε τις αναδρομικές δομές των RNNs και LSTMs και βασίστηκε αποκλειστικά στον μηχανισμό της προσοχής (attention).

Ο μηχανισμός αυτό-προσοχής (self-attention) επιτρέπει σε κάθε λέξη μιας πρότασης να λαμβάνει υπόψη της όλες τις υπόλοιπες, ανεξάρτητα από τη θέση τους, προσφέροντας έτσι τη δυνατότητα κατανόησης μακροχρόνιων εξαρτήσεων. Επιπλέον, ο πολυκεφαλικός μηχανισμός προσοχής (multi-head attention) δίνει τη δυνατότητα στο μοντέλο να εστιάζει ταυτόχρονα σε διαφορετικές πτυχές των δεδομένων, ενώ η κωδικοποίηση θέσης (positional encoding) εξασφαλίζει ότι η σειρά των λέξεων λαμβάνεται υπόψη, παρά την απουσία αναδρομής [12].



Εικόνα 2 - Η αρχιτεκτονική του Transformer [6]

Η αρχιτεκτονική του Transformer είναι ιδιαίτερα αποδοτική, καθώς επιτρέπει παράλληλη επεξεργασία όλων των στοιχείων της εισόδου, μειώνοντας τον χρόνο εκπαίδευσης σε σχέση με τα RNNs. Αυτή η αποδοτικότητα, σε συνδυασμό με την ικανότητα κλιμάκωσης, άνοιξε τον δρόμο για την εκπαίδευση των σημερινών LLMs σε τεράστια δεδομένα [9], [14].

Συνολικά, ο Transformer δεν αποτέλεσε απλώς μια νέα αρχιτεκτονική, αλλά τη βάση για την ανάπτυξη των πιο ισχυρών LLMs της τελευταίας δεκαετίας, από το GPT-3 έως τα σύγχρονα ανοικτά και εμπορικά μοντέλα [13].

2.2.2 Ορισμός και Βασικές Αρχές

Τα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models – LLMs) αποτελούν την πιο πρόσφατη και σημαντική εξέλιξη στον χώρο της Επεξεργασίας Φυσικής Γλώσσας. Στην ουσία πρόκειται για νευρωνικά δίκτυα πολύ μεγάλης κλίμακας, με δισεκατομμύρια παραμέτρους, τα οποία εκπαιδεύονται σε τεράστιους όγκους δεδομένων με στόχο την πρόβλεψη της επόμενης λέξης σε μια ακολουθία [15]. Μέσω αυτής της διαδικασίας μαθαίνουν τα στατιστικά μοτίβα της γλώσσας και αποκτούν ικανότητα παραγωγής συνεκτικού και νοηματικά πλούσιου κειμένου.

Ένα από τα σημαντικότερα χαρακτηριστικά τους είναι ότι, όσο αυξάνεται η κλίμακα τους, εμφανίζονται αναδυόμενες ικανότητες που δεν υπήρχαν σε μικρότερα μοντέλα. Παραδείγματα αποτελούν η εκτέλεση λογικών συλλογισμών, η κατανόηση σύνθετων οδηγιών και η προσαρμογή σε νέες εργασίες χωρίς ρητή εκπαίδευση για αυτές [10], [15].

Ένα σημαντικό χαρακτηριστικό των LLM είναι το context window τους. Το context window αναφέρεται στο μέγιστο μήκος κειμένου που ένα LLM μπορεί να επεξεργαστεί ταυτόχρονα, μετρούμενο σε tokens (υπομονάδες λέξεων ή χαρακτήρων). Ουσιαστικά καθορίζει πόσο «μνήμη» έχει το μοντέλο κατά τη διάρκεια μιας αλληλεπίδρασης. Περιλαμβάνει τόσο την είσοδο του χρήστη (prompt) όσο και την απάντηση που παράγει το μοντέλο [16].

Τα πρώτα LLMs είχαν περιορισμένο context window (π.χ. 2.048 ή 4.096 tokens), γεγονός που περιόριζε την ικανότητά τους να διαχειρίζονται μακροσκελή έγγραφα ή πολύπλοκες συνομιλίες. Τα σύγχρονα μοντέλα έχουν επεκτείνει σημαντικά αυτό το όριο, με ορισμένα να φτάνουν τα εκατοντάδες χιλιάδες ή και εκατομμύρια tokens, επιτρέποντας την ανάλυση ολόκληρων βιβλίων, επιστημονικών άρθρων ή εκτεταμένων βάσεων κωδικού σε μία αλληλεπίδραση [17].

Το μέγεθος του context window επηρεάζει άμεσα την απόδοση του μοντέλου σε εργασίες που απαιτούν μακροπρόθεσμη συνοχή, όπως η σύνοψη μεγάλων κειμένων, η απάντηση

ερωτημάτων βασισμένη σε εκτεταμένο υλικό (retrieval-augmented generation), ή η διατήρηση συνεκτικών συνομιλιών σε πολλές ανταλλαγές μηνυμάτων.

Η εκπαίδευση των LLMs περιλαμβάνει δύο βασικά στάδια:

- **Προ-εκπαίδευση (pre-training):** το μοντέλο εκπαιδεύεται σε τεράστια σώματα κειμένου χωρίς επισημάνσεις, με στόχο την εκμάθηση γενικών γλωσσικών αναπαραστάσεων.
- **Προσαρμογή (fine-tuning):** στη συνέχεια το μοντέλο μπορεί να εξειδικευθεί σε συγκεκριμένα καθήκοντα, χρησιμοποιώντας μικρότερα αλλά επισημασμένα σύνολα δεδομένων, ώστε να βελτιώσει την απόδοσή του σε ειδικά πεδία [13], [14].

Πέρα από την εκπαίδευση, τα LLMs μπορούν να καθοδηγηθούν απευθείας μέσω προτροπών (prompts). Η τεχνική αυτή, γνωστή ως prompting, επιτρέπει την εκτέλεση εργασιών μόνο μέσω μιας οδηγίας σε φυσική γλώσσα. Σε πολλές περιπτώσεις, τα LLMs αξιοποιούν και το φαινόμενο της μάθησης από τα συμφραζόμενα (in-context learning), δηλαδή την ικανότητα να μαθαίνουν νέες εργασίες μόνο από λίγα παραδείγματα που περιλαμβάνονται στο ίδιο το prompt, χωρίς να απαιτείται νέα εκπαίδευση [8], [15]

Με τα χαρακτηριστικά αυτά, την κλίμακα, τις αναδυόμενες ικανότητες, την προ-εκπαίδευση και την ευελιξία μέσω prompting, τα LLMs αποτελούν σήμερα «μοντέλα θεμελίων» πάνω στα οποία χτίζονται πλήθος εφαρμογών ΤΝ [9].

2.2.3 Βασικές Ικανότητες και Εφαρμογές των LLMs

Τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) έχουν καθιερωθεί ως «μοντέλα θεμελίων» της σύγχρονης Τεχνητής Νοημοσύνης, καθώς συνδυάζουν μια σειρά από ικανότητες που τα καθιστούν εξαιρετικά ευέλικτα και ισχυρά. Η βασική τους δυνατότητα είναι η **κατανόηση και παραγωγή φυσικής γλώσσας** με τρόπο που συχνά προσεγγίζει το ανθρώπινο επίπεδο. Μέσα από την εκπαίδευσή τους σε τεράστιους όγκους δεδομένων, τα LLMs αποκτούν την ικανότητα να παράγουν συνεκτικά και νοηματικά πλούσια κείμενα, κάτι που τα καθιστά ιδανικά για εφαρμογές όπως η αυτόματη συγγραφή, οι συνομιλιακοί πράκτορες και η δημιουργία περιεχομένου [9], [10].

Ένα ιδιαίτερα σημαντικό χαρακτηριστικό τους είναι το φαινόμενο της **μάθησης από τα συμφραζόμενα (in-context learning)**. Χάρη στην κλίμακα τους, μοντέλα όπως το GPT-3 έχουν επιδείξει ότι μπορούν να προσαρμόζονται σε νέες εργασίες απλώς λαμβάνοντας ως είσοδο μερικά παραδείγματα στο πλαίσιο μιας προτροπής, χωρίς να απαιτείται περαιτέρω εκπαίδευση. Αυτή η ικανότητα, που συνδέεται με τις λεγόμενες «αναδυόμενες ικανότητες»

(emergent abilities), διαφοροποιεί τα LLMs από τα παραδοσιακά μοντέλα μηχανικής μάθησης και ανοίγει τον δρόμο για μια πιο γενικευμένη και δυναμική χρήση τους [10], [15].

Επιπλέον, τα LLMs έχουν δείξει αυξανόμενη ικανότητα **λογικού συλλογισμού** και **επίλυσης προβλημάτων**, συμπεριλαμβανομένων μαθηματικών ή συλλογιστικών εργασιών πολλών βημάτων. Παρά το γεγονός ότι η αξιοπιστία τους σε τέτοιες διαδικασίες δεν είναι πάντοτε σταθερή και αποτελεί ενεργό πεδίο έρευνας, η προοπτική χρήσης τους σε γνωσιακά απαιτητικά περιβάλλοντα είναι ιδιαίτερα ελκυστική [10]. Η προσαρμοστικότητα ενισχύεται περαιτέρω μέσω τεχνικών όπως το fine-tuning ή το instruction tuning, που επιτρέπουν τη στόχευση σε συγκεκριμένους τομείς, από την ιατρική και τη νομική μέχρι την ανάλυση επιστημονικών δεδομένων [13], [18].

Οι εφαρμογές των LLMs καλύπτουν ένα ευρύ φάσμα. Χρησιμοποιούνται σε συστήματα ερωταποκρίσεων που αξιοποιούν μεγάλες βάσεις γνώσης για να δίνουν απαντήσεις σε φυσική γλώσσα [13], στη μηχανική μετάφραση και στην πολύγλωσσή επεξεργασία όπου επιτυγχάνουν επιδόσεις που ξεπερνούν τις παραδοσιακές μεθόδους [9], αλλά και στη δημιουργία περιεχομένου, από την παραγωγή περίληψης έως την αυτόματη παραγωγή κώδικα [8]. Σημαντική είναι και η αξιοποίηση τους στην επιστημονική και ιατρική υποβοήθηση, όπου χρησιμοποιούνται για αναζήτηση και σύνοψη επιστημονικής βιβλιογραφίας ή για τη στήριξη ιατρικών αποφάσεων [10]. Τέλος, σε συνδυασμό με τεχνικές όπως το Retrieval-Augmented Generation (RAG), τα LLMs χρησιμοποιούνται όλο και περισσότερο σε εφαρμογές αναζήτησης και πληροφόρησης [14], αλλά και στην εκπαίδευση, προσφέροντας εξατομικευμένη υποστήριξη μάθησης με βάση τις ανάγκες κάθε χρήστη [9].

Συνολικά, τα LLMs συνιστούν μια καθολική πλατφόρμα γλωσσικής νοημοσύνης, με δυνατότητες που υπερβαίνουν τις στενά ορισμένες εφαρμογές και με προοπτική να ενσωματωθούν σε ένα ευρύ φάσμα τομέων της ανθρώπινης δραστηριότητας. Ο συνδυασμός της κλίμακας, της μάθησης από συμφραζόμενα και της προσαρμοστικότητας τα καθιστά αναπόσπαστο κομμάτι του σύγχρονου οικοσυστήματος της ΤΝ.

2.2.4 Περιορισμοί και Προβλήματα των LLMs

Παρά τις εντυπωσιακές τους δυνατότητες, τα Μεγάλα Γλωσσικά Μοντέλα παρουσιάζουν σημαντικούς περιορισμούς, οι οποίοι επηρεάζουν την αξιοπιστία, την ηθική χρήση και τη

βιωσιμότητά τους. Τα προβλήματα αυτά σχετίζονται με τη φύση της εκπαίδευσής τους, την εξάρτηση από τεράστιους πόρους και τη δυσκολία ελέγχου της συμπεριφοράς τους.

Περιορισμός γνώσεων και χρονικοί περιορισμοί: Τα LLM διαθέτουν γνώσεις μόνο μέχρι την ημερομηνία λήξης των δεδομένων εκπαίδευσής τους, γεγονός που τα καθιστά ανίκανα να έχουν πρόσβαση ή να συλλογιστούν πληροφορίες, γεγονότα ή εξελίξεις που συνέβησαν μετά από αυτό το σημείο. Αυτός ο χρονικός περιορισμός είναι ιδιαίτερα προβληματικός σε τομείς που εξελίσσονται ραγδαία, όπως τα τρέχοντα γεγονότα, η επιστημονική έρευνα, η τεχνολογία και τα ρυθμιστικά πλαίσια. Η στατική φύση της γνώσης του μοντέλου δημιουργεί μια θεμελιώδη αποσύνδεση μεταξύ της εσωτερικής αναπαράστασης του μοντέλου και της δυναμικής κατάστασης του πραγματικού κόσμου.

Επιπλέον, ακόμη και για πληροφορίες που αφορούν την περίοδο εκπαίδευσης, τα LLM στερούνται ρητής χρονικής συνείδησης και ενδέχεται να συγχέουν πληροφορίες από διαφορετικές χρονικές περιόδους ή να μην αναγνωρίζουν πότε τα γεγονότα έχουν αλλάξει με την πάροδο του χρόνου. Αυτός ο περιορισμός γίνεται όλο και πιο σοβαρός καθώς διευρύνεται το χάσμα μεταξύ της ημερομηνίας λήξης της εκπαίδευσης και του χρόνου ανάπτυξης.[9]

Ψευδαισθήσεις και ανακρίβειες: Ένα διαδεδομένο πρόβλημα στα αποτελέσματα των LLM είναι το φαινόμενο των ψευδαισθήσεων, όπου τα μοντέλα παράγουν πληροφορίες που ακούγονται εύλογες, αλλά είναι ανακριβείς ή εντελώς πλασματικές. Αυτό συμβαίνει επειδή τα LLM είναι βασικά συστήματα αντιστοίχισης προτύπων και δημιουργίας κειμένου και όχι βάσεις δεδομένων γνώσης με επαληθευμένα γεγονότα. Βελτιστοποιούν τη γλωσσική ευλογοφάνεια και όχι την ακρίβεια των γεγονότων, οδηγώντας σε σίγουρες διαβεβαιώσεις ψευδών πληροφοριών, πλασματικών αναφορών και εσωτερικά ασυνεπών δηλώσεων.

Οι ψευδαισθήσεις προκύπτουν από διάφορες πηγές: μεροληψίες στα δεδομένα εκπαίδευσης, την τάση του μοντέλου να συμπληρώνει μοτίβα ακόμη και όταν δεν διαθέτει σχετική γνώση και την εγγενή αμφισημία στις αναπαραστάσεις της εκμάθησης. Είναι σημαντικό να σημειωθεί ότι τα LLM συνήθως δεν παρέχουν καμία ένδειξη αβεβαιότητας ή εμπιστοσύνης στα αποτελέσματά τους, καθιστώντας δύσκολο για τους χρήστες να διακρίνουν τις αξιόπιστες πληροφορίες από το περιεχόμενο που είναι αποτέλεσμα ψευδαισθήσεων.[19]

Έλλειψη βάσης και αναφοράς πηγών: Τα LLM δεν έχουν βάση σε εξωτερικές πηγές γνώσης, πράγμα που σημαίνει ότι δεν μπορούν να αναφέρουν ή να παραπέμπουν σε συγκεκριμένα έγγραφα, να επαληθεύουν ισχυρισμούς σε σχέση με έγκυρες πηγές ή να παρέχουν αποδεικτικά στοιχεία για τους ισχυρισμούς τους. Η γνώση που είναι κωδικοποιημένη στις παραμέτρους του

μοντέλου κατανέμεται σε δισεκατομμύρια βαρύτητες, καθιστώντας πρακτικά αδύνατο να εντοπιστούν συγκεκριμένα αποτελέσματα σε συγκεκριμένα παραδείγματα εκπαίδευσης ή να ενημερωθούν συγκεκριμένα γεγονότα χωρίς επανεκπαίδευση.

Αυτή η απουσία αναφοράς πηγών δημιουργεί σημαντικές προκλήσεις για τομείς που απαιτούν επαληθευσιμότητα, όπως η ακαδημαϊκή έρευνα, η νομική ανάλυση, η ιατρική διάγνωση και η δημοσιογραφική κάλυψη. Οι χρήστες δεν μπορούν να αξιολογήσουν την προέλευση ή την αξιοπιστία των πληροφοριών που παρέχονται από τα LLM, ούτε μπορούν να ακολουθήσουν τις αναφορές για να επαληθεύσουν ή να διερευνήσουν θέματα σε μεγαλύτερο βάθος.[20], [21]

Περιορισμοί του context window: Παρά τις προόδους στην επέκταση των context windows, τα LLM παραμένουν ουσιαστικά περιορισμένα από το πεπερασμένο μήκος των εισροών. Η υπολογιστική πολυπλοκότητα του μηχανισμού προσοχής αυξάνεται τετραγωνικά με το μήκος της ακολουθίας, επιβάλλοντας πρακτικούς περιορισμούς στον όγκο του context που μπορεί να υποβληθεί σε επεξεργασία. Αυτός ο περιορισμός είναι ιδιαίτερα προβληματικός όταν δουλεύουμε με μεγάλα έγγραφα, εκτεταμένες βάσεις δεδομένων, πολλαπλές πηγές ή εργασίες που απαιτούν την ενσωμάτωση πληροφοριών σε μεγάλα σώματα κειμένων.

Ακόμη και εντός του context window, τα LLM ενδέχεται να παρουσιάζουν μειωμένη απόδοση σε πληροφορίες που βρίσκονται στο μέσο μεγάλων περιβαλλόντων (το πρόβλημα «lost in the middle») και ενδέχεται να δυσκολεύονται να συνθέσουν αποτελεσματικά πληροφορίες που είναι κατανεμημένες σε πολλά διαφορετικά αποσπάσματα.[22]

Αδυναμία πρόσβασης σε πληροφορίες σε πραγματικό χρόνο ή ιδιωτικές πληροφορίες: Τα LLM δεν μπορούν να έχουν πρόσβαση σε εξωτερικές βάσεις δεδομένων, ροές πληροφοριών σε πραγματικό χρόνο, ιδιόκτητα έγγραφα ή δεδομένα συγκεκριμένων χρηστών, εκτός εάν παρέχονται ρητά στο πλαίσιο των εισροών τους. Αυτή η απομόνωση από εξωτερικές πηγές πληροφοριών περιορίζει σοβαρά τη χρησιμότητά τους για εφαρμογές που απαιτούν τρέχοντα δεδομένα (όπως τιμές μετοχών, καιρικές συνθήκες ή έκτακτα νέα), εξατομικευμένες πληροφορίες (όπως προτιμήσεις ή ιστορικό χρηστών) ή γνώσεις συγκεκριμένες για την επιχείρηση (όπως εσωτερικές πολιτικές ή ιδιόκτητα δεδομένα).[14]

Περιορισμοί στη συλλογιστική και συστηματικά σφάλματα: Αν και τα LLM επιδεικνύουν εντυπωσιακές ικανότητες συλλογιστικής σε ορισμένες εργασίες, παρουσιάζουν συστηματικές αστοχίες που αποκαλύπτουν θεμελιώδεις περιορισμούς στις διαδικασίες συλλογιστικής τους. Αυτές περιλαμβάνουν:

- Αριθμητικά και μαθηματικά σφάλματα: Τα LLM συχνά κάνουν λάθη στους υπολογισμούς, ιδιαίτερα με μεγάλους αριθμούς ή μαθηματικά προβλήματα πολλαπλών βημάτων [23].
- Λογικές ασυνέπειες: Τα μοντέλα μπορεί να παράγουν αντιφατικές δηλώσεις μέσα στην ίδια απάντηση ή να αποτυγχάνουν να διατηρήσουν τη λογική συνοχή σε εκτεταμένες αλυσίδες συλλογιστικής [24].
- Αποτυχίες στη σύνθετη συλλογιστική: Τα LLM δυσκολεύονται με προβλήματα που απαιτούν συστηματική σύνθεση πολλαπλών βημάτων συλλογιστικής ή ενσωμάτωση διαφορετικών πληροφοριών [25].
- Περιορισμοί στην αντιφατική συλλογιστική: Τα μοντέλα συχνά δυσκολεύονται με υποθετικά σενάρια ή συλλογιστική σχετικά με εναλλακτικές λύσεις στα δεδομένα εκπαίδευσής τους [26].

Αυτές οι αποτυχίες υποδηλώνουν ότι τα LLM βασίζονται κυρίως στην αντιστοίχιση προτύπων και στις στατιστικές συσχετίσεις και όχι σε μηχανισμούς τυπικής συλλογιστικής, περιορίζοντας την αξιοπιστία τους για εργασίες που απαιτούν αυστηρή λογική συλλογή.[10]

Προκαταλήψεις και περιορισμοί των δεδομένων εκπαίδευσης: Τα LLM κληρονομούν και ενδεχομένως ενισχύουν τις προκαταλήψεις που υπάρχουν στα δεδομένα εκπαίδευσής τους, συμπεριλαμβανομένων των κοινωνικών προκαταλήψεων, των πολιτισμικών υποθέσεων και των δημογραφικών ανισορροπιών. Επιπλέον, η σύνθεση των δεδομένων εκπαίδευσης επηρεάζει τις δυνατότητες του μοντέλου με τρόπους που ενδέχεται να μην συνάδουν με τις ανάγκες ανάπτυξης. Οι υπερ-εκπροσωπούμενοι τομείς στα δεδομένα εκπαίδευσης αποδίδουν καλύτερες επιδόσεις, ενώ οι υπο-εκπροσωπούμενοι τομείς υποφέρουν από κακή γενίκευση. Αυτό αποτελεί και πρόβλημα για τους εξειδικευμένους τομείς όπως η ιατρική, το δίκαιο, η μηχανική ή η επιστημονική έρευνα που συχνά στερούνται του βάθους και της ακρίβειας που απαιτείται. Η αδιαφάνεια της σύνθεσης των δεδομένων εκπαίδευσης σε πολλά εμπορικά LLM περιπλέκει περαιτέρω τις προσπάθειες κατανόησης και μετριασμού αυτών των προκαταλήψεων, καθώς η προέλευση και τα χαρακτηριστικά των σωμάτων εκπαίδευσης είναι συχνά αποκλειστικές πληροφορίες.[21]

Έλλειψη διαφάνειας και ερμηνευσιμότητα: Η κατανεμημένη φύση της αναπαράστασης της γνώσης στα LLM τα καθιστά ουσιαστικά δύσκολα να ερμηνευθούν. Το να κατανοήσουμε γιατί ένα μοντέλο παράγει ένα συγκεκριμένο αποτέλεσμα, σε ποιες πληροφορίες βασίζεται ή πώς να διορθώσουμε συγκεκριμένα σφάλματα παραμένει μια ανοιχτή πρόκληση. Αυτή η αδιαφάνεια είναι προβληματική για την εδραίωση της εμπιστοσύνης, τον εντοπισμό σφαλμάτων, τη

διασφάλιση της συμμόρφωσης με τους κανονισμούς και την κατανόηση της συμπεριφοράς του μοντέλου σε εφαρμογές κρίσιμες για την ασφάλεια.[20]

2.3 Retrieval-Augmented Generation (RAG)

Για να αντιμετωπιστούν κάποιοι από τους κρίσιμους περιορισμούς των σύγχρονων LLMs μία από τις μεθόδους που αναπτύχθηκε είναι το RAG. Το RAG είναι ένα υβριδικό αρχιτεκτονικό πλαίσιο που συνδυάζει ένα LLM με μηχανισμούς ανάκτησης πληροφορίας, επιτρέποντας στο μοντέλο να αξιοποιεί εξωτερικές πηγές γνώσης κατά τη δημιουργία απαντήσεων [27]. Το RAG έρχεται να μετριάσει τα προβλήματα των LLMs, επιτρέποντας στο μοντέλο να ανακτά δυναμικά τα πιο σχετικά κομμάτια πληροφορίας από εξωτερικές βάσεις γνώσεων (π.χ. έγγραφα, άρθρα, βάσεις δεδομένων) και να τα χρησιμοποιεί ως συμπληρωματικό πλαίσιο κατά την παραγωγή της απάντησης [28]. Με αυτόν τον τρόπο, το RAG συνδυάζει τη γνώση που είναι ενσωματωμένη στις παραμέτρους του LLM με την επικαιροποιημένη γνώση από εξωτερικές πηγές, ενισχύοντας την ακρίβεια και την αξιοπιστία των απαντήσεων του μοντέλου.[13] Ιδιαίτερα σε γνώσιο-εντατικά αιτήματα (knowledge-intensive tasks), η χρήση εξωτερικών δεδομένων μέσω RAG μειώνει την πιθανότητα παραγωγής πραγματολογικά λανθασμένου περιεχομένου και επιτρέπει την ενσωμάτωση πρόσφατης ή εξειδικευμένης πληροφορίας στο παραγόμενο κείμενο . [29]

2.3.1 Αρχιτεκτονικό Σχήμα και Ροή

Μια τυπική υλοποίηση RAG αποτελείται από δύο βασικά μέρη: τον μηχανισμό ανάκτησης (retriever) και τον γεννήτορα κειμένου (generator) που συνήθως είναι το ίδιο το LLM . Η ροή λειτουργίας ξεκινά με το ερώτημα του χρήστη: ο retriever λαμβάνει το ερώτημα και αναζητά στην εξωτερική βάση γνώσεων τα πλέον σχετικά αποσπάσματα κειμένου ή δεδομένα που ταιριάζουν στο θέμα του ερωτήματος. [28] Αυτή η αναζήτηση μπορεί να γίνεται με υπολογισμό σημασιολογικής ομοιότητας, π.χ. μετατροπή του ερωτήματος και των υποψήφιων εγγράφων σε διανυσματικές αναπαραστάσεις και υπολογισμό συνημίτονου μεταξύ τους, έτσι ώστε να εντοπιστούν τα κείμενα με τη μεγαλύτερη συνάφεια προς το ερώτημα. [13] Ο retriever επιστρέφει τα κορυφαία K σχετικά τεμάχια πληροφορίας (π.χ. προτάσεις ή παραγράφους) από την γνώση βάσης.

Στη συνέχεια, αυτά τα ανακτημένα αποσπάσματα συνδυάζονται με το αρχικό ερώτημα και δίνονται ως είσοδος (prompt) στον γεννήτορα, το LLM. Ο γεννήτορας επεξεργάζεται το

ερώτημα έχοντας πλέον πρόσβαση και στα επιπλέον δεδομένα που αντλήθηκαν και παράγει την τελική απάντηση. [13] Ουσιαστικά, το LLM καλείται να συνθέσει μια απάντηση χρησιμοποιώντας τόσο τη δική του «εγγενή» γνώση (αυτή που έχει μάθει από τα δεδομένα προπόνησης) όσο και τις «επικουρικές» πληροφορίες από τον retriever. Αυτό επιτρέπει στο σύστημα να παρέχει απαντήσεις που είναι πιο πλήρεις και τεκμηριωμένες, ακόμα και για ερωτήματα που υπερβαίνουν τις γνώσεις που είχαν αρχικά κωδικοποιηθεί στο μοντέλο. Είναι αξιοσημείωτο ότι πριν την φάση της ανάκτησης προϋποθέτεται μια φάση ευρετηρίασης (indexing) της εξωτερικής γνώσης: τα έγγραφα της βάσης γνώσης επεξεργάζονται (π.χ. διαχωρίζονται σε μικρά τμήματα, chunks, και μετατρέπονται σε διανύσματα μέσω μοντέλων embedding) και αποθηκεύονται σε μια δομή όπως μια διανυσματική βάση δεδομένων, ώστε οι μετέπειτα αναζητήσεις να είναι αποδοτικές . Ωστόσο, ο πυρήνας του RAG παραμένει η αλληλεπίδραση retriever και generator: ο retriever προσθέτει γνώση στο πλαίσιο και το LLM την αξιοποιεί για να παραγάγει την απάντηση.

2.3.2 Πλεονεκτήματα του RAG για τα LLMs

Η ενσωμάτωση του RAG σε συστήματα LLM προσφέρει πολλαπλά πλεονεκτήματα όσον αφορά την ποιότητα, επικαιρότητα και προσαρμοστικότητα των παραγόμενων αποτελεσμάτων:

Βελτιωμένη ακρίβεια και αξιοπιστία απαντήσεων: Επειδή το LLM τεκμηριώνει τις απαντήσεις του με πραγματικά δεδομένα από εξωτερικές πηγές, μειώνεται αισθητά η πιθανότητα παραγωγής αναληθών ισχυρισμών ή «hallucinations».[29] Το RAG παρέχει στο μοντέλο πρόσβαση σε επαληθευμένες πληροφορίες την ώρα της απάντησης, ενισχύοντας έτσι την πραγματολογική ορθότητα και την εγκυρότητα του περιεχομένου. Μελέτες έχουν δείξει ότι οι τεχνικές RAG μετριάζουν τις παραισθήσεις και βελτιώνουν τη συνολική ποιότητα της απόκρισης του μοντέλου [29], καθιστώντας τες ιδιαίτερα χρήσιμες σε εφαρμογές όπως η απάντηση σε ερωτήσεις γνώσης (knowledge-intensive QA).

Επικαιρότητα πληροφοριών: Σε αντίθεση με ένα στατικά προπονημένο LLM, ένα σύστημα με RAG μπορεί να παρέχει ενημερωμένες απαντήσεις που αντανακλούν τη νεότερη διαθέσιμη γνώση. Εφόσον η βάση γνώσεων ενημερώνεται συνεχώς, το RAG μπορεί να ανακτά πρόσφατες πληροφορίες (π.χ. γεγονότα που συνέβησαν μετά την εκπαίδευση του μοντέλου) και να τις ενσωματώνει στις απαντήσεις. Αυτό βελτιώνει σημαντικά την επικαιρότητα των αποκρίσεων

των LLMs, επιτρέποντάς τους να παραμένουν χρήσιμα παρά την παλαιότητα του training data. [28] Η δυνατότητα αξιοποίησης up-to-date πληροφορίας έχει καταστήσει το RAG κεντρική τεχνολογία για την ανάπτυξη προηγμένων chatbot και συστημάτων ερωταπαντήσεων στο πραγματικό περιβάλλον. [13]

Ευκολία προσαρμογής σε νέα γνώση και τομείς: Ένα από τα μεγαλύτερα πλεονεκτήματα του RAG είναι ότι δίνει τη δυνατότητα στα LLMs να προσαρμόζονται σε εξειδικευμένους τομείς ή νέες γνωσιακές περιοχές χωρίς να απαιτείται επανεκπαίδευση των μοντέλων. Αρκεί η παροχή μιας κατάλληλης συλλογής εγγράφων ή δεδομένων ως εξωτερική γνώση, και το LLM μέσω του retriever μπορεί να αντλήσει από αυτά τις σχετικές πληροφορίες κατά την απάντηση. Αυτό σημαίνει ότι μπορούμε να έχουμε ταχεία ανάπτυξη εφαρμογών βασισμένων σε LLM για συγκεκριμένους οργανισμούς ή θεματολογίες, χωρίς να τροποποιηθούν οι παράμετροι του μοντέλου, απλώς εμπλουτίζοντας τη γνώση που είναι προσβάσιμη μέσω του μηχανισμού ανάκτησης. [29]

2.3.3 Γενικές Κατηγορίες Προσεγγίσεων RAG

Η υλοποίηση ενός συστήματος RAG μπορεί να διαφοροποιηθεί σημαντικά ανάλογα με τις χρησιμοποιούμενες τεχνικές ανάκτησης και ενίσχυσης. Γενικά, οι προσεγγίσεις διακρίνονται στις παρακάτω κατηγορίες, καθεμία με τα δικά της χαρακτηριστικά:

Αραιή ανάκτηση (Sparse Retrieval): Σε αυτή την κατηγορία ανήκουν οι λεξικοκεντρικές μέθοδοι ανάκτησης, όπου η συνάφεια μεταξύ ερωτήματος και εγγράφου βασίζεται σε αντιστοίχιση όρων και στατιστικά μέτρα συχνότητας όρων. Κλασικό παράδειγμα αποτελεί η χρήση του αλγορίθμου BM25, ο οποίος μοντελοποιεί τη συνάφεια βάσει της συχνότητας και σπανιότητας των λέξεων-κλειδιών στο έγγραφο. Η αραιή ανάκτηση δεν χρησιμοποιεί πυκνές διανυσματικές αναπαραστάσεις, αλλά αξιοποιεί ευρετήρια αντίστροφων λιστών (inverted index) για ταχεία εύρεση εγγράφων που περιέχουν συγκεκριμένους όρους του ερωτήματος. [13] Αυτή η προσέγγιση είναι αποτελεσματική σε περιπτώσεις όπου το λεξιλόγιο του ερωτήματος ταιριάζει άμεσα με του εγγράφου, όμως μπορεί να αποτύχει να εντοπίσει αποτελέσματα όταν διατυπώνονται συνώνυμα ή παραφράσεις (καθώς δεν κατανοεί την εννοιολογική συνάφεια πέρα από τις ακριβείς λέξεις).[8]

Πυκνή ανάκτηση (Dense Retrieval): Εδώ η ανάκτηση πραγματοποιείται σε επίπεδο σημασιολογικής ομοιότητας αντί απλής αντιστοίχισης λέξεων. Τα ερωτήματα και τα έγγραφα

μετατρέπονται σε διανύσματα ενσωμάτωσης (embedding vectors) υψηλών διαστάσεων μέσω νευρωνικών μοντέλων encoder (π.χ. μοντέλα βασισμένα σε BERT), και η συνάφεια μετριέται με αποστάσεις ή συσχέτιση μεταξύ αυτών των διανυσμάτων (π.χ. συνημίτονο γωνίας). Η πυκνή ανάκτηση μπορεί να «καταλάβει» την ουσία του ερωτήματος και του κειμένου ακόμα κι αν δεν μοιράζονται τις ίδιες λέξεις, βρίσκοντας σχετικές πληροφορίες μέσω εννοιολογικής αντιστοίχισης. Πρόσφατες έρευνες έχουν εισαγάγει πληθώρα από ισχυρά μοντέλα ενσωμάτωσης ειδικά για retrieval (π.χ. Contriever, E5, BGE), τα οποία εκπαιδεύονται συχνά με συγκριτική μάθηση (contrastive learning) ώστε να τοποθετούν ερωτήματα και σχετικά έγγραφα κοντά στον διανυσματικό χώρο. [13] Η αποτελεσματικότητα της πυκνής ανάκτησης την έχει κάνει δημοφιλή επιλογή για RAG, ιδιαίτερα όταν οι ερωτήσεις διατυπώνονται σε φυσική γλώσσα και απαιτείται κατανόηση συμφραζομένων.

Υβριδικές μέθοδοι: Καμία από τις παραπάνω προσεγγίσεις δεν υπερτερεί απόλυτα σε όλες τις περιπτώσεις, γι' αυτό συχνά χρησιμοποιείται ένας συνδυασμός αραιής και πυκνής ανάκτησης για καλύτερα αποτελέσματα. Οι λεγόμενες υβριδικές τεχνικές εκτελούν ταυτόχρονα ή διαδοχικά και τους δύο τύπους αναζήτησης, αξιοποιώντας τα συμπληρωματικά πλεονεκτήματα καθενός . Για παράδειγμα, μια στρατηγική είναι να χρησιμοποιηθεί πρώτα ένα λεξικό σύστημα (π.χ. BM25) για τον εντοπισμό αρχικών υποψήφιων εγγράφων και στη συνέχεια η σειρά ή η επιλογή αυτών των εγγράφων να βελτιστοποιηθεί με ένα μοντέλο πυκνής ανάκτησης που θα φιλτράρει όσα είναι πραγματικά συναφή σε σημασιολογικό επίπεδο. Αντιστρόφως, οι αραιές μέθοδοι μπορούν να λειτουργήσουν επικουρικά στις πυκνές, παρέχοντας κάλυψη σε περιπτώσεις όπου εμφανίζονται σπάνιες λέξεις ή οντότητες στο ερώτημα, κάτι που οι πυκνοί ανακτητές μπορεί να μην χειριστούν καλά όταν δεν έχουν επαρκή μάθηση πάνω σε αυτές. Έρευνες δείχνουν ότι τέτοιες υβριδικές προσεγγίσεις βελτιώνουν τη robustness του συστήματος, αυξάνοντας τόσο την ανάκληση (recall) όσο και την ακρίβεια (precision) της ανάκτησης υπό ευρύ φάσμα συνθηκών. [13]

Μετασχηματισμός ή αναδιατύπωση ερωτήματος (Query Rewriting/Transformation): Πέρα από την ίδια τη μέθοδο ανάκτησης, σημαντικό ρόλο παίζει και το πώς διατυπώνεται το ερώτημα. Σε πολλές περιπτώσεις εφαρμόζονται τεχνικές βελτίωσης ή μετατροπής του αρχικού query πριν αυτό περαστεί στον retriever, με σκοπό να καταστεί πιο αποτελεσματική η αναζήτηση. Μια απλή μορφή είναι η αναδιατύπωση του ερωτήματος: το σύστημα μπορεί να επεκτείνει το ερώτημα προσθέτοντας συνώνυμα ή διευκρινίσεις, ή να το μετασχηματίσει σε μια πιο "ανακτήσιμη" μορφή (π.χ. μετατρέποντας μια ερώτηση σε δήλωση) . Πιο προηγμένες προσεγγίσεις παράγουν ένα υποθετικό έγγραφο ή απάντηση χρησιμοποιώντας το ίδιο το LLM (γνωστό ως pseudo-response ή pseudo-document generation) και στη συνέχεια κάνουν

ανάκτηση συγκρίνοντας αυτήν την υποθετική απάντηση με τα πραγματικά έγγραφα, μια τεχνική που εφαρμόζει, για παράδειγμα, το σύστημα HyDE (Hypothetical Document Embedding). Αυτές οι τεχνικές ενισχύουν την πληροφοριακή κάλυψη του ερωτήματος, διότι παρέχουν στον retriever περισσότερο πλαίσιο ή πολλαπλές διατυπώσεις, με αποτέλεσμα να βρίσκονται ευκολότερα οι σχετικές πληροφορίες ακόμα κι αν το αρχικό query ήταν ελλιπές ή διατυπωμένο με τρόπο που δεν ταίριαζε ακριβώς με τα έγγραφα. Έχει δειχθεί ότι μέθοδοι όπως το Query2Doc και το HyDE (που δημιουργούν pseudo-documents από το query) μπορούν να βελτιώσουν σημαντικά την απόδοση ανάκτησης. [29] Συγγενείς προσεγγίσεις περιλαμβάνουν και τον εμπλουτισμό του query με επιπλέον όρους (query expansion) ή την αυτόματη διάσπασή του σε απλούστερα ερωτήματα, όπου αυτό σχετίζεται με το επόμενο σημείο.

Ανάκτηση με πολλαπλά άλματα (Multi-hop Retrieval): Αφορά περιπτώσεις σύνθετων ερωτημάτων όπου μία μόνο πράξη ανάκτησης δεν επαρκεί για να συλλεγεί όλη η απαραίτητη πληροφορία. Σε αυτά τα σενάρια, το σύστημα RAG μπορεί να πραγματοποιήσει διαδοχικές ανακτήσεις σε πολλαπλά βήματα. Μία στρατηγική είναι η διάσπαση του σύνθετου ερωτήματος σε μικρότερα υπο-ερωτήματα (query decomposition): το αρχικό ερώτημα αναλύεται σε επιμέρους ερωτήματα που μπορούν να απαντηθούν ξεχωριστά, γίνεται ανάκτηση για το καθένα, και τέλος οι επιμέρους πληροφορίες συνδυάζονται για να παραχθεί η τελική απάντηση.[29] Αυτή η τεχνική εφαρμόζεται, για παράδειγμα, από μεθόδους όπως το ToC (Tree-of- Thought Clarification), που δημιουργεί ένα δέντρο διευκρινίσεων και υπο ερωτημάτων για να αντιμετωπίσει διαφορετικές πτυχές ενός ασαφούς ή πολυδιάστατου ερωτήματος . Μια άλλη προσέγγιση είναι η επαναληπτική ανάκτηση (iterative retrieval), όπου το σύστημα εναλλάσσει μεταξύ ανάκτησης και γενετικής διεργασίας σε έναν βρόχο: αρχικά ανακτά κάποια πληροφορία και δίνει μια πρόχειρη απάντηση ή ενδιάμεσο αποτέλεσμα, κατόπιν επαναδιατυπώνει ή εξειδικεύει το ερώτημα με βάση αυτό το αποτέλεσμα και ανακτά επιπλέον πληροφορίες, και συνεχίζει έτσι μέχρι να συγκεντρωθεί αρκετό γνώση για μια τελική απάντηση.[13] Αυτού του τύπου οι πολύ-βηματικές ανακτήσεις είναι κρίσιμες για ερωτήσεις που απαιτούν αλυσιδωτή λογική ή συνδυασμό γνώσεων από πολλαπλές πηγές. Με τον iterative ή multi-hop τρόπο, το RAG μπορεί να εξερευνήσει ευρύτερα τον χώρο γνώσης, βελτιώνοντας τη πληρότητα και ορθότητα της απάντησης σε σύνθετα προβλήματα, αν και απαιτεί πιο περίπλοκο συντονισμό μεταξύ retriever και generator. Σε αυτές τις περιπτώσεις είναι αρκετά ωφέλιμη η αξιοποίηση Knowledge Graphs για την ανάκτηση γνώσεων.

2.4 Naive RAG

Η Naive εκδοχή του RAG ήταν η πρώτη και πιο απλή μορφή προσέγγισης της ιδέας του RAG, η οποία εμφανίστηκε αμέσως μετά την ευρεία διάδοση μοντέλων όπως το GPT3 και το GPT4. Στη βασική αυτή προσέγγιση, το σύστημα ακολουθεί μια παραδοσιακή διαδικασία «ανάκτησης-ανάγνωσης» (Retrieve-Read) για την παραγωγή απαντήσεων [7].

2.4.1 Βασικά βήματα

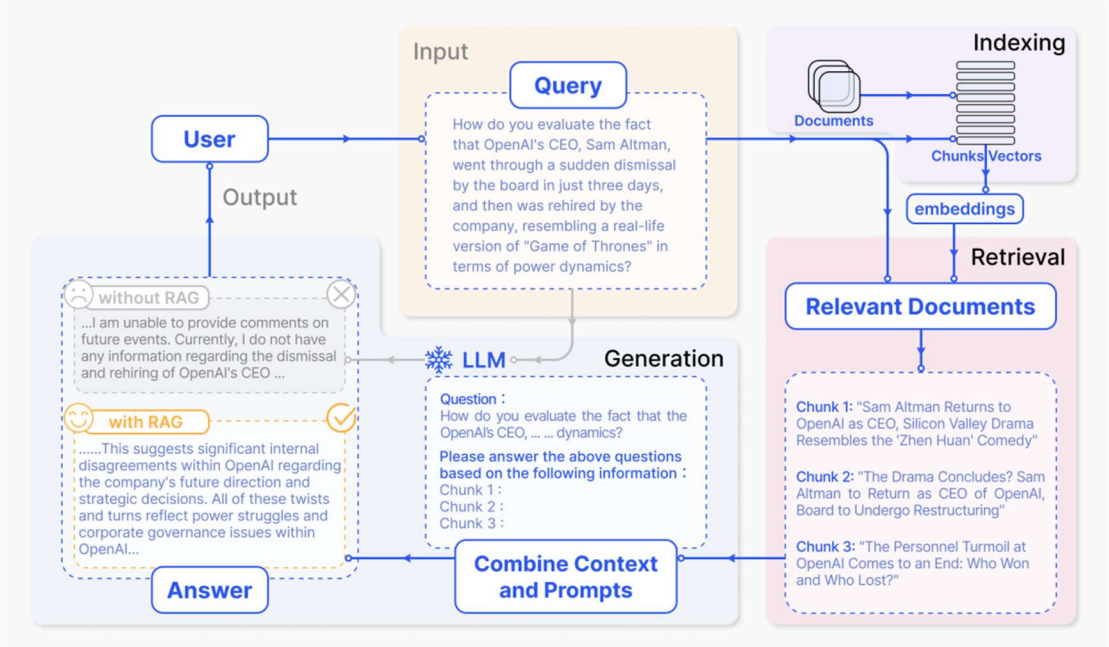
Ευρετηρίαση (Indexing): Τα διαθέσιμα έγγραφα αρχικά επεξεργάζονται και διασπώνται σε μικρότερα τμήματα κειμένου (text chunks), λόγω του περιορισμού που υπάρχει στο context window των LLM. Κάθε τμήμα κειμένου καθαρίζεται από μορφοποιήσεις και μετατρέπεται σε απλό κείμενο, και στη συνέχεια μετατρέπεται σε διανυσματική αναπαράσταση μέσω ενός μοντέλου εξαγωγής χαρακτηριστικών (embedding model). Τα διανύσματα αυτά αποθηκεύονται σε μια βάση δεδομένων διανυσμάτων (vector database) που λειτουργεί ως ευρετήριο. Η επιλογή κατάλληλου μεγέθους «κομματιού» κειμένου είναι κρίσιμη: πολύ μεγάλα τμήματα περιέχουν πλούσιο πλαίσιο αλλά δυσχεραίνουν την ανάκτηση, ενώ πολύ μικρά μπορεί να αποσπούν πληροφορίες εκτός συμφραζομένων. Η διαδικασία ευρετηρίασης επομένως στοχεύει σε μια ισορροπία, διασφαλίζοντας ότι τα τμήματα είναι αρκετά μικρά για αποδοτική αναζήτηση αλλά και αρκετά περιεκτικά νοηματικά. [13]

Ανάκτηση (Retrieval): Όταν διατυπώνεται ένα ερώτημα χρήστη, το σύστημα το μετατρέπει επίσης σε διάνυσμα (χρησιμοποιώντας το ίδιο μοντέλο ενσωμάτωσης που χρησιμοποιήθηκε και για τα έγγραφα). Μέσω ενός μέτρου ομοιότητας (π.χ. συνημιτονική απόσταση cosine) συγκρίνονται το διάνυσμα του ερωτήματος με τα διανύσματα όλων των αποθηκευμένων τμημάτων κειμένου. Το σύστημα ανακτά τα Top-K πιο σχετικά τμήματα, δηλαδή εκείνα με τη μεγαλύτερη σημασιολογική συνάφεια προς το ερώτημα. Τα τμήματα αυτά θεωρούνται το επεκταμένο πλαίσιο γνώσης για το μοντέλο και επισυνάπτονται στο ερώτημα.

Γεννήτρια Απάντησης (Generation): Το αρχικό ερώτημα μαζί με τα ανακτηθέντα αποσπάσματα κειμένου συνενώνονται σε μια ενιαία προτροπή (prompt) προς το LLM, το οποίο καλείται να παραγάγει την τελική απάντηση . Κατά τη φάση αυτή, το LLM μπορεί είτε να αντλήσει από τις δικές του γνώσεις (τις παραμέτρους του) είτε , ιδανικά, να βασιστεί αποκλειστικά στις παρεχόμενες πληροφορίες από τα έγγραφα ώστε η απάντηση να είναι τεκμηριωμένη. Σε περιπτώσεις συστημάτων διαλόγου, τυχόν προηγούμενο ιστορικό

συνομιλίας μπορεί επίσης να συμπεριληφθεί στην προτροπή, επιτρέποντας αλληλεπιδράσεις πολλών γύρων με διατήρηση συμφραζομένων.

Τα βασικά βήματα που περιεγράφηκαν παραπάνω αναπαρίστανται στην *Εικόνα 3*.



Εικόνα 3 - Αναπαράσταση της Naive RAG pipeline [13]

2.4.2 Περιορισμοί και Προκλήσεις της Naive RAG

Παρά τη χρησιμότητά του, το Naive RAG εμφανίζει αρκετές αδυναμίες που επηρεάζουν τόσο την ποιότητα ανάκτησης όσο και τη γενεσιουργό διαδικασία:

Προκλήσεις Ανάκτησης: Το στάδιο της ανάκτησης πάσχει συχνά από ζητήματα precision και recall. Το σύστημα ενδέχεται να ανακτήσει αποσπάσματα άσχετα ή μη ευθυγραμμισμένα με το ερώτημα, παραπλανώντας έτσι το μοντέλο. Αντίστροφα, υπάρχει περίπτωση να μην ανακτηθούν κρίσιμες πληροφορίες που βρίσκονται διασκορπισμένες σε άλλα τμήματα κειμένου, ειδικά όταν το ερώτημα απαιτεί σύνθεση πολλαπλών πηγών. Η εξάρτηση από την αρχική διατύπωση του ερωτήματος και η ανάκτηση μόνο μέσω σημασιολογικής ομοιότητας σημαίνει ότι αν το ερώτημα δεν "ταιριάξει" επαρκώς με σχετικές διατυπώσεις στα έγγραφα, σημαντικό περιεχόμενο μπορεί να χαθεί [30], [31].

Προβλήματα Γεννήτριας: Κατά τη δημιουργία της απάντησης, το LLM μπορεί να αντιμετωπίσει το φαινόμενο της ψευδοπληροφόρησης (hallucination), να παράγει δηλαδή πληροφορίες που δεν υποστηρίζονται από τα ανακτηθέντα χωρία. Αυτό συμβαίνει όταν το μοντέλο "μαντεύει" με βάση τις παραμέτρους του, ειδικά αν τα παρεχόμενα δεδομένα δεν επαρκούν ή δεν είναι αξιόπιστα. Επίσης, η ποιότητα της απάντησης μπορεί να υποβαθμιστεί από άσχετο ή τοξικό περιεχόμενο που πιθανώς περιλαμβάνεται στα ανακτηθέντα κείμενα, ή από εγγενείς προκαταλήψεις του μοντέλου [32]. Όλα αυτά υπονομεύουν την αξιοπιστία και εγκυρότητα των παραγόμενων απαντήσεων.

Δυσκολίες Ενσωμάτωσης Πληροφορίας: Η ενσωμάτωση των ανακτηθέντων δεδομένων στο τελικό κείμενο δεν είναι πάντα ομαλή. Συχνά, το μοντέλο δυσκολεύεται να συνδυάσει πληροφορίες από πολλές πηγές, οδηγώντας σε απαντήσεις αποσπασματικές ή ασύνδετες. Επιπλέον, εάν διαφορετικά αποσπάσματα περιέχουν πλεονάζουσες ή επαναλαμβανόμενες πληροφορίες, η απάντηση μπορεί να γίνει φλύαρη και επαναληπτική. Μια ακόμη πρόκληση είναι να κριθεί ποια από τα ανακτηθέντα στοιχεία είναι όντως σημαντικά για το ερώτημα, το μοντέλο δυσκολεύεται να αξιολογήσει τη σχετικότητα και βαρύτητα κάθε κομματιού πληροφορίας και να διατηρήσει ένα ενιαίο ύφος και τόνο στην τελική απάντηση [33].

Περιορισμός Μονοβηματικής Ανάκτησης: Σε ιδιαίτερα σύνθετα ή πολυδιάστατα ερωτήματα, μία μόνο φορά ανάκτησης ίσως δεν επαρκεί για να συγκεντρωθεί όλη η απαραίτητη γνώση. Το Naive RAG εκτελεί την ανάκτηση μόνο μια φορά, πριν από τη δημιουργία της απάντησης. Αν το αρχικό ερώτημα δεν διατυπωθεί πλήρως ή αν απαιτείται ενδιάμεση συλλογιστική (π.χ. πολλαπλά βήματα εύρεσης στοιχείων), το σύστημα ενδέχεται να αποτύχει να βρει όλα τα απαραίτητα δεδομένα με την πρώτη προσπάθεια. Δεν υπάρχει μηχανισμός ανατροφοδότησης (feedback) ή περαιτέρω διερώτησης της βάσης γνώσης με βάση ενδιάμεσα συμπεράσματα [13], [33].

Κίνδυνος "παπαγαλίας" του Περιεχομένου: Τέλος, έχει παρατηρηθεί ότι τα γενεσιουργά μοντέλα ενίοτε στηρίζονται υπέρμετρα στα ανακτηθέντα κείμενα, παρουσιάζοντας την απάντηση σαν μια απλή συρραφή ή επανάληψη των αποσπασμάτων αντί να τα συγκεράσουν δημιουργικά [34]. Αυτό μπορεί να οδηγήσει σε απαντήσεις που, αν και περιέχουν σωστά στοιχεία, δεν προσθέτουν διορατικότητα ή σύνθεση, απλώς επαναλαμβάνουν πληροφορίες χωρίς επεξηγήσεις ή συμπεράσματα .

Τα παραπάνω προβλήματα ανέδειξαν την ανάγκη για βελτιωμένες προσεγγίσεις RAG. Πράγματι, η έρευνα προχώρησε πέρα από το Naive RAG, προτείνοντας πιο εξελιγμένες τεχνικές (Advanced RAG) που βελτιστοποιούν επιμέρους στάδια, καθώς και υβριδικά ή αρθρωτά συστήματα (Modular RAG) που εισάγουν νέα υποσυστήματα και επαναληπτικές διαδικασίες αναζήτησης. Στο επόμενο κεφάλαιο θα εξετάσουμε μια σύγχρονη τέτοια τεχνική, το RAPTOR RAG, που αντιμετωπίζει ορισμένες από τις παραπάνω αδυναμίες μέσω μιας δενδρικής, αναδρομικής προσέγγισης στην ανάκτηση και περίληψη πληροφοριών.

2.5 RAPTOR RAG

Η τεχνική RAPTOR (Recursive Abstractive Processing for Tree-Organized Retrieval) [35] αποτελεί μια πρόσφατη εξέλιξη (2024) στον χώρο του RAG, σχεδιασμένη ειδικά για να υπερβεί τους περιορισμούς της παραδοσιακής Naive προσέγγισης. Οι συμβατικές μέθοδοι RAG, όπως είδαμε, ανακτούν μόνο αποσπάσματα τοπικού χαρακτήρα, μικρά συνεχόμενα τμήματα κειμένου, από τη βάση γνώσης, γεγονός που περιορίζει την ολιστική κατανόηση του συνολικού περιεχομένου ενός μεγάλου εγγράφου. Αυτό έχει ως συνέπεια το απλό RAG να δυσκολεύεται ιδιαιτέρως σε σύνθετα ερωτήματα πολλαπλών βημάτων (multi-hop queries), όπου η απαιτούμενη πληροφορία είναι διασπαρμένη ή απαιτεί συσχέτιση πολλών διαφορετικών σημείων ενός ή περισσότερων εγγράφων. [1] Για παράδειγμα, σε μια ερώτηση που ζητά σύγκριση ή συνδυασμό γνώσεων από διάφορες ενότητες ενός κειμένου, το Naive RAG πιθανώς θα ανακτήσει μόνο μία-δύο σχετικές παραγράφους και θα αποτύχει να συνθέσει πλήρη απάντηση. Το RAPTOR RAG προτάθηκε για να αντιμετωπίσει αυτή την αδυναμία, εισάγοντας μια δενδρική δομή ανάκτησης και περίληψης που επιτρέπει στο σύστημα να συλλάβει τόσο τις λεπτομέρειες όσο και τη συνολική εικόνα ενός μεγάλου κειμένου. [35]

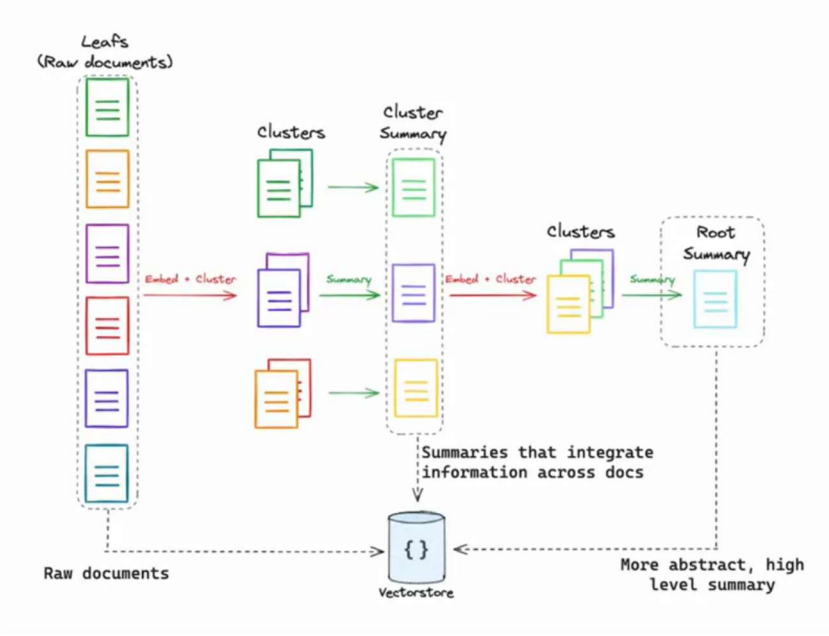
2.5.1 Μεθοδολογία RAPTOR – Δενδροειδής Οργάνωση Γνώσης

Αντί για μια επίπεδη συλλογή διανυσμάτων από αυτόνομα τμήματα κειμένου, το RAPTOR οργανώνει τα δεδομένα ιεραρχικά σε μορφή δέντρου. Η κατασκευή του δέντρου γίνεται σε στάδιο προ-επεξεργασίας και συνοψίζεται στα εξής βήματα:

- Αρχικά, ολόκληρο το έγγραφο (ή το σύνολο εγγράφων) διασπάται σε σχετικά μικρά τμήματα κειμένου και ενσωματώνονται διανυσματικά, όπως και στο Naive RAG.
- Στη συνέχεια, αντί τα διανύσματα αυτά να αποθηκευτούν απλώς σε μια λίστα, το σύστημα εφαρμόζει έναν αλγόριθμο ομαδοποίησης (clustering): τα τμήματα που παρουσιάζουν θεματική/σημασιολογική συνάφεια μεταξύ τους ανιχνεύονται και

ομαδοποιούνται σε συστάδες. Έτσι, αν ένα έγγραφο έχει, για παράδειγμα, ενότητες με κοινό θέμα, αυτά τα κομμάτια πιθανώς θα βρεθούν στην ίδια ομάδα.

- Για κάθε τέτοια ομάδα κειμένων, ένα μεγάλο γλωσσικό μοντέλο χρησιμοποιείται ώστε να παράξει μια περίληψη που συμπυκνώνει το περιεχόμενό τους. Αυτή η περίληψη είναι ένα νέο τεχνητό κείμενο που αποσκοπεί να αναπαραστήσει συλλογικά τις πληροφορίες όλων των επιμέρους τμημάτων της ομάδας. Πρόκειται δηλαδή για συνοπτική, αφαιρετική αναπαράσταση της ομάδας δεδομένων, όχι απλώς επιλογή προτάσεων, αλλά δημιουργία νέου συνοπτικού κειμένου (abstractive summary).
- Αφού δημιουργηθούν περιλήψεις για όλες τις ομάδες του πρώτου επιπέδου, αυτές οι περιλήψεις με τη σειρά τους μετατρέπονται σε διανύσματα και μπορούν να αντιμετωπιστούν ως «δεύτερης τάξης» τμήματα κειμένου. Κατόπιν, επαναλαμβάνεται η διαδικασία: οι περιλήψεις αυτές ομαδοποιούνται περαιτέρω βάσει ομοιότητας και παράγονται περιλήψεις υψηλότερου επιπέδου. Η διαδικασία συνεχίζεται αναδρομικά, ενσωμάτωση, clustering, περίληψη, ξανά και ξανά μέχρι να μην είναι δυνατή άλλη ομαδοποίηση ή έως ότου κατασκευαστεί μια μοναδική τελική περίληψη που αντιπροσωπεύει ολόκληρο το σύνολο των δεδομένων. Το αποτέλεσμα είναι ένα πολυεπίπεδο δενδρικό ευρετήριο περιλήψεων, όπου:
    - τα φύλλα (leaves) του δέντρου είναι τα αρχικά τμήματα κειμένου,
    - οι ενδιάμεσοι κόμβοι είναι περιλήψεις ομάδων τμημάτων,
    - και η ρίζα του δέντρου είναι μια σύνοψη που καλύπτει το σύνολο του περιεχομένου [35]



Εικόνα 4 - Αναπαράσταση RAPTOR pipeline

Με αυτή τη δομή, *Εικόνα 4,* το RAPTOR επιτυγχάνει μια οργανωμένη ιεραρχία γνώσης. Κάθε επίπεδο του δέντρου αντιστοιχεί σε διαφορετικό βαθμό αφαίρεσης: στα κατώτερα επίπεδα βρίσκονται συγκεκριμένες λεπτομέρειες, ενώ όσο ανέρχεται κανείς προς τη ρίζα, τόσο πιο συνοπτική και γενική γίνεται η πληροφορία . Είναι σημαντικό να τονιστεί ότι όλες οι περιλήψεις παράγονται με αφαιρετικό τρόπο (abstractive), αξιοποιώντας την ικανότητα του LLM να διατυπώνει περιεκτικό λόγο, αυτό σημαίνει ότι ενδέχεται να εκφράζουν ρητά σχέσεις και συμπεράσματα που υπονοούνται στα δεδομένα, δίνοντας στο μοντέλο μια συνεκτική "εικόνα" του τι λέει κάθε ομάδα κειμένων. [35]

2.5.2 Ανάκτηση και Απόκριση μέσω του Δέντρου

Αφού δημιουργηθεί το παραπάνω πολυεπίπεδο ευρετήριο, το στάδιο της ανάκτησης πληροφορίας (κατά την απάντηση σε ένα ερώτημα) εκτελείται πλέον πάνω σε αυτό το δέντρο και όχι απευθείας στα ωμά δεδομένα. Συγκεκριμένα, όταν δοθεί ένα ερώτημα:

- Το ερώτημα μετατρέπεται σε διάνυσμα και συγκρίνεται με τα διανύσματα όλων των κόμβων του δέντρου (όχι μόνο των φύλλων). Έτσι, το σύστημα μπορεί να αναζητήσει τη συνάφεια του ερωτήματος τόσο με πολύ ειδικές πληροφορίες (φύλλα) όσο και με συνοπτικές έννοιες (περιλήψεις υψηλότερων κόμβων).
- Η διαδικασία ανάκτησης μπορεί να φέρει ως αποτέλεσμα ένα σύνολο κόμβων διαφόρων επιπέδων που θεωρούνται σχετικοί. Για παράδειγμα, αν το ερώτημα είναι γενικό, ίσως μια περίληψη ανώτερου επιπέδου να έχει υψηλή συνάφεια και να επιλεγεί. Αν είναι συγκεκριμένο, πιθανώς κάποια φύλλα ή χαμηλού επιπέδου περιλήψεις να ταιριάξουν καλύτερα. Το κλειδί είναι ότι το RAPTOR μπορεί να αξιοποιήσει πληροφορία σε πολλαπλές κλίμακες: τόσο λεπτομερειακά στοιχεία όσο και συνολικές περιλήψεις, ανάλογα με το τι απαιτεί το ερώτημα.
- Στη συνέχεια, οι επιλεγμένοι κόμβοι (που αποτελούν το ανακτηθέν σύνολο γνώσης) χρησιμοποιούνται για τη διαμόρφωση της προτροπής προς το LLM, μαζί φυσικά με το αρχικό ερώτημα , όπως και στο παραδοσιακό RAG. Η παραγωγή της απάντησης γίνεται από το LLM έχοντας στη διάθεσή του τόσο συγκεκριμένα αποσπάσματα κειμένου όσο και περιλήψεις ευρύτερων ενοτήτων, επιτρέποντάς του να ενσωματώσει πληροφορίες από διαφορετικά σημεία του εγγράφου και σε διαφορετικό βαθμό λεπτομέρειας . Με τον τρόπο αυτό, το σύστημα έχει τη δυνατότητα να κατανοήσει και να χρησιμοποιήσει το πλαίσιο ολόκληρου του εγγράφου, κάτι ανέφικτο στην Naive προσέγγιση όπου κάθε ανακτηθέν κομμάτι είναι απομονωμένο.

2.5.3 Διαφορές και Πλεονεκτήματα έναντι του Παραδοσιακού RAG

Η προσέγγιση RAPTOR διαφέρει ριζικά από το απλό RAG και προσφέρει πολλαπλά οφέλη:

Ολιστική κατανόηση: Ενώ το Naive RAG περιορίζεται στην τοπική πληροφορία, το RAPTOR διατηρεί τη συνολική δομή και θεματική του εγγράφου μέσω των περιλήψεων. Αυτό σημαίνει ότι μπορεί να απαντήσει σε γενικές ή σφαιρικές ερωτήσεις για ένα έγγραφο , π.χ. «ποιο είναι το κύριο θέμα;», κάτι που το επίπεδο RAG αδυνατεί να κάνει επαρκώς. Ο συνδυασμός λεπτομερειών και περιλήψεων δίνει στο μοντέλο μια σφαιρική "αντίληψη" του περιεχομένου.

Πολυ-βηματική συλλογιστική: Χάρη στη δενδρική δομή, το RAPTOR μπορεί να εντοπίσει συσχετισμένες πληροφορίες που βρίσκονται διασκορπισμένες και να τις συνδυάσει. Στις περιπτώσεις ερωτήσεων πολλαπλών βημάτων ή συνθετικών ερωτημάτων (που απαιτούν multi-hop reasoning), υπερτερεί σημαντικά του παραδοσιακού RAG, το οποίο όπως αναφέρθηκε αποτυγχάνει όταν η απάντηση δεν βρίσκεται αυτούσια σε ένα μόνο απόσπασμα. Το RAPTOR ουσιαστικά λειτουργεί σαν να "διαβάζει" το έγγραφο σε βάθος, πρώτα συγκεντρώνοντας τα επιμέρους στοιχεία και μετά βλέποντας τη μεγάλη εικόνα, με αποτέλεσμα να απαντά με μεγαλύτερη ακρίβεια σε σύνθετα ζητήματα.

Μείωση πληροφοριακού θορύβου: Οι αφαιρετικές περιλήψεις λειτουργούν ως φίλτρο γνώσης. Πολλές άχρηστες ή λιγότερο σχετικές λεπτομέρειες αποκλείονται στις περιλήψεις υψηλότερου επιπέδου, ενώ διατηρούνται τα βασικά σημεία. Έτσι, όταν το ερώτημα είναι γενικό, η ανάκτηση μπορεί να φέρει κυρίως περιλήψεις αντί για ωμά αποσπάσματα, ελαχιστοποιώντας τον θόρυβο και την περίσσεια πληροφορίας που θα έπρεπε να επεξεργαστεί το LLM. Αυτό καθιστά την τελική απάντηση πιο ευθύγραμμη και συγκεντρωμένη στα σημαντικά.

Βελτιωμένη απόδοση: Σύμφωνα με τα αποτελέσματα που αναφέρονται στην εργασία όπου προτάθηκε, το RAPTOR πέτυχε σαφώς καλύτερες επιδόσεις από τις παραδοσιακές μεθόδους RAG σε μια σειρά από δοκιμασίες . Για παράδειγμα, σε ερωτήσεις που απαιτούν περίπλοκη, πολυ-βηματική συλλογιστική (όπως ορισμένα datasets ερωταποκρίσεων κατανόησης κειμένου), το RAPTOR κατέγραψε state-of-the-art αποτελέσματα. Χαρακτηριστικά, αναφέρεται ότι συνδυάζοντας τον μηχανισμό ανάκτησης του RAPTOR με την ισχύ ενός μεγάλου μοντέλου όπως το GPT-4, επιτεύχθηκε βελτίωση της ακρίβειας κατά 20% (απόλυτη

τιμή) στο σύνολο δοκιμών QuALITY, σε σχέση με το προηγούμενο καλύτερο σύστημα . Αυτή η εντυπωσιακή άνοδος υπογραμμίζει το όφελος της δενδρικής αναπαράστασης: το μοντέλο μπόρεσε να αξιοποιήσει πληρέστερα το περιεχόμενο μεγάλων κειμένων, αποφεύγοντας κενά πληροφορίας και λάθη κατανόησης. [35]

2.6 Knowledge Graphs (KGs)

Παρά την αποδοτικότητα τεχνικών όπως το RAPTOR, οι οποίες ενισχύουν τη συνοπτική ικανότητα των LLMs μέσω ιεραρχικής επεξεργασίας πολυάριθμων αποσπασμάτων κειμένου, η αναπαράσταση της γνώσης παραμένει αποσπασματική και συνήθως περιορίζεται σε αδόμητο κείμενο. Οι τεχνικές αυτές στηρίζονται στην αλληλουχία φυσικής γλώσσας χωρίς να αξιοποιούν πλήρως τις δυνατότητες εννοιολογικής οργάνωσης και λογικής συσχέτισης που παρέχουν οι Γράφοι Γνώσης (Knowledge Graphs). Οι Γράφοι Γνώσης προσφέρουν μια συμπαγή, ερμηνεύσιμη και εξελισσόμενη αναπαράσταση της γνώσης, καθιστώντας τους ιδανικούς για εφαρμογές που απαιτούν υψηλό βαθμό ακρίβειας και εννοιολογικής πλοήγησης. Η μετάβαση από καθαρά κειμενοκεντρικά μοντέλα σε υβριδικές αρχιτεκτονικές που συνδυάζουν LLMs και γράφους γνώσης (όπως το KG-RAG) καθιστά εφικτή την ανάπτυξη συστημάτων που είναι ταυτόχρονα ισχυρά σε λογισμό και λιγότερο επιρρεπή σε παραισθήσεις. Στη συνέχεια, εξετάζουμε τη θεωρητική θεμελίωση των Γράφων Γνώσης και τις σύγχρονες μεθόδους ενσωμάτωσής τους με μεγάλες γλωσσικές μοντέλα στο πλαίσιο της RAG αρχιτεκτονικής.

2.6.1 Τι είναι τα Knowledge Graphs (KGs)

Τα Knowledge Graphs (KGs), ή «Γράφοι Γνώσης», αποτελούν δομές δεδομένων που αναπαριστούν γνώση με μορφή γραφημάτων από οντότητες και σχέσεις. Συγκεκριμένα, ένα KG ορίζει ένα σύνολο από οντότητες (π.χ. πρόσωπα, τοποθεσίες, έννοιες) οι οποίες συνδέονται μεταξύ τους μέσω σχέσεων που φέρουν σημασιολογικό νόημα.[7] Κάθε γεγονός εκφράζεται συνήθως ως τριάδα μορφής (υποκείμενο, κατηγόρημα, αντικείμενο), για παράδειγμα (Albert Einstein, WinnerOf, Nobel Prize), σύμφωνα με το πρότυπο RDF. Οι τριάδες αυτές μπορούν να θεωρηθούν ακμές ενός γράφου όπου οι κόμβοι είναι οι οντότητες και οι ακμές οι τύποι σχέσεων που τις συνδέουν. Σε πολλά KGs ακολουθείται επίσης το μοντέλο του property graph, δηλαδή κάθε κόμβος ή σχέση μπορεί να φέρει επιπρόσθετες ιδιότητες/γνωρίσματα (attributes) που περιγράφουν λεπτομερέστερα την οντότητα ή τη σχέση. Έτσι, ένα KG μπορεί να ιδωθεί και ως σημασιολογική βάση γνώσης: η γραφική του δομή επιτρέπει την οπτική απεικόνιση

συνδέσεων, ενώ το πλούσιο λεξιλόγιο τύπων και ιδιοτήτων επιτρέπει την εξαγωγή λογικών συμπερασμάτων (reasoning) πάνω στις αποθηκευμένες γνώσης. Στην πράξη, οι όροι «knowledge graph» και «βάση γνώσης» χρησιμοποιούνται συχνά εναλλακτικά.

2.6.2 Βασικά χαρακτηριστικά των KGs

Ένα γνώρισμα των KGs είναι η σημασιολογική πληρότητα και η δομημένη μορφή τους. Τα δεδομένα οργανώνονται με σαφώς ορισμένους τύπους σχέσεων και οντοτήτων, διευκολύνοντας την κατανόηση από μηχανές και την εκτέλεση πολύπλοκων ερωτημάτων. Επιπλέον, ένα KG μπορεί να ενσωματώνει ετερογενείς πηγές γνώσης (π.χ. συνδυάζοντας δεδομένα από κείμενα, βάσεις δεδομένων και λεξικά) και να υποστηρίζει πολύγλωσση πληροφορία. Για παράδειγμα, το Wikidata είναι ένα ελεύθερο συνεργατικό υπόβαθρο γνώσης που δημιουργείται και συντηρείται από εθελοντές, με στόχο την κεντρική διαχείριση των δεδομένων της Wikipedia, καλύπτει εκατοντάδες διαφορετικές γλώσσες και περιέχει εκατομμύρια οντότητες και γεγονότα. Αντίστοιχα, το ConceptNet αποτελεί ένα ανοικτό πολύγλωσσο γράφημα γενικών γνώσεων, εστιασμένο κυρίως στην κοινή λογική (commonsense) γνώση. Σε αντίθεση με τα εγκυκλοπαιδικά knowledge graphs όπως το Wikidata, το ConceptNet περιέχει συχνά και μη ντετερμινιστική γνώση, κάθε σχέση συνοδεύεται από έναν βαθμό εμπιστοσύνης (confidence score) που υποδεικνύει πόσο αξιόπιστο θεωρείται το δεδομένο γεγονός. [7]

2.6.3 Πλεονεκτήματα

Δομημένη αναπαράσταση & σημασιολογικός εμπλουτισμός: Τα KGs αποθηκεύουν τη γνώση σε δομημένη μορφή γραφήματος, γεγονός που μειώνει τον θόρυβο και διευκολύνει την κατανόηση πολύπλοκων σχέσεων μεταξύ εννοιών. Ο σαφής σημασιολογικός καθορισμός των κόμβων/ακμών (οντολογίες) επιτρέπει σε συστήματα τεχνητής νοημοσύνης να αντιληφθούν βαθύτερα το νόημα ενός ερωτήματος ή μιας πληροφορίας, σε αντίθεση με μια απλή συλλογή κειμένων. [28] Έτσι, τα KGs διευκολύνουν το πολυ-βηματικό reasoning (π.χ. multi-hop ερωτήσεις) αφού η σχετική πληροφορία μπορεί να ανακτηθεί μέσω συνδεδεμένων οντοτήτων και όχι μεμονωμένων εγγράφων.

Λογικός έλεγχος και επαγωγή: Δεδομένου ότι τα γεγονότα σε ένα KG έχουν φορμαλισμένη σημασιολογία, είναι δυνατή η εφαρμογή λογικών κανόνων και μεθόδων inference πάνω τους. Αυτό σημαίνει ότι ένα σύστημα μπορεί να εξάγει νέες γνώσεις από υπάρχουσες (π.χ. αν $A - isSubclassOf - B$ και $B - isSubclassOf - C$,

συμπεραίνουμε $A - isSubclassOf - C$ και να ελέγξει τη συνέπεια των δεδομένων. Τέτοιες ικανότητες reasoning είναι δυσκολότερο να επιτευχθούν σε ανοργάνωτα κείμενα.

Δυναμική ενημέρωση και εξέλιξη: Τα περισσότερα knowledge graphs μπορούν να ενημερώνονται συνεχώς με νέες τριάδες, επιτρέποντας την ενσωμάτωση της πιο πρόσφατης γνώσης σε πραγματικό χρόνο.[28] Αυτό δίνει ένα συγκριτικό πλεονέκτημα έναντι των στατικών εκπαιδευμένων γλωσσικών μοντέλων: η γνώση σε ένα KG μπορεί να επεκταθεί ή να διορθωθεί απλά εισάγοντας ή αφαιρώντας δεδομένα, χωρίς να απαιτείται επανεκπαίδευση ενός μοντέλου από την αρχή.

2.6.4 Μειονεκτήματα

Πληρότητα & κάλυψη της γνώσης: Οι πραγματικοί γράφοι γνώσης είναι συχνά ελλιπείς, δεν περιέχουν όλα τα πιθανά γεγονότα, και μπορεί να περιέχουν θορυβώδεις ή λανθασμένες πληροφορίες, ιδιαίτερα όταν κατασκευάζονται αυτόματα.[36] Για παράδειγμα, οντολογίες που εξορύσσονται αυτόματα από τον ιστό (π.χ. NELL) ενδέχεται να περιλαμβάνουν λανθασμένες συσχετίσεις. Η ανοιχτή φύση (open-world assumption) των KGs σημαίνει ότι η απουσία μιας τριάδας δεν συνεπάγεται απαραίτητα ψευδές γεγονός, γεγονός που δυσχεραίνει την εξαγωγή αρνητικών συμπερασμάτων.

Κόστος κατασκευής & συντήρησης: Η δημιουργία και ενημέρωση ενός πλούσιου KG απαιτεί σημαντική προσπάθεια. Ορισμένα, όπως το Wikidata, βασίζονται σε ανθρώπινη επιμέλεια (crowdsourcing), κάτι που μπορεί να μην επεκτείνεται εύκολα σε όλους τους τομείς γνώσης. Άλλα πάλι απαιτούν αυτόματες τεχνικές εξόρυξης γνώσης (Knowledge Extraction) από κείμενα, που όμως μπορεί να σφάλουν και να εισαγάγουν σφάλματα ή αβεβαιότητες.

Ετερογένεια & πολυπλοκότητα: Καθώς τα KGs ενσωματώνουν δεδομένα από ποικίλες πηγές, συχνά αντιμετωπίζουν προβλήματα ευθυγράμμισης οντοτήτων (entity alignment) μεταξύ διαφορετικών συνόλων δεδομένων ή σχημάτων. Η πολυπλοκότητα της γραφικής δομής (με εκατοντάδες εκατομμύρια ακμές/κόμβους σε μεγάλα KGs) καθιστά δύσκολη την κλιμάκωση ορισμένων αλγορίθμων επεξεργασίας και reasoning, όπως επίσης και την αποδοτική αποθήκευση/ανάκτηση πληροφοριών.

Συνοψίζοντας, τα Knowledge Graphs προσφέρουν ένα εύρωστο πλαίσιο αναπαράστασης γνώσης που ευνοεί την λογική επεξεργασία και τη σύνδεση πληροφοριών. Χρησιμοποιούνται

ήδη σε πλήθος εφαρμογών, από ερωταποκρίσεις και διαλόγους μέχρι συστάσεις. Παραδείγματα όπως το Google Knowledge Graph και το Microsoft Satori έχουν αναδείξει την πρακτική τους αξία στην ενίσχυση υπηρεσιών που απαιτούν κατανόηση κοινής γνώσης. [7] Ωστόσο, η ατελής φύση τους και η δυσκολία ενσωμάτωσης νέων ή εξειδικευμένων γνώσεων παραμένουν προκλήσεις, ειδικά όσο το πεδίο της Τεχνητής Νοημοσύνης εξελίσσεται.

2.7 Retrieval-Augmented Generation με Knowledge Graphs (KG-RAG)

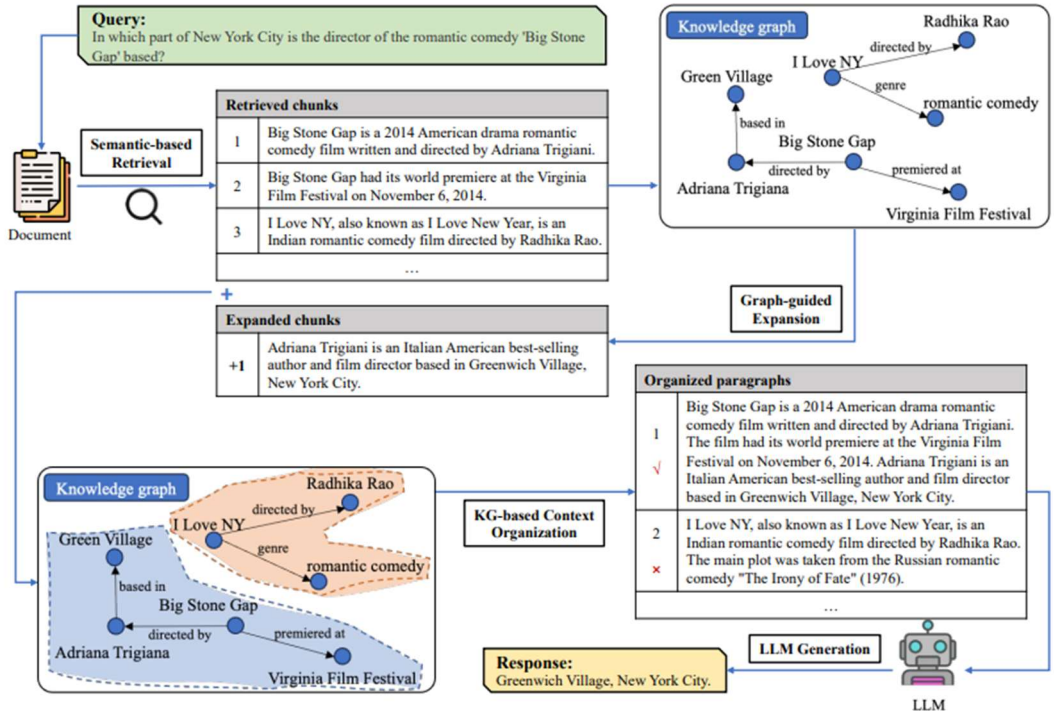
Η σύζευξη των knowledge graphs με τα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models, LLMs) αναδύεται ως μια πολλά υποσχόμενη προσέγγιση για την αξιοποίηση του καλύτερου από δύο κόσμους: των συμβολικών αναπαραστάσεων γνώσης και των νευρωνικών μοντέλων κατανόησης γλώσσας. Τα LLMs (π.χ. GPT-3/4, BERT) έχουν εκπαιδευτεί σε τεράστιο όγκο κειμένων και διαθέτουν εκτεταμένη γνώση για τον κόσμο, αλλά συχνά αδυνατούν σε αυστηρά λογικούς ή πολυ-βηματικούς συλλογισμούς και τείνουν σε «ψευδαισθήσεις» (hallucinations) όταν τους λείπει μια πληροφορία. Αντίθετα, τα KGs προσφέρουν ακριβή και δομημένη γνώση και είναι καταλληλότερα για structured reasoning (π.χ. μπορούν ευκολότερα να χειριστούν αρνήσεις ή σύνθετα λογικά ερωτήματα) . Ως εκ τούτου, ο συνδυασμός τους στοχεύει στο να παντρέψει την ευρύτητα γνώσεων των LLMs με την λογική αυστηρότητα και διαφάνεια των KGs. [37]

Μια από τις πλέον πρακτικές μεθόδους ενσωμάτωσης των knowledge graphs σε συστήματα μεγάλων γλωσσικών μοντέλων είναι το λεγόμενο Retrieval-Augmented Generation (RAG) με χρήση γνώσης από KG, εν συντομία KG-RAG. Στα συστήματα RAG, επιχειρείται να βελτιωθεί η ακρίβεια και η επικαιρότητα των απαντήσεων ενός LLM παρέχοντάς του εξωτερικές πηγές γνώσης τη στιγμή που παράγει κείμενο.[28] Παραδοσιακά, αυτό υλοποιείται με ανάκτηση συναφών εγγράφων ή αποσπασμάτων κειμένου από μια βάση γνώσης (π.χ. Wikipedia) και προσθήκη τους ως μέρος του input στο LLM, ώστε η απόκρισή του να θεμελιώνεται σε πραγματικά δεδομένα και όχι μόνο στις παραμέτρους του μοντέλου. Στο πλαίσιο του KG-RAG, η εξωτερική γνώση δεν είναι απλώς μια συλλογή επίπεδων κειμένων, αλλά ένα ή περισσότερα γραφήματα γνώσης. Η αξιοποίηση δομημένης γνώσης στο RAG στοχεύει στην αντιμετώπιση ορισμένων αδυναμιών των κλασικών συστημάτων RAG: αυτά συχνά αγνοούν τις σχέσεις μεταξύ εννοιών επειδή χειρίζονται κάθε έγγραφο ανεξάρτητα, δυσκολεύονται σε ερωτήματα που απαιτούν συνδυαστική λήψη πληροφοριών (π.χ. multi-hop

queries), και ενδέχεται να επιστρέψουν εκτενή ή άσχετα κείμενα ως πλαίσιο, εισάγοντας θόρυβο. Ένα knowledge graph μπορεί να λειτουργήσει ως αντίδοτο, παρέχοντας δομημένη αποθήκευση πληροφοριών και ενισχυμένη κατανόηση των συσχετίσεων, όπως θα δούμε παρακάτω .

2.7.1 Δομή και φάσεις ενός KG-RAG pipeline

Συνήθως, ένας αλγόριθμος KG-RAG αποτελείται από διακριτά στάδια, *Εικόνα 5*, που εξασφαλίζουν ότι το LLM θα λάβει στο input του τα πλέον σχετικά και συμπυκνωμένα δεδομένα:



Εικόνα 5 - KG-RAG pipeline [38]

Ανάκτηση γνώσης από το γράφο (graph retrieval & spreading activation): Δεδομένου ενός ερωτήματος χρήστη, το σύστημα ξεκινά εντοπίζοντας στο knowledge graph τις πιο συναφείς οντότητες/κόμβους. Αυτό μπορεί να γίνει με ευθείς τρόπους (π.χ. αναζήτηση ενός ονόματος στον γράφο) ή με πιο έξυπνες μεθόδους, όπως η χρήση spreading activation. Το spreading activation είναι ένας γνωστικός μηχανισμός από την ψυχολογία, ο οποίος περιγράφει πώς οι άνθρωποι ανακαλούν σχετικές έννοιες διαμέσου συνειρμών – ξεκινώντας από μια ιδέα

«ενεργοποιούνται» γειτονικές συνδεδεμένες έννοιες κ.ο.κ. [39]. Σε ένα KG, αυτό υλοποιείται διασχίζοντας ακμές του γράφου: ξεκινώντας π.χ. από έναν κόμβο που αντιστοιχεί σε μια λέξη-κλειδί του ερωτήματος, ακολουθούμε τις συνδέσεις σε κοντινούς κόμβους για μερικά βήματα, συλλέγοντας ένα υποσύνολο σχετικών γεγονότων. Η διαδικασία αυτή διευρύνει το πλαίσιο της ερώτησης και μπορεί να ανασύρει έμμεσα συσχετισμένες πληροφορίες που δεν αναφέρονται ρητά στην ερώτηση αλλά είναι σχετικές (π.χ. συνώνυμες έννοιες, σχετικοί πόροι). Σημαντικό πλεονέκτημα είναι ότι η ανάκτηση γίνεται σημασιολογικά καθοδηγούμενη: αντί να αναζητούμε απλώς όρους σε έγγραφα, "διαχέουμε" την ενεργοποίηση μέσω σχέσεων με ουσιαστικό νόημα, γεγονός που αυξάνει την ερμηνευσιμότητα και την πιθανότητα ανάκτησης βάσιμων στοιχείων . Το αποτέλεσμα του σταδίου 1 είναι ένα σύνολο από γεγονότα-τριάδες ή ένας υπο-γράφος από το KG, που θεωρείται ότι περιέχει την σχετική γνώση.

Επέκταση του ερωτήματος (query expansion & multi-source retrieval): Στη συνέχεια, το αρχικό ερώτημα του χρήστη μπορεί να εμπλουτιστεί με τις πληροφορίες που βρέθηκαν στο knowledge graph. Αυτό σημαίνει ότι δημιουργούμε μια εκτεταμένη εκδοχή του query, η οποία περιλαμβάνει λέξεις-κλειδιά ή ονόματα οντοτήτων από τα ανακτηθέντα KG facts. Ο εμπλουτισμός αποσκοπεί στο να βελτιώσει την επόμενη φάση ανάκτησης: εάν το σύστημα RAG διαθέτει επίσης μια συλλογή από μη δομημένα κείμενα (π.χ. άρθρα, σελίδες ιστού), ένα ενισχυμένο query που περιέχει σχετικούς όρους θα ανακτήσει ακριβέστερα αποσπάσματα από το κειμενικό σώμα. Με άλλα λόγια, τα facts του KG λειτουργούν ως «γέφυρα» ανάμεσα στη διατύπωση του χρήστη και στις διατυπώσεις των εγγράφων – παρέχοντας επιπλέον στοιχεία στο query μειώνεται η πιθανότητα να χαθεί μια καλή απάντηση λόγω απλής αναντιστοιχίας λέξεων. Ορισμένες προσεγγίσεις KG-RAG υλοποιούν ακόμα πιο σύνθετα σχήματα: π.χ. πολυ-γύρους ανάκτησης, όπου εναλλάξ γίνεται διάλογος μεταξύ LLM και γνώσης για να διευκρινιστεί το ερώτημα (chain-of-thought με ανάκτηση) . Ωστόσο, η πιο κοινή τεχνική είναι η εξής: από τη στιγμή που έχουμε έναν υπο-γράφο ή λίστα facts σχετικών με το query, είτε τα μετατρέπουμε σε κείμενο είτε εξάγουμε από αυτά κρίσιμες λέξεις, και τα προσθέτουμε ως μέρος του ερωτήματος προς έναν κειμενικό retriever (π.χ. μια μηχανή αναζήτησης ή ένα embedding-based retriever). Έτσι επιτυγχάνεται συνδυαστική ανάκτηση από πολλές πηγές: και από το KG (δομημένα facts) και από το corpus (μη δομημένα κείμενα).[38] Σημειώνεται ότι υπάρχουν και προσεγγίσεις όπου δεν υπάρχει εξωτερικό corpus, αλλά το ίδιο το KG θεωρείται η βάση γνώσης, σε αυτές, η επέκταση ερωτήματος μπορεί να μην είναι απαραίτητη· το σύστημα απλώς χρησιμοποιεί τον υπο-γράφο που βρέθηκε ως το μόνο πλαίσιο.

Γενετική απάντηση με γνώση (knowledge-aware generation): Στο τελικό στάδιο, το LLM παράγει την απάντηση ή το ζητούμενο κείμενο, έχοντας πλέον στη διάθεσή του το εμπλουτισμένο prompt. Αυτό το prompt περιλαμβάνει την αρχική ερώτηση του χρήστη καθώς και όλη την υποστηρικτική πληροφορία που συλλέχθηκε: τις σχετικές τριάδες γνώσης και/ή αποσπάσματα από το corpus. Το LLM, εκπαιδευμένο να χειρίζεται μεγάλο εύρος φυσικής γλώσσας, συνδυάζει τώρα το ενσωματωμένο γνώσι του με τα παρεχόμενα στοιχεία. Χάρη στο δομημένο χαρακτήρα των KG facts, το μοντέλο μπορεί να αντιληφθεί καλύτερα το πλαίσιο και να συνθέσει μια απάντηση με μεγαλύτερη ακρίβεια και συνάφεια . Για παράδειγμα, αντί να προσπαθεί να ανασύρει από τη «μνήμη» του μια πιθανή απάντηση – ρισκάροντας μια εικασία, το LLM βλέπει ρητά τις σχετικές οντότητες και τις συνδέσεις τους, κάτι που μειώνει τις πιθανότητες λανθασμένων ή κατασκευασμένων πληροφοριών. Επιπλέον, επειδή τα δεδομένα από το KG είναι συνοπτικά (π.χ. μια τριάδα είναι πολύ πιο πυκνή πληροφορίας από μια παράγραφο κειμένου), το τελικό input παραμένει σχετικά μικρό σε μέγεθος, αποφεύγοντας την υπέρβαση των ορίων παραθύρου συμφραζομένων του μοντέλου και εστιάζοντας στα πιο κρίσιμα στοιχεία.[28]

2.7.2 Συγκριτικά πλεονεκτήματα του KG-RAG

Η ενσωμάτωση των γνώσεων γραφήματος στο RAG έχει αποδειχθεί ευεργετική σε διάφορες διαστάσεις. Πρώτον, η δομημένη αναπαράσταση που προσφέρει το KG επιτρέπει στο σύστημα να αποθηκεύει και να ανακτά πληροφορίες με λιγότερο θόρυβο: αντί να παραθέτει ολόκληρα κείμενα, τροφοδοτεί το μοντέλο με συγκεκριμένα γεγονότα και περιφράσεις που ταιριάζουν ακριβώς στο ζητούμενο.[28] Αυτό βελτιώνει την ακρίβεια των τελικών απαντήσεων και μειώνει το ρίσκο εισαγωγής άσχετων λεπτομερειών. Δεύτερον, το γράφημα προσδίδει σημασιολογική επίγνωση στο στάδιο της ανάκτησης: το LLM μπορεί να αξιοποιήσει τις σχέσεις του KG για να κατανοήσει καλύτερα το ερώτημα και να απαιτήσει συγκεκριμένα κομμάτια γνώσης (π.χ. να ακολουθήσει μια σχέση μέρους-όλου ή αιτίας-αποτελέσματος), κάτι που οδηγεί σε βαθύτερο reasoning και δυνατότητα χειρισμού πολύ-βηματικών ερωτήσεων.[1] Τρίτον, τα knowledge graphs υποστηρίζουν εγγενώς τις δυναμικές ενημερώσεις: μπορούν να επεκταθούν ή να διορθωθούν εύκολα, επιτρέποντας σε ένα RAG σύστημα να παραμένει ενημερωμένο με τις τελευταίες γνώσεις χωρίς να απαιτείται πλήρης επανεκπαίδευση του LLM . Επιπλέον, η δομημένη φύση τους διευκολύνει την επεξηγησιμότητα του συστήματος: μπορούμε να παρουσιάσουμε τις τριάδες ή τα μονοπάτια γραφήματος που οδήγησαν σε μια απάντηση, δίνοντας στον χρήστη εμπιστοσύνη στην ορθότητά της. Τέλος, σε εξειδικευμένα πεδία (ιατρικά, νομικά κ.λπ.), όπου απαιτείται αυστηρή λογική συνοχή, έχει φανεί ότι τα

γραφήματα ως φορείς γνώσης βοηθούν στη διατήρηση της ακρίβειας και συνέπειας των απαντήσεων καλύτερα από ό,τι η χρήση απλών εγγράφων.[1]

3 Η Προτεινόμενη Προσέγγιση

3.1 Εισαγωγή

Όπως είδαμε και στο προηγούμενο κεφάλαιο τα LLMs, όπως οι GPT, LLaMA και Gemini, έχουν φέρει επανάσταση στην Επεξεργασία Φυσικής Γλώσσας, επιτυγχάνοντας εντυπωσιακά αποτελέσματα σε ποικιλία εργασιών: μετάφραση, περίληψη, παραγωγή κειμένου, ερώτηση–απάντηση. Ωστόσο, παρά την ισχύ τους, τα LLMs έχουν κάποιους περιορισμούς. Η τεχνική Retrieval-Augmented Generation (RAG) προτάθηκε για να ξεπεράσει αυτούς τους περιορισμούς. Η βασική της ιδέα είναι να εμπλουτίζει το prompt ενός LLM με πληροφορία που ανακτάται δυναμικά από μια εξωτερική βάση γνώσης (π.χ. αποθετήρια κειμένων, επιστημονικές βάσεις, knowledge graphs).

Με αυτόν τον τρόπο:

- Οι απαντήσεις **βασίζονται σε πραγματικά και ενημερωμένα δεδομένα** αντί σε παραδοχές του μοντέλου → μείωση των hallucinations [40], [41].
- Το μοντέλο αποκτά **πρόσβαση σε επικαιροποιημένη γνώση** χωρίς να χρειάζεται retraining [42].
- Δίνεται η δυνατότητα ενσωμάτωσης **domain-specific πληροφορίας** (π.χ. ιατρικά άρθρα, νομικά έγγραφα) [43], [44].
- Οι πηγές που ανακτώνται μπορούν να παρουσιαστούν στον χρήστη → βελτίωση **διαφάνειας και εμπιστοσύνης** [40].

Παράλληλα η τεχνική RAG έχει και επιπλέον πλεονεκτήματα:

- **Αποδοτικότητα**: Αποφεύγεται το δαπανηρό fine-tuning μεγάλων μοντέλων, αφού η νέα γνώση "κουμπώνει" εξωτερικά [45].
- **Επεκτασιμότητα**: Η μέθοδος προσαρμόζεται εύκολα σε διαφορετικά domains, αρκεί να υπάρχει διαθέσιμη σχετική βάση δεδομένων [40].
- **Modularity**: Ο retriever και ο generator μπορούν να βελτιώνονται ανεξάρτητα [25].

Είναι, λοιπόν, σημαντικό να μελετηθεί η απόδοση διαφορετικών υλοποιήσεων της τεχνικής RAG και να εξεταστεί κατά πόσο η ενσωμάτωση δομημένης γνώσης μέσω Knowledge Graphs μπορεί να βελτιώσει την ποιότητα και την αξιοπιστία των παραγόμενων απαντήσεων.

3.2 Αντικείμενο της Εργασίας

Είδαμε παραπάνω ότι η τεχνική RAG είναι μια αρκετά υποσχόμενη μέθοδος η οποία φαίνεται ικανή να λύσει και να βελτιώσει σημαντικά προβλήματα των LLMs. Παρά τα σημαντικά πλεονεκτήματα που προσφέρει όμως η τεχνική RAG, η πρακτική εφαρμογή της αντιμετωπίζει κρίσιμες προκλήσεις που επηρεάζουν άμεσα την ποιότητα των παραγόμενων απαντήσεων.

3.2.1 Προβληματισμοί

Καλούμαστε να αντιμετωπίσουμε το πρόβλημα της ποιότητας και της σχετικότητας της ανακτώμενης πληροφορίας μας που δίνεται από τον retriever ως context στον Generator και τελικά έχει κύριο λόγο στην ποιότητα της απάντησης που παίρνουμε. Η απλή προσθήκη κειμενικών δεδομένων στο context window ενός LLM δεν εγγυάται την επιτυχή ανάκτηση και χρήση της σωστής πληροφορίας. Συγκεκριμένα:

Πρόβλημα Σχετικότητας (Relevance): Τα παραδοσιακά συστήματα retrieval συχνά ανακτούν κείμενα με λεξιλογική ομοιότητα προς το ερώτημα, χωρίς όμως να εγγυώνται σημασιολογική σχετικότητα. Αυτό οδηγεί σε "θόρυβο" (noise) στο context, που μπορεί να παραπλανήσει το μοντέλο [46].

Πρόβλημα Πληρότητας (Coverage): Σε σύνθετα ερωτήματα που απαιτούν πολλαπλά κομμάτια πληροφορίας από διαφορετικά σημεία των δεδομένων, ένα naive retrieval σύστημα μπορεί να αποτύχει να συλλέξει όλη την απαραίτητη γνώση [47].

Πρόβλημα Οργάνωσης: Η μη δομημένη φύση των κειμενικών chunks περιορίζει την ικανότητα του συστήματος να κατανοήσει σχέσεις μεταξύ εννοιών, οντοτήτων και γεγονότων [48]. Ο retriever βλέπει απλά "κομμάτια κειμένου", όχι **δομημένη γνώση**.

Πρόβλημα Ιεραρχίας: Πληροφορίες σε διαφορετικά επίπεδα αφαίρεσης (λεπτομέρειες vs. γενικές έννοιες) αντιμετωπίζονται ομοιόμορφα, παρόλο που κάποια ερωτήματα απαιτούν υψηλού επιπέδου επισκόπηση ενώ άλλα χρειάζονται συγκεκριμένες λεπτομέρειες [49].

3.2.2 Η προσέγγισή μας

Για να απαντηθούν τα παραπάνω ερωτήματα, η παρούσα εργασία υλοποιεί και συγκρίνει **τρεις διαφορετικές αρχιτεκτονικές RAG**:

3.2.2.1 Naïve RAG

Η πρώτη προσέγγιση είναι η **Naive εκδοχή του RAG**, η οποία υλοποιείται ως γραμμή βάσης (baseline) για τη σύγκριση με τις πιο προχωρημένες εκδοχές.
Η μέθοδος αυτή ακολουθεί την κλασική αρχιτεκτονική RAG, συνδυάζοντας έναν μηχανισμό ανάκτησης πληροφορίας (retriever) με ένα γενετικό μοντέλο (LLM) που παράγει την τελική απάντηση.

Αρχικά, δημιουργείται η βάση γνώσης (knowledge base). Για να το πετύχουμε αυτό, το αρχικό corpus διασπάται σε μικρότερα τμήματα κειμένου (text chunks) συγκεκριμένου και σταθερού μεγέθους. Κάθε chunk μετατρέπεται σε διανυσματική αναπαράσταση (embedding vector) μέσω ενός κατάλληλου μοντέλου ενσωμάτωσης (embedding model). Τα παραγόμενα διανύσματα αποθηκεύονται σε μια βάση δεδομένων διανυσμάτων (vector store), ώστε να είναι δυνατή η αποδοτική αναζήτηση με βάση τη σημασιολογική ομοιότητα [1].

Για κάθε ερώτηση του χρήστη, ακολουθείται η εξής διαδικασία:

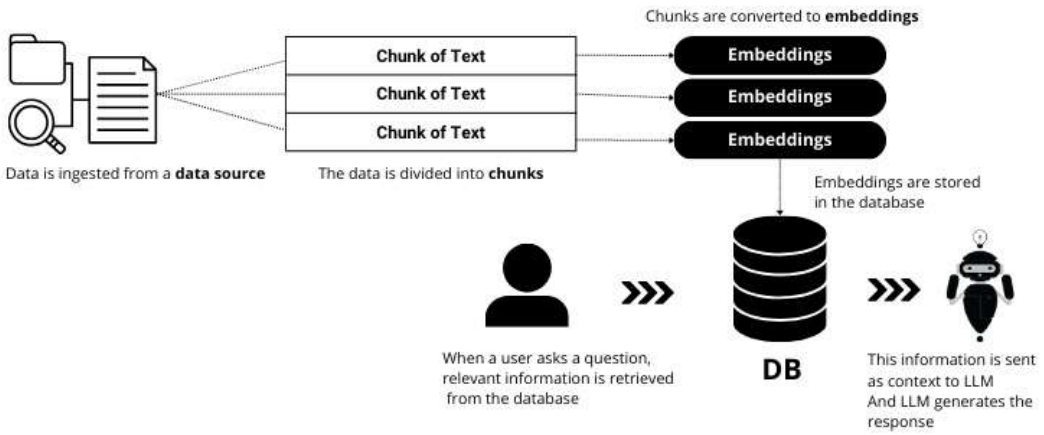
Μετατροπή ερωτήματος σε διάνυσμα: Η ερώτηση του χρήστη μετατρέπεται σε διανυσματική αναπαράσταση χρησιμοποιώντας το ίδιο embedding μοντέλο, ώστε να εξασφαλιστεί η σημασιολογική συμβατότητα με τα διανύσματα του corpus.

Ανάκτηση σχετικών αποσπασμάτων: Ο **retriever** συγκρίνει το διάνυσμα της ερώτησης με τα αποθηκευμένα διανύσματα και αναζητά τα **πιο σχετικά τμήματα κειμένου** (Top-k chunks), χρησιμοποιώντας μέτρα ομοιότητας, όπως το **cosine similarity**.

Εμπλουτισμός του prompt: Τα ανακτηθέντα τμήματα συνενώνονται με την αρχική ερώτηση, σχηματίζοντας ένα **εμπλουτισμένο prompt** που παρέχει στο LLM πρόσθετο πλαίσιο (context) γνώσης.

Παραγωγή απάντησης: Το LLM λαμβάνει το εμπλουτισμένο prompt και παράγει την τελική απάντηση, συνδυάζοντας τις πληροφορίες από τη βάση γνώσης με τις δικές του παραμετρικές γνώσεις.

Τα βήματα αυτά φαίνονται και στην *Εικόνα 6.*



Εικόνα 6 - Αναπαράσταση της Naive RAG pipeline

Η προσέγγιση αυτή, ενώ παρέχει ήδη σημαντικά οφέλη έναντι ενός LLM χωρίς πρόσβαση σε έγγραφα (καθώς μειώνει δραστικά τις παραισθήσεις μέσω τεκμηρίωσης των απαντήσεων [13]), ενδέχεται όπως αναφέρθηκε να αποτύχει σε περιπτώσεις που απαιτούν βαθύτερη σύνθεση πληροφορίας. Ενδεικτικά, αν η απάντηση ενός ερωτήματος προϋποθέτει στοιχεία διασκορπισμένα σε πολλά σημεία ενός μεγάλου κειμένου, το μοντέλο μπορεί να μην λάβει επαρκές πλαίσιο από μια περιορισμένη επιλογή αποσπασμάτων. Αυτούς τους περιορισμούς έρχονται να αντιμετωπίσουν οι επόμενες δύο προσεγγίσεις.

3.2.2.2 RAPTOR RAG

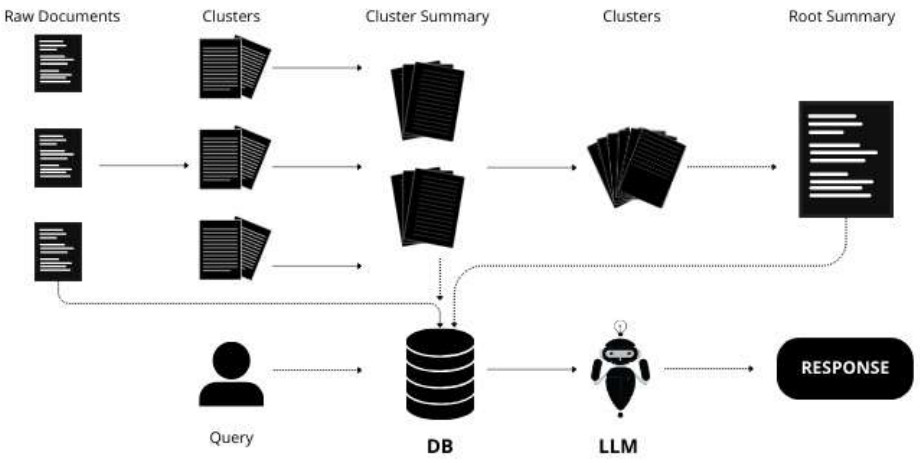
Η μέθοδος RAPTOR εισάγει μια δενδρική, ιεραρχική δομή στην οργάνωση της γνώσης που ανακτάται, με σκοπό να ξεπεραστεί το πρόβλημα της αποσπασματικής πληροφορίας. Βασική ιδέα του RAPTOR είναι η περίληψη και ομαδοποίηση πληροφοριών σε πολλαπλά επίπεδα ανάλυσης πριν από την ανάκτηση, δημιουργώντας έτσι ένα ευρετήριο γνώσης που αντικατοπτρίζει τις ανθρώπινες αντιλήψεις από τις λεπτομέρειες έως τις υψηλού επιπέδου έννοιες [35], [50].

Συγκεκριμένα, το RAPTOR πραγματοποιεί τα εξής βήματα κατά την φάση προ επεξεργασίας των δεδομένων:

Κατασκευή φύλλων (Leaf nodes): Αρχικά, όλα τα έγγραφα και οι πηγές κειμένου τεμαχίζονται στα γνωστά μικρά αποσπάσματα (chunks), αυτά αποτελούν τα φύλλα του δέντρου γνώσης. Κάθε φύλλο είναι ένα λεπτομερές κομμάτι πληροφορίας (π.χ. παράγραφος).

Ομαδοποίηση σε θεματικές ενότητες: Στη συνέχεια, εφαρμόζεται αλγόριθμος clustering στα φύλλα, ομαδοποιώντας τα σε θεματικά συναφή σύνολα. Για παράδειγμα, χρησιμοποιούνται τεχνικές όπως Γκαουσσιανά μίγματα ή ιεραρχικό clustering, πάνω στα ενσωματωμένα διανύσματα των αποσπασμάτων, ώστε τα φύλλα που πραγματεύονται παρόμοιες έννοιες να συσπειρωθούν μαζί [50]. Κάθε τέτοιο σύνολο φύλλων θα αποτελέσει τους απογόνους ενός γονικού κόμβου στο δενδρικό μοντέλο.

Δημιουργία περιλήψεων (ενδιάμεσοι κόμβοι): Για κάθε σύμπλεγμα/ομάδα φύλλων που δημιουργήθηκε, το σύστημα γεννά μια σύντομη περίληψη που αποτυπώνει τα κύρια σημεία ή το θέμα της ομάδας. Αυτή η περίληψη παράγεται με γενετικό μοντέλο (LLM) ή αλγόριθμο εξαγωγικής περίληψης, και αντιστοιχεί σε έναν γονικό κόμβο στο επόμενο επίπεδο του δέντρου [35]. Έτσι, κάθε γονικός κόμβος περιέχει συνοπτικά την πληροφορία των παιδιών του. Στη συνέχεια, οι γονικοί κόμβοι μπορεί με τη σειρά τους να ομαδοποιηθούν περαιτέρω και να παραχθούν περιλήψεις σε ανώτερο επίπεδο, επανειλημμένα και αναδρομικά, έως ότου δημιουργηθεί μια ιεραρχία πολλαπλών επιπέδων από γενικές έως ειδικές πληροφορίες (από τη ρίζα έως τα φύλλα). Το αποτέλεσμα είναι ένα δέντρο γνώσης όπου τα κατώτερα επίπεδα είναι εξειδικευμένα (πρωτογενή αποσπάσματα) και τα ανώτερα επίπεδα περιληπτικά/γενικά.



Εικόνα 7 - Αναπαράσταση RAPTOR pipeline

Αφού κατασκευαστεί αυτό το πολυεπίπεδο ευρετήριο, η διαδικασία ανάκτησης **σε χρόνο ερωτήματος** λειτουργεί διαφορετικά από τη βασική RAG. Όταν εισαχθεί ένα ερώτημα:

Το RAPTOR δύναται να **ανακτά πληροφορία από πολλαπλά επίπεδα** του δέντρου, ανάλογα με τη φύση του ερωτήματος. Για παράδειγμα, αν το ερώτημα είναι ευρύ ή συνοπτικό (π.χ.

"Ποια είναι τα κύρια θέματα ενός δοθέντος corpus;"), το σύστημα μπορεί να ανασύρει περιλήψεις από υψηλότερους κόμβους (που συνοψίζουν μεγάλα μέρη του κειμένου) αντί για αποσπάσματα φύλλων [35]. Αντίστροφα, για ένα συγκεκριμένο ερώτημα, θα ανακτηθούν συγκεκριμένα φύλλα ή κόμβοι χαμηλού επιπέδου που σχετίζονται άμεσα. Συχνά, μια συνδυαστική προσέγγιση χρησιμοποιείται: πρώτα εντοπίζονται σχετικά υψηλού επιπέδου τμήματα μέσω των περιλήψεων, και κατόπιν μπορούν να εμβαθύνουν στο δέντρο παίρνοντας και επιμέρους λεπτομερή αποσπάσματα από εκείνες τις ενότητες.

Στη συνέχεια, το LLM λαμβάνει ως context τόσο τις συνοπτικές πληροφορίες όσο και επιμέρους λεπτομέρειες, επιτρέποντάς του να **ενσωματώσει γνώση σε πολλαπλές κλίμακες** κατά την παραγωγή της απάντησης . Με τον τρόπο αυτό, επιτυγχάνεται μια ισορροπία μεταξύ **σφαιρικής κατανόησης** (global context) και **τοπικής λεπτομέρειας**: το μοντέλο έχει εικόνα των γενικών θεμάτων μέσω των περιλήψεων, ενώ ταυτόχρονα μπορεί να αντλήσει ακριβή στοιχεία από τα φύλλα.

Η RAPTOR προσέγγιση, *Εικόνα 7,* ουσιαστικά διατηρεί τη δομή της πληροφορίας του αρχικού γνώσης corpus μέσα από το δενδρικό ευρετήριο, αντί να την ισοπεδώνει σε μια απλή λίστα αποσπασμάτων. Έρευνες έχουν δείξει ότι η ανάκτηση από μια τέτοια πολυεπίπεδη δομή επιτρέπει στο σύστημα να απαντά πιο αποτελεσματικά σε ερωτήματα που απαιτούν ολιστική σύνοψη ή σύνθετη συλλογιστική, επιτυγχάνοντας σημαντικές βελτιώσεις έναντι της παραδοσιακής RAG σε εργασίες ερωταποκρίσεων πολλαπλών βημάτων [35]. Με απλά λόγια, το RAPTOR "κρατάει απλή" τη διαδικασία κατά το χρόνο ερώτησης (δεν προσθέτει πολλά στάδια επεξεργασίας τότε), έχοντας προετοιμάσει έξυπνα το ευρετήριο εκ των προτέρων, ώστε να παρέχει πλουσιότερο και πιο οργανωμένο πλαίσιο στο LLM [50].

3.2.2.3 KG-RAG

Η τρίτη προσέγγιση επεκτείνει το κλασικό RAG ενσωματώνοντας ρητά δομημένη γνώση από υπάρχοντες Γράφους Γνώσης (Knowledge Graphs), πέρα από το απλό κείμενο. Αναφερόμαστε σε αυτήν ως KG-RAG, δηλαδή RAG με Knowledge Graph. Η συλλογιστική πίσω από την εν λόγω μέθοδο είναι η εξής: οι γράφοι γνώσης περιέχουν κόμβους (οντότητες) και ακμές (σχέσεις) που αντιστοιχούν σε πραγματικά γεγονότα ή σχέσεις μεταξύ εννοιών. Αν το ερώτημα του χρήστη περιέχει συγκεκριμένες οντότητες ή αφορά κάποια θεματολογία, τότε ένας γνώσης γράφος μπορεί να βοηθήσει να βρεθούν συνδεδεμένες πληροφορίες που δεν αναφέρονται ρητά στο ίδιο το ερώτημα, αλλά είναι σχετικές μέσω σημασιολογικών συνδέσεων [39]. Με άλλα λόγια, η χρήση του KG προσδίδει ένα είδος "νοητικής διάχυσης" (spreading activation) όπως

κάνει ο ανθρώπινος εγκέφαλος: από μια ιδέα ενεργοποιούνται συνδεδεμένες έννοιες, διευρύνοντας την αναζήτηση γνώσης πέρα από τα επιφανειακά ταιριάσματα λέξεων.

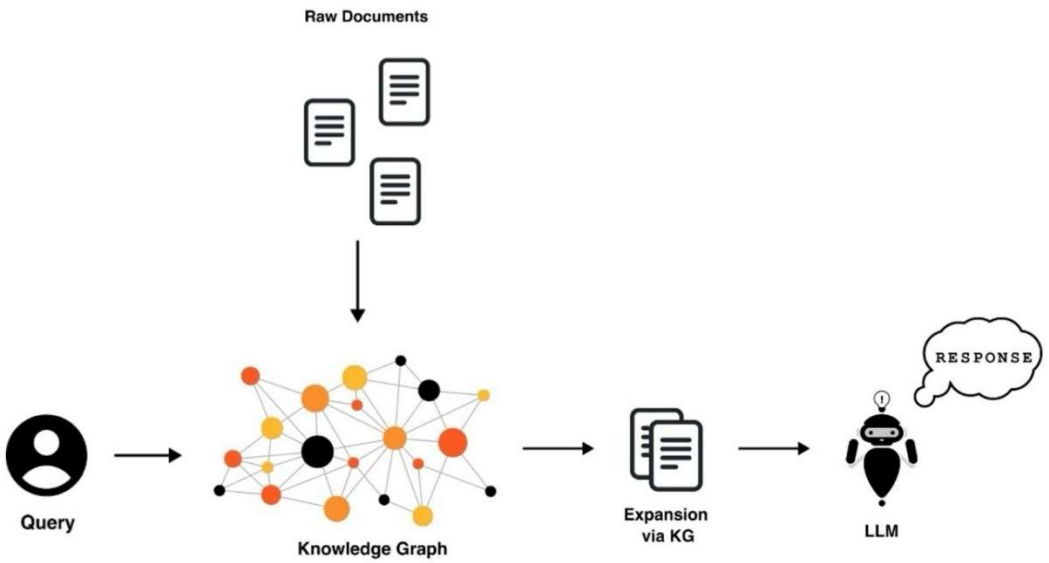
Η συνολική ροή της μεθόδου KG-RAG περιλαμβάνει τα εξής στάδια [38]:

Προεπεξεργασία εγγράφων: Τα έγγραφα διασπώνται σε λογικές ενότητες (chunks») και κάθε ενότητα συσχετίζεται με τον γράφο γνώσης μέσω αναγνώρισης οντοτήτων και σχέσεων. Με αυτόν τον τρόπο καταγράφονται εκ των προτέρων οι σχέσεις γεγονότων μεταξύ των τμημάτων.

Σημασιολογική αρχική ανάκτηση: Με βάση ένα δοθέν ερώτημα, εκτελείται σημασιολογική αναζήτηση για τον εντοπισμό αρχικών σχετικών τμημάτων («seed chunks»). Αυτά τα αρχικά τμήματα χρησιμεύουν ως αφετηρία για περαιτέρω διερεύνηση μέσω του γράφου.

Διεύρυνση μέσω γράφου γνώσης: Από τον γράφο επιλέγεται υπο-σύνολο σχετικό με τα αρχικά τμήματα και μέσω πλοήγησης στον γράφο αναγνωρίζονται επιπλέον τμήματα που συνδέονται με κοινές οντότητες ή σχέσεις. Η επέκταση αυτή αυξάνει την ποικιλία και την κάλυψη των πληροφοριών που ανακτώνται, δημιουργώντας ένα πιο ολοκληρωμένο δίκτυο γνώσης.

Οργάνωση περιεχομένου: Τέλος, τα ανακτηθέντα τμήματα φιλτράρονται ώστε να διατηρούνται μόνο τα πιο σημαντικά στοιχεία, και αναδιατάσσονται σε εσωτερικά συνεκτικές παραγράφους με βάση τη δομή του γράφου γνώσης. Η γνώση οργανώνεται δηλαδή με τον γράφο ως «σκελετό», εξασφαλίζοντας ομαλή ροή και συνάφεια στη συνολική απάντηση.



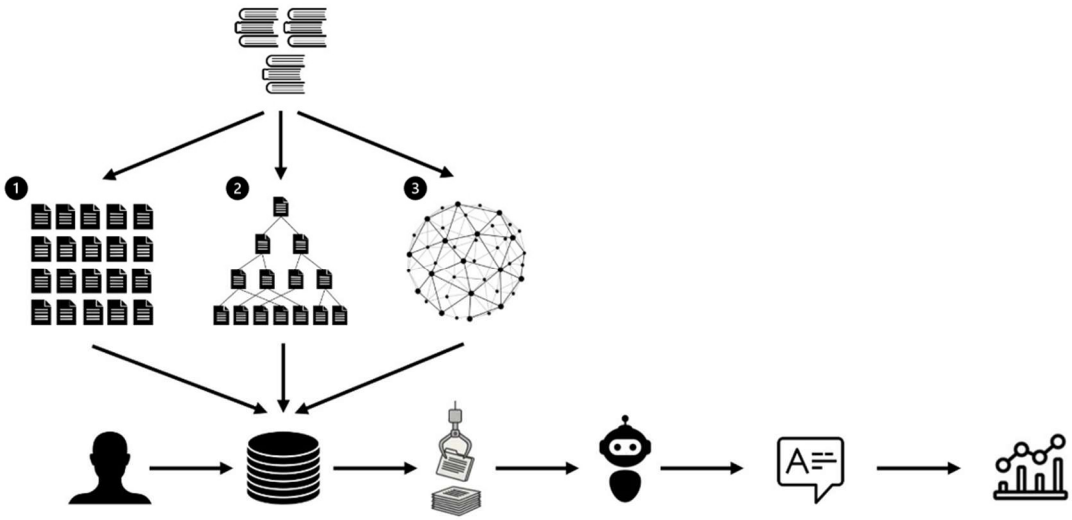
Εικόνα 8 - Αναπαράσταση KG-RAG pipeline [38]

Με την KG-RAG προσέγγιση, *Εικόνα 8*, αναμένουμε ότι θα βελτιωθεί η ακρίβεια των απαντήσεων (καθώς πολλά λάθη των LLMs οφείλονται σε έλλειψη συγκεκριμένης γνώσης, την οποία ο KG παρέχει) και θα μειωθούν περαιτέρω οι παραισθήσεις. Για παράδειγμα, αν ρωτηθεί το σύστημα για μια χρονολογία ή μια σχέση μεταξύ δύο οντοτήτων, ο γράφος μπορεί να δώσει απευθείας την απάντηση ή να κατευθύνει το μοντέλο στο σωστό τεκμήριο. Επίσης, η μέθοδος αυτή συνδυάζει δομημένη και μη δομημένη γνώση (multi-source retrieval) [38], αξιοποιώντας το καλύτερο από δύο κόσμους: το μεγάλο κείμενο corpus για λεπτομερείς περιγραφές και το γράφο για επαλήθευση κρίσιμων γεγονότων.

Σημειώνεται ότι η ενσωμάτωση ενός KG εισάγει κάποια επιπλέον πολυπλοκότητα στο pipeline, συγκριτικά με τις άλλες μεθόδους, καθώς απαιτείται σύνδεση με εξωτερική βάση γνώσης και λογική επέκτασης των ερωτημάτων. Ωστόσο, όπως έχει δειχθεί σε σχετικές μελέτες, η προσθήκη τέτοιων μηχανισμών μπορεί να αποδώσει σημαντικά οφέλη. Ενδεικτικά, μια πρόσφατη υλοποίηση RAG με KG πέτυχε μείωση του ποσοστού παραισθήσεων κατά 73% σε σχέση με ένα καθαρά κειμενικό μοντέλο GPT-4 και παράλληλα αύξησε την πληρότητα των απαντήσεων, συνδυάζοντας τεκμήρια από γράφο και έγγραφα [51]. Αν και τα συγκεκριμένα νούμερα αφορούν διαφορετικό πλαίσιο (πολυ-πρακτορικό σύστημα RAG-KG-IL), καταδεικνύουν τη δυναμική του εμπλουτισμού με δομημένη γνώση στην αντιμετώπιση των αδυναμιών των LLMs.

Μέσω συστηματικής αξιολόγησης των τριών αυτών προσεγγίσεων, στοχεύουμε να κατανοήσουμε πότε και γιατί η δομημένη γνώση υπερτερεί της μη δομημένης, και πώς μπορεί να βελτιωθεί η απόδοση συστημάτων ερώτησης-απάντησης και να επιτευχθεί μεγαλύτερη ακρίβεια, διαφάνεια και χρηστικότητα.

Έτσι η συνολική διαδικασία που θα πραγματοποιήσουμε συνοψίζεται στην *Εικόνα 9.* Ξεκινώντας από ένα κοινό dataset θα ακολουθήσουμε 3 διαφορετικούς τρόπους να δημιουργήσουμε τη βάση γνώσης μας. Στη συνέχεια θα θέτουμε ερωτήματα και θα αξιολογήσουμε τις απαντήσεις που θα παράξει κάθε διαφορετική μέθοδος RAG που εξετάζουμε, ώστε να βγάλουμε συμπεράσματα για την απόδοσή τους τόσο της κάθε μίας ξεχωριστά όσο και συγκριτικά μεταξύ τους.



Εικόνα 9 - Συνολική προσέγγιση ΔΕ

3.2.3 Ερευνητικά ερωτήματα

Με τη διαδικασία που θα ακολουθήσουμε στην παρούσα διπλωματική εργασία καλούμαστε να απαντήσουμε τα εξής ερευνητικά ερωτήματα που έχουν προκύψει από τις προκλήσεις και τους προβληματισμούς γύρω από τα RAG:

1. Πώς επηρεάζει η μέθοδος οργάνωσης και ανάκτησης πληροφορίας την ποιότητα των απαντήσεων σε ένα σύστημα RAG;
2. *Σε ποιο βαθμό βελτιώνει η ιεραρχική προσέγγιση RAPTOR την ικανότητα του συστήματος να ανακτά και να αξιοποιεί πληροφορίες από μεγάλα ή πολύπλοκα έγγραφα, σε σύγκριση με μια απλή RAG;* Εδώ διερευνάται αν το δενδρικό μοντέλο συνοπτικών αναπαραστάσεων της γνώσης μπορεί να οδηγήσει σε πληρέστερες και ακριβέστερες απαντήσεις έναντι της απλής ανάκτησης επίπεδων αποσπασμάτων.
3. *Πώς συμβάλλει η ενσωμάτωση ενός KG στην ποιότητα της ανάκτησης και της γενετικής διαδικασίας;* Εξετάζεται αν η παροχή δομημένων γεγονότων από ένα KG, μέσω επέκτασης του ερωτήματος ή παράλληλης ανάκτησης, βελτιώνει την πραγματολογική ορθότητα (μείωση παραισθήσεων) και την κάλυψη των απαντήσεων σε σχέση με την RAG χωρίς γραφήματα.
4. *Ποιες είναι οι διαφοροποιήσεις στον σχεδιασμό του pipeline και στην απόδοση μεταξύ των τριών προσεγγίσεων;* Αυτό το ερώτημα στοχεύει να αποσαφηνίσει ποια στάδια ή παράμετροι του συστήματος επηρεάζονται από κάθε προσέγγιση (π.χ. ανάγκη για προ επεξεργασία/περίληψη, επιβάρυνση χρόνου ανάκτησης) και πώς αυτές οι διαφορές

μεταφράζονται σε μετρικές επίδοσης. Ουσιαστικά, επιδιώκεται μια σύγκριση πολυδιάστατη: όχι μόνο ως προς την ακρίβεια των απαντήσεων, αλλά και ως προς την αποδοτικότητα και την πολυπλοκότητα υλοποίησης.

Απαντώντας στα παραπάνω ερωτήματα, η εργασία θα τεκμηριώσει ποια από τις τεχνικές RAG υπερέχει υπό ποιες συνθήκες και θα συνεισφέρει γνώση για τον σχεδιασμό βελτιωμένων RAG συστημάτων στο μέλλον.

4 Υλοποίηση

4.1 Εισαγωγή

Στην παρούσα εργασία, όλες οι υλοποιήσεις των συστημάτων Retrieval-Augmented Generation (RAG) βασίστηκαν σε LLMs τα οποία εκτελέστηκαν σε self-hosted περιβάλλον. Η επιλογή αυτή έγινε για δύο λόγους:

1. Ανεξαρτησία και έλεγχος: Η χρήση self-hosted λύσεων επιτρέπει πλήρη έλεγχο πάνω στην υποδομή, στα δεδομένα και στα πειράματα, χωρίς εξάρτηση από cloud APIs ή εμπορικές υπηρεσίες.
2. Αναπαραγωγιμότητα και κόστος: Η τοπική εκτέλεση επιτρέπει την αναπαραγωγή των πειραμάτων χωρίς πρόσθετο κόστος κλήσεων API και χωρίς περιορισμούς χρήσης.

Για το σκοπό αυτό χρησιμοποιήθηκε το εργαλείο **Ollama**, ένα framework που επιτρέπει την εύκολη εκτέλεση LLMs τοπικά. Μέσω του Ollama παρέχεται πρόσβαση σε διάφορα μοντέλα, όπως το **Mistral-small (24B)** για παραγωγή απαντήσεων, και το **nomic-embed-text** για δημιουργία διανυσματικών αναπαραστάσεων (embeddings).

Εικόνα 10 - Ollamas

Η ενσωμάτωση με την πλατφόρμα **LangChain** έδωσε τη δυνατότητα:

- να συνδεθούν τα μοντέλα Ollama με τα pipelines ανάκτησης και δημιουργίας,
- να υλοποιηθούν retrievers πάνω σε FAISS vector stores,
- και να οργανωθεί η συνολική διαδικασία RAG με modular τρόπο.

Συνεπώς, όλες οι επόμενες υλοποιήσεις (Naive RAG, RAPTOR RAG, RAG με Knowledge Graph) αξιοποιούν την ίδια υποδομή Ollama για λόγους συνέπειας και συγκρισιμότητας.

4.2 Υλοποίηση Naive RAG (Baseline)

Η πρώτη υλοποίηση που αναπτύχθηκε στο πλαίσιο της εργασίας είναι το **Naive RAG**, το οποίο λειτουργεί ως γραμμή βάσης για τη σύγκριση με τις πιο σύνθετες προσεγγίσεις. Η αρχιτεκτονική του ακολουθεί το κλασικό σχήμα **"retrieve-then-read"**, στο οποίο ένα ερώτημα χρήστη μετατρέπεται σε διανυσματική αναπαράσταση, αναζητούνται τα πιο σχετικά τμήματα κειμένου και στη συνέχεια αυτά παρέχονται στο γλωσσικό μοντέλο για την παραγωγή απάντησης.

4.2.1 Φόρτωση και προετοιμασία δεδομένων

Αρχικά, τα κείμενα φορτώνονται από αρχείο JSON και συνενώνονται σε μεγαλύτερα αποσπάσματα. Στη συνέχεια εφαρμόζεται **τεμαχισμός κειμένων (chunking)** με τη μέθοδο *RecursiveCharacterTextSplitter*, ώστε να προκύψουν τμήματα μεγέθους περίπου 1000 χαρακτήρων με επικαλύψεις. Ο σκοπός είναι να δημιουργηθούν μικρότερες μονάδες κειμένου, οι οποίες μπορούν να ανακτηθούν πιο αποδοτικά και να καλύψουν διαφορετικά σημεία του corpus.

4.2.2 Δημιουργία διανυσματικού χώρου (vector store)

Για την αναπαράσταση των κειμένων χρησιμοποιούνται διανυσματικές ενσωματώσεις (embeddings). Κάθε chunk μετατρέπεται σε διάνυσμα μέσω του μοντέλου OllamaEmbeddings. Οι αναπαραστάσεις αποθηκεύονται σε ευρετήριο FAISS, το οποίο επιτρέπει γρήγορη αναζήτηση με βάση την ομοιότητα (cosine similarity / L2 distance).

Η δομή αυτή (vector store) λειτουργεί ως «βάση γνώσης» για τον retriever, και επιτρέπει την άμεση ανάκτηση των πιο σχετικών τμημάτων με βάση το εκάστοτε ερώτημα.

4.2.3 Μηχανισμός ανάκτησης (Retriever)

Ο retriever υλοποιείται πάνω από το FAISS index και ρυθμίζεται να επιστρέφει τα **πέντε πιο σχετικά chunks (k=5)**. Επιπλέον, εφαρμόζεται η τεχνική **Maximal Marginal Relevance (MMR)**, η οποία επιτυγχάνει ισορροπία ανάμεσα στη συνάφεια και την ποικιλία των ανακτήσεων. Έτσι, αποφεύγεται η επανάληψη σχεδόν ίδιων αποσπασμάτων, ενώ ταυτόχρονα καλύπεται μεγαλύτερο εύρος περιεχομένου.

4.2.4 Δημιουργία prompt και σύνθεση απάντησης

Τα chunks που επιστρέφει ο retriever μορφοποιούνται σε ένα ενιαίο κείμενο και ενσωματώνονται σε **πρότυπο prompt (template)**. Το prompt περιλαμβάνει:

- την πληροφορία συμφραζομένων (context),
- την ερώτηση του χρήστη,
- και οδηγίες προς το μοντέλο να απαντήσει με σύντομη και τεκμηριωμένη απάντηση.

Ως **γλωσσικό μοντέλο (generator)** χρησιμοποιείται το ChatOllama με το μοντέλο mistral-small:24b. Η απάντηση που παράγεται εξάγεται ως καθαρό κείμενο μέσω parser.

4.2.5 Παραγωγή και αποθήκευση αποτελεσμάτων

Το σύστημα απαντά σε κάθε ερώτηση επιστρέφοντας:

- την τελική απάντηση,
- τα αποσπάσματα που χρησιμοποιήθηκαν ως συμφραζόμενα,
- και υποθετικά “supporting facts”, τα οποία αντιστοιχούν στις πηγές των απαντήσεων.

Τα αποτελέσματα αποθηκεύονται σε αρχείο JSON, ώστε να μπορούν να χρησιμοποιηθούν αργότερα στην αξιολόγηση (Κεφάλαιο 5).

4.2.6 Συνολική εκτίμηση

Η υλοποίηση του Naive RAG παρέχει ένα λειτουργικό baseline, το οποίο είναι απλό αλλά ικανό να δώσει τεκμηριωμένες απαντήσεις βασισμένες σε ανακτημένα αποσπάσματα. Ωστόσο, παρουσιάζει περιορισμούς σε ερωτήσεις που απαιτούν **πολυβηματικό συλλογισμό** ή καλύτερη οργάνωση πληροφορίας. Οι περιορισμοί αυτοί αποτέλεσαν το έναυσμα για την ανάπτυξη των πιο σύνθετων προσεγγίσεων (RAPTOR RAG και KG-RAG), που θα παρουσιαστούν παρακάτω.

4.3 Υλοποίηση RAPTOR RAG

Η δεύτερη υλοποίηση που αναπτύχθηκε είναι το RAPTOR RAG (Retrieval with Abstracted Passage Tree Organizing and Reasoning), το οποίο επιχειρεί να ξεπεράσει τους περιορισμούς της baseline εκδοχής εισάγοντας μια ιεραρχική δομή αναπαράστασης και ανάκτησης γνώσης.

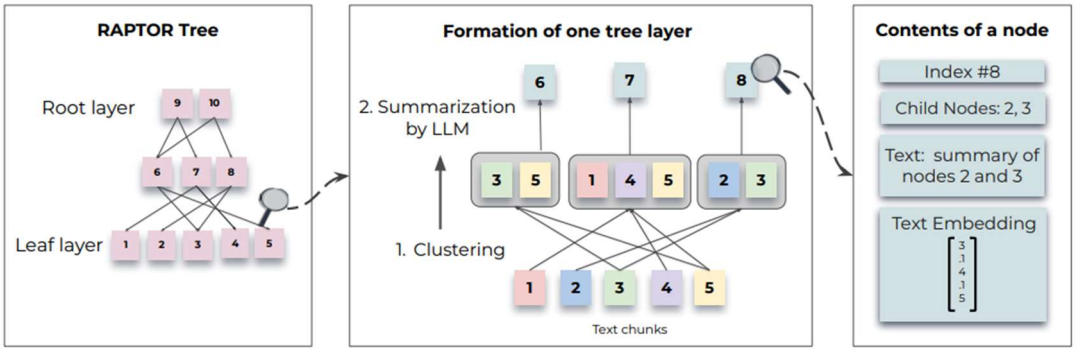
4.3.1 Ιδέα της μεθόδου

Ενώ στο Naive RAG κάθε chunk αντιμετωπίζεται ισότιμα, το RAPTOR δημιουργεί μια **δενδρική δομή (tree)** από τα κείμενα:

- Στα κατώτερα επίπεδα βρίσκονται τα αρχικά αποσπάσματα κειμένου.
- Σε ανώτερα επίπεδα, ομάδες κειμένων **συνοψίζονται** με χρήση του LLM ώστε να παραχθούν κόμβοι-συνοψίσεις.
- Το αποτέλεσμα είναι μια πολυεπίπεδη οργάνωση, που επιτρέπει **πρώτα ανάκτηση σε υψηλό επίπεδο** και στη συνέχεια **διάνοιξη (drill-down)** στα πιο λεπτομερή τμήματα.

Έτσι, το RAPTOR μειώνει τον θόρυβο, οργανώνει καλύτερα τα δεδομένα και διευκολύνει την ανάκτηση πληροφοριών που σχετίζονται με πιο σύνθετες ερωτήσεις.

Στην *Εικόνα 11* παρουσιάζεται και σχηματικά η διαδικασία κατασκευής της δενδρικής δομής της RAPTOR.



Εικόνα 11 - Μέθοδος κατασκευής RAPTOR tree

4.3.2 Δημιουργία ιεραρχίας (RAPTOR Tree)

Η διαδικασία ξεκινά με τα αρχικά κείμενα και περνάει από διαδοχικά επίπεδα:

1. **Υπολογισμός embeddings** για όλα τα αποσπάσματα με χρήση OllamaEmbeddings.
2. **Ομαδοποίηση (clustering):** Τα embeddings ομαδοποιούνται με **Gaussian Mixture Models (GMM)**, σχηματίζοντας clusters.
3. **Σύνοψη κάθε cluster:** Για κάθε ομάδα κειμένων, το LLM παράγει μία συνοπτική αναπαράσταση (summary). Αυτές οι συνοψίσεις αποτελούν τους κόμβους του επόμενου επιπέδου.
4. **Επανάληψη:** Η ίδια διαδικασία επαναλαμβάνεται, δημιουργώντας νέα επίπεδα συνοψίσεων, μέχρι να μείνει ένας ή λίγοι κόμβοι στην κορυφή του δέντρου.

Η πληροφορία για κάθε κόμβο συνοδεύεται από metadata (επίπεδο, προέλευση, παιδικοί κόμβοι), ώστε να διατηρείται η σύνδεση με τα αρχικά αποσπάσματα.

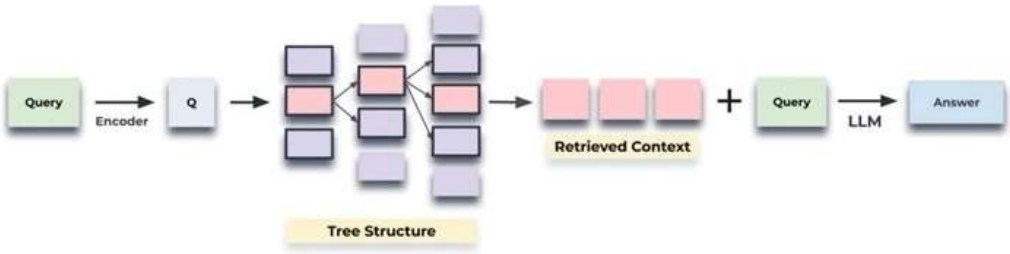
4.3.3 Δημιουργία διανυσματικού χώρου και retriever

Όλοι οι κόμβοι του δέντρου (τόσο τα πρωτογενή αποσπάσματα όσο και οι συνοψίσεις) εισάγονται σε έναν **FAISS vector store**. Έτσι, η αναζήτηση μπορεί να πραγματοποιηθεί τόσο στα επίπεδα των αρχικών chunks όσο και στις συνοπτικές αναπαραστάσεις.

Ο μηχανισμός ανάκτησης δεν λειτουργεί απλά με similarity search. Εφαρμόζεται **ιεραρχική αναζήτηση (hierarchical retrieval)**:

- Αρχικά γίνεται αναζήτηση στα υψηλότερα επίπεδα (summaries).
- Έπειτα, αν βρεθούν σχετικά summaries, γίνεται διάνοιξη στους child κόμβους τους.
- Η διαδικασία συνεχίζεται μέχρι να φτάσουμε στα χαμηλότερα επίπεδα, ώστε να συγκεντρωθούν οι πιο συναφείς λεπτομέρειες.

Η ιεραρχική αναζήτηση παρουσιάζεται στην *Εικόνα 12.*



Εικόνα 12 - Διαδικασία ανάκτησης από RAPTOR tree

Για την περαιτέρω βελτίωση, χρησιμοποιείται ένας **Contextual Compression Retriever**. Ο retriever αυτός εφαρμόζει ένα επιπλέον φίλτρο με LLM (LLMChainExtractor), το οποίο επιλέγει τα πιο σχετικά αποσπάσματα από το ήδη ανακτημένο σύνολο, μειώνοντας τον όγκο άσχετων πληροφοριών.

4.3.4 Δημιουργία prompt και απάντηση

Τα αποσπάσματα που ανακτώνται μέσω της ιεραρχικής αναζήτησης συγκεντρώνονται και ενσωματώνονται σε prompt. Όπως και στο Naive RAG, το prompt περιέχει:

- τις πληροφορίες συμφραζομένων,
- την ερώτηση του χρήστη,

- και οδηγίες για παραγωγή σύντομης factoid απάντησης.

Το γλωσσικό μοντέλο που χρησιμοποιείται είναι ξανά το ChatOllama με το μοντέλο mistral-small:24b. Η απάντηση παράγεται με βάση το εμπλουτισμένο context και αποθηκεύεται μαζί με τα μεταδεδομένα (επίπεδο κόμβου, προέλευση κ.λπ.).

4.3.5 Παραγωγή και αποθήκευση αποτελεσμάτων

Για κάθε ερώτηση, το RAPTOR RAG επιστρέφει:

- την τελική απάντηση,
- τα έγγραφα/συνοψίσεις που ανακτήθηκαν,
- πληροφορίες για το επίπεδο του δέντρου από όπου προήλθε κάθε τεκμήριο.

Τα αποτελέσματα αποθηκεύονται σε αρχείο JSON, ώστε να είναι διαθέσιμα για αξιολόγηση (Κεφάλαιο 5).

4.3.6 Συνολική εκτίμηση

Η υλοποίηση του RAPTOR RAG εισάγει ένα σημαντικό βήμα πέρα από το Naive RAG. Η ιεραρχική οργάνωση και η συμπίεση περιεχομένου καθιστούν δυνατή την καλύτερη κάλυψη πολύπλοκων ερωτήσεων, μειώνοντας ταυτόχρονα τον πλεονασμό και τον θόρυβο. Παράλληλα, η δυνατότητα οπτικοποίησης των clusters και των επιπέδων παρέχει μια πιο ερμηνεύσιμη εικόνα για το πώς οργανώνεται η γνώση. Ωστόσο, η πολυπλοκότητα της διαδικασίας και η πρόσθετη υπολογιστική επιβάρυνση αποτελούν βασικούς παράγοντες που θα εξεταστούν στην αξιολόγηση.

4.4 Υλοποίηση RAG με Knowledge Graph (KG-RAG)

Η τρίτη υλοποίηση της εργασίας είναι το **KG-RAG**, το οποίο συνδυάζει τις βασικές αρχές του RAG με τη δύναμη των **Knowledge Graphs (KGs)**. Σε αντίθεση με τις προηγούμενες μεθόδους που στηρίζονταν αποκλειστικά σε μη δομημένα αποσπάσματα κειμένου, εδώ αξιοποιείται επιπλέον μια σημασιολογικά δομημένη αναπαράσταση της γνώσης υπό μορφή τριπλετών. Στόχος είναι η βελτίωση της ακρίβειας, της ερμηνευσιμότητας και της ικανότητας εκτέλεσης πολυβηματικού συλλογισμού.

4.4.1 Ιδέα της μεθόδου

Η βασική αρχή του KG-RAG είναι ότι κάθε ερώτηση δεν απαντάται μόνο με βάση τα σχετικά αποσπάσματα κειμένου, αλλά και με τη **συμπληρωματική πληροφορία που προκύπτει από**

τον Knowledge Graph. Ο γράφος παρέχει οντότητες, σχέσεις και paths, τα οποία λειτουργούν ως δομημένο πλαίσιο γνώσης. Με αυτόν τον τρόπο:

- οι απαντήσεις στηρίζονται σε **επαληθεύσιμα facts**,
- το σύστημα μπορεί να αξιοποιεί **multi-hop reasoning** (π.χ. πώς συνδέονται δύο οντότητες μέσω ενδιάμεσων κόμβων),
- και η τελική έξοδος γίνεται πιο **εξηγήσιμη**, αφού το provenance κάθε πληροφορίας συνδέεται με κόμβους του γράφου.

4.4.2 Φόρτωση και προετοιμασία δεδομένων

Η διαδικασία ξεκινά με την ανάγνωση του corpus και των αντίστοιχων Knowledge Graphs.

- Για κάθε οντότητα των context εγγράφων φορτώνεται το αντίστοιχο υπογράφημα (sub-KG) από αποθηκευμένα JSON αρχεία.
- Τα context κείμενα τεμαχίζονται σε μικρότερα αποσπάσματα (nodes), τα οποία συνδέονται με τις οντότητες του γράφου.
- Παράλληλα, διατηρείται ένα ευρετήριο που συνδέει κάθε απόσπασμα κειμένου με το σχετικό του KG, ώστε να είναι δυνατή η συνδυαστική αναζήτηση.

4.4.3 Ενσωμάτωση LLM και Embeddings

Το pipeline βασίζεται στο **Ollama** για την εκτέλεση τόσο του LLM (mistral-small:24b) όσο και των embeddings (nomic-embed-text).

- Τα αποσπάσματα κειμένων μετατρέπονται σε διανύσματα και οργανώνονται σε **VectorStoreIndex**.
- Ο μηχανισμός αναζήτησης υλοποιείται με έναν **VectorIndexRetriever**, ο οποίος επιστρέφει τα πιο σχετικά κείμενα ως αρχικά candidates.

4.4.4 Μηχανισμοί Post-Processing με βάση τον Knowledge Graph

Η βασική καινοτομία του KG-RAG έγκειται στη χρήση **post-processors** που εφαρμόζονται στα αποτελέσματα του retriever. Ο στόχος τους είναι να ενσωματώσουν σημασιολογική πληροφορία από τον Knowledge Graph, να εμπλουτίσουν την ανάκτηση και να φιλτράρουν αποτελεσματικά τα άσχετα τεκμήρια. Συγκεκριμένα υλοποιούνται οι εξής τρεις μηχανισμοί:

- **NaivePostprocessor:**
  Εκτελεί μια απλή αναδιάταξη και ομαδοποίηση των ανακτηθέντων αποσπασμάτων,

έτσι ώστε οι πληροφορίες να εμφανίζονται με συνεκτικό τρόπο ανά οντότητα. Λειτουργεί ως baseline βήμα καθαρισμού.

- **KGRetrievePostProcessor:**
  Επεκτείνει τα ανακτημένα αποσπάσματα με βάση τις τριπλέτες του Knowledge Graph. Για κάθε οντότητα που εντοπίζεται, αναζητά συνδεδεμένες οντότητες και σχέσεις μέσα από το γράφημα, ώστε να εμπλουτίσει το σύνολο των τεκμηρίων. Με αυτόν τον τρόπο, το σύστημα μπορεί να ανακαλύψει πληροφορία που δεν ήταν άμεσα διαθέσιμη από τα κείμενα, αλλά προκύπτει από τις συνδέσεις στον γράφο.
- **GraphFilterPostProcessor:**
  Δημιουργεί έναν γράφο με κόμβους τις οντότητες και ακμές τις σχέσεις που αντλούνται από τα αποσπάσματα. Στη συνέχεια εντοπίζει τα **συνδεδεμένα components** που σχετίζονται περισσότερο με την ερώτηση (μέσω overlap οντοτήτων/σχέσεων με το query) και επιλέγει τα πιο υποσχόμενα paths. Ο μηχανισμός ενισχύεται από reranker (**BAAI/bge-reranker-large**), ο οποίος κατατάσσει τις πιθανές απαντήσεις βάσει σημασιολογικής εγγύτητας με την ερώτηση. Έτσι, απορρίπτονται τα άσχετα ή θορυβώδη αποσπάσματα.

Η συνδυαστική χρήση των παραπάνω μηχανισμών οδηγεί σε τρία κρίσιμα οφέλη:

1. **Καλύτερη κάλυψη (coverage):** μέσω της επέκτασης των αποσπασμάτων με γειτονικές οντότητες από τον γράφο.
2. **Υψηλότερη ακρίβεια:** μέσω του φιλτραρίσματος και reranking με βάση τη δομή του KG και τη συνάφεια με την ερώτηση.
3. **Πιο εξηγήσιμα αποτελέσματα:** καθώς κάθε επιλογή τεκμηρίου μπορεί να ανιχνευθεί πίσω σε συγκεκριμένες τριπλέτες.

Έτσι, ο Knowledge Graph δεν λειτουργεί μόνο ως επιπλέον πηγή γνώσης, αλλά και ως εργαλείο για τον **έλεγχο ποιότητας** της ανάκτησης.

4.4.5 Μηχανισμός Σύνθεσης Απαντήσεων (Response Synthesizer)

Το τελευταίο στάδιο του KG-RAG pipeline αφορά τη **σύνθεση της τελικής απάντησης** από τα αποσπάσματα που έχουν ανακτηθεί και φιλτραριστεί μέσω των post-processors. Η υλοποίηση βασίζεται στην αρχή του **Refine-based synthesis**, όπου η απάντηση παράγεται σταδιακά και βελτιώνεται επαναληπτικά καθώς εισάγονται νέα κομμάτια πληροφορίας.

Η βασική ιδέα είναι ότι, αντί το μοντέλο να εξετάσει όλα τα αποσπάσματα ταυτόχρονα, ακολουθείται μια **σειριακή διαδικασία**:

1. Το πρώτο απόσπασμα χρησιμοποιείται για τη δημιουργία μιας αρχικής απάντησης.

2. Κάθε νέο απόσπασμα αξιολογείται και, αν περιέχει χρήσιμες πληροφορίες, η απάντηση **αναθεωρείται και εμπλουτίζεται**.
3. Η διαδικασία συνεχίζεται έως ότου εξαντληθεί όλο το διαθέσιμο context.

Με αυτόν τον τρόπο η απάντηση που παράγεται είναι το αποτέλεσμα **συσσώρευσης και ενοποίησης πληροφορίας** από πολλαπλές πηγές, γεγονός που μειώνει τον κίνδυνο παραλείψεων ή αντιφάσεων.

4. Υλοποίηση του μηχανισμού

Στο πλαίσιο της εργασίας υλοποιήθηκαν δύο βασικές παραλλαγές:

- **Refine:**
  Πρόκειται για τον βασικό μηχανισμό, όπου κάθε νέο chunk κειμένου αξιοποιείται για να βελτιώσει ή να διορθώσει την ήδη υπάρχουσα απάντηση. Αν το απόσπασμα δεν προσφέρει νέα πληροφορία, η απάντηση παραμένει αμετάβλητη. Επιπλέον, υποστηρίζεται η χρήση του μοντέλου StructuredRefineResponse, το οποίο ελέγχει αν το υπάρχον context επαρκεί ώστε να ικανοποιηθεί η ερώτηση. Με αυτό τον τρόπο μειώνεται η πιθανότητα παραγωγής μη-τεκμηριωμένων απαντήσεων.
- **CompactAndRefine:**
  Επεκτείνει την παραπάνω μέθοδο, προσθέτοντας έναν μηχανισμό **συμπίεσης (compaction)** των αποσπασμάτων, ώστε αυτά να προσαρμόζονται καλύτερα στα όρια του context window του LLM. Με αυτόν τον τρόπο αποφεύγεται η απώλεια κρίσιμης πληροφορίας λόγω περιορισμών μεγέθους, και το σύστημα διατηρεί την πλήρη εικόνα του γνωσιακού γραφήματος.

Σημασία για το KG-RAG

Η ενσωμάτωση του μηχανισμού Refine προσφέρει τρία βασικά πλεονεκτήματα:

1. **Συνοχή:** η απάντηση βελτιώνεται σταδιακά και ενσωματώνει ομαλά τα νέα στοιχεία, αποφεύγοντας αντιφατικές δηλώσεις.
2. **Φιλτράρισμα θορύβου:** chunks που δεν προσθέτουν αξιόλογη πληροφορία αγνοούνται, με αποτέλεσμα καθαρότερο και πιο αξιόπιστο output.
3. **Επεκτασιμότητα:** ο μηχανισμός μπορεί να χρησιμοποιηθεί ανεξάρτητα από τον retriever ή το είδος των δεδομένων, καθιστώντας το pipeline ευέλικτο για διαφορετικά σενάρια.

Συνολικά, ο μηχανισμός σύνθεσης απαντήσεων λειτουργεί ως ο «τελικός σταθμός» του KG-RAG, μετατρέποντας τις ακατέργαστες πληροφορίες από τον knowledge graph και τα κείμενα σε μια συνεκτική και υψηλής ποιότητας απάντηση, κατάλληλη για αξιολόγηση και χρήση σε πραγματικά συστήματα ερώτησης-απάντησης.

4.4.6 Συνολική εκτίμηση

Η υλοποίηση του KG-RAG εισάγει ένα επίπεδο **σημασιολογικής κατανόησης** που δεν υπήρχε στα προηγούμενα συστήματα. Μέσω της αξιοποίησης του Knowledge Graph, το σύστημα μπορεί να απαντά πιο ακριβώς σε πολυβηματικές ερωτήσεις, να παρέχει επεξηγήσιμες απαντήσεις και να μειώνει την πιθανότητα δημιουργίας λανθασμένων πληροφοριών (hallucinations). Ωστόσο, η διαδικασία συνεπάγεται αυξημένη πολυπλοκότητα, εξάρτηση από την ποιότητα του ίδιου του Knowledge Graph και μεγαλύτερο υπολογιστικό κόστος. Αυτοί οι παράγοντες θα εξεταστούν αναλυτικά στην πειραματική αξιολόγηση.

4.5 Δημιουργία Knowledge Graph από Κειμενικά Δεδομένα

Η αξιοποίηση ενός Knowledge Graph στο πλαίσιο της παρούσας εργασίας απαιτεί πρώτα τη δημιουργία του γράφου από τα διαθέσιμα κειμενικά δεδομένα. Η βασική ιδέα είναι ότι κάθε απόσπασμα κειμένου μπορεί να μετατραπεί σε ένα σύνολο από τριπλέτες (triplets) της μορφής *(οντότητα – σχέση – οντότητα)*, οι οποίες συγκροτούν τη δομή του γραφήματος.

Για την εξαγωγή των τριπλετών χρησιμοποιήθηκε το LLM Ollama (Mistral-small:24b) το οποίο καθοδηγείται μέσω ειδικά σχεδιασμένων προτροπών (prompts). Το μοντέλο καλείται να εντοπίσει ρητά δηλωμένες σχέσεις στο κείμενο και να τις αποδώσει σε κανονικοποιημένη μορφή.

Η διαδικασία ακολουθεί τα εξής βήματα:

1. **Κατασκευή προτροπής (prompting):** Παρέχονται στο μοντέλο παραδείγματα εισόδου/εξόδου, όπου ένα κείμενο συνοδεύεται από τις αντίστοιχες τριπλέτες. Έτσι το μοντέλο μαθαίνει το ζητούμενο φορμάτ [52]. Το ακριβές prompt που χρησιμοποιήθηκε φαίνεται στην *Εικόνα 13*.

Prompt for Triplet Extraction

Instruction:
Extract informative triplets directly from the text following the examples. Do not add any extra words, line breaks, or spaces.
Example 1:
Text: Scott Derrickson (born July 16, 1966) is an American director, screenwriter and producer.
Triplets:
<Scott Derrickson, born in, 1966>,
<Scott Derrickson, nationality, America>,
<Scott Derrickson, occupation, director>,
<Scott Derrickson, occupation, screenwriter>,
<Scott Derrickson, occupation, producer>
Example 2:
Text: A Kiss for Corliss is a 1949 American comedy film directed by Richard Wallace and written by Howard Dimsdale.
Triplets:
<A Kiss for Corliss, year, 1949>,
<A Kiss for Corliss, country, America>,
<A Kiss for Corliss, genre, comedy film>,
<A Kiss for Corliss, director, Richard Wallace>,
<A Kiss for Corliss, writer, Howard Dimsdale>

Target Text: <target text>
Triplets:

Εικόνα 13 - Prompt που χρησιμοποιήθηκε για την εξαγωγή τριπλετών από το κείμενο

2. **Εξαγωγή σχέσεων:** Για κάθε αποσπασματικό κείμενο (π.χ. μία πρόταση από το dataset), το μοντέλο επιστρέφει τριπλέτες όπως:
    - <Scott Derrickson##born in##1966>
    - <Scott Derrickson##occupation##director>
3. **Φιλτράρισμα και καθαρισμός:** Οι παραγόμενες τριπλέτες ελέγχονται ώστε να αποκλείονται περιπτώσεις με μη έγκυρες τιμές (π.χ. unknown, null, ή περιττές επαναλήψεις). Επίσης, γίνεται έλεγχος ώστε οι σχέσεις και οι οντότητες να αντιστοιχούν όντως στο αρχικό κείμενο.
4. **Αποθήκευση σε αρχεία JSON:** Για κάθε οντότητα (π.χ. τίτλο άρθρου) δημιουργείται ένα υπο-αρχείο JSON που περιλαμβάνει τις σχετικές τριπλέτες. Έτσι σχηματίζεται σταδιακά το πλήρες Knowledge Graph.

Το τελικό γράφημα έχει τη μορφή συνόλου από **υπο-γραφήματα** ανά οντότητα, όπου κάθε κόμβος αντιπροσωπεύει μια οντότητα (π.χ. *Shirley Temple*) και κάθε ακμή μια σχέση (π.χ.

served as → Chief of Protocol). Η οργάνωση σε μορφή JSON διευκολύνει την ενσωμάτωση στον μηχανισμό αναζήτησης του KG-RAG.

Η αυτόματη εξαγωγή τριπλετών παρέχει:

- **Δομημένη αναπαράσταση γνώσης**, η οποία υπερβαίνει τη γραμμική φύση του κειμένου.
- **Σαφήνεια στις σχέσεις μεταξύ οντοτήτων**, ώστε ο μηχανισμός ανάκτησης να βασίζεται σε σημασιολογικά συνδεδεμένη πληροφορία.
- **Επαναχρησιμοποίηση**: Ο ίδιος γράφος μπορεί να χρησιμοποιηθεί και σε άλλα ερωτήματα ή πειράματα, χωρίς την ανάγκη επανεπεξεργασίας των αρχικών κειμένων.

Με τον τρόπο αυτό, ολοκληρώνεται η διαδικασία δημιουργίας του Knowledge Graph, ο οποίος στη συνέχεια ενσωματώνεται στο KG-RAG pipeline, ενισχύοντας την ικανότητα του συστήματος να απαντά σε σύνθετα ερωτήματα πολλαπλών βημάτων.

5 Αξιολόγηση

5.1 Dataset – HotpotQA

Για την αξιολόγηση των μεθόδων που υλοποιήσαμε, χρησιμοποιήσαμε το HotpotQA dataset [53], το οποίο αποτελεί ένα από τα πιο καθιερωμένα σύνολα δεδομένων για το πρόβλημα του multi-hop question answering (QA). Σε αντίθεση με πιο απλά datasets, όπου η απάντηση μπορεί να εντοπιστεί σε μία μόνο πρόταση, το HotpotQA έχει σχεδιαστεί ώστε η απάντηση να απαιτεί συνδυασμό πληροφοριών από πολλαπλά κείμενα της Wikipedia. Έτσι, αξιολογεί όχι μόνο την ικανότητα ενός συστήματος να ανακτά πληροφορία, αλλά και να πραγματοποιεί συνδυαστικές σκέψεις πηγαίνοντας από τη μία πληροφορία στην επόμενη (reasoning) προκειμένου να καταλήξει στη σωστή απάντηση.

Το dataset περιλαμβάνει περισσότερες από 112.000 ερωτήσεις, καθεμία από τις οποίες συνοδεύεται από:

- το κείμενο πηγής,
- τη σωστή τεκμηριωμένη απάντηση,
- καθώς και ένα σύνολο από supporting facts, δηλαδή προτάσεις που δικαιολογούν το γιατί η συγκεκριμένη απάντηση είναι η ορθή.

Με τον τρόπο αυτό, το HotpotQA δίνει τη δυνατότητα όχι μόνο να αξιολογηθεί η ακρίβεια ενός συστήματος, αλλά και ο βαθμός στον οποίο οι παραγόμενες απαντήσεις είναι ερμηνεύσιμες και τεκμηριωμένες.

Στην παρούσα εργασία επιλέξαμε να χρησιμοποιήσουμε την εκδοχή Distractor Setting. Σε αυτήν την εκδοχή, για κάθε ερώτηση παρέχονται δέκα υποψήφια κείμενα (passages), από τα οποία μόνο τα δύο είναι σχετικά με την απάντηση, ενώ τα υπόλοιπα οκτώ λειτουργούν ως distractors, εισάγοντας θόρυβο και παραπλανητική πληροφορία. Η επιλογή αυτή έγινε διότι:

1. Προσφέρει ένα ελεγχόμενο περιβάλλον αξιολόγησης, χωρίς να απαιτείται αναζήτηση σε ολόκληρη τη Wikipedia.
2. Αντικατοπτρίζει πιο ρεαλιστικά σενάρια πληροφόρησης, όπου το ζητούμενο είναι η διάκριση ανάμεσα σε χρήσιμη και άσχετη πληροφορία.

3. Είναι ιδιαίτερα διαδεδομένο στη βιβλιογραφία για την αξιολόγηση RAG συστημάτων, γεγονός που επιτρέπει συγκρίσεις με άλλες μεθόδους.[13], [29]

Η επιλογή του HotpotQA Distractor Set εξασφαλίζει ότι η αξιολόγηση των μεθόδων RAG που υλοποιήθηκαν βασίζεται σε ένα απαιτητικό και καθιερωμένο benchmark, το οποίο αναγκάζει τα συστήματα όχι μόνο να ανακτούν πληροφορίες αλλά και να επιδεικνύουν ικανότητες συλλογιστικής πολλαπλών βημάτων (multi-hop easoning), στοιχείο καίριο για την επιτυχία τους σε πραγματικές εφαρμογές.

5.1.1 Προεπεξεργασία Δεδομένων

Προκειμένου να είναι δυνατή η αξιολόγηση των τριών διαφορετικών υλοποιήσεων RAG, πραγματοποιήθηκε προ επεξεργασία του HotPotQA Distractor Set ώστε τα δεδομένα να μετατραπούν σε μορφή κατάλληλη για την κατασκευή του συνόλου γνώσης κάθε μεθόδου.

Κάθε στοιχείο του HotPotQA περιέχει:

- id (μοναδικό αναγνωριστικό)
- question (η ερώτηση)
- answer (η σωστή απάντηση)
- supporting_facts (που βρίσκονται οι πληροφορίες για την απάντηση της ερώτησης)
- context: λίστα από ζεύγη (τίτλος άρθρου, λίστα προτάσεων)

Ακολουθεί παράδειγμα στον *Πίνακα 1*:

id	*5a8b57f25542995d1e6f1371*
answer	yes
question	Were Scott Derrickson and Ed Wood of the same nationality?
supporting_facts	["Scott Derrickson",0],["Ed Wood",0]
context	Paragraphs with titles: Ed Wood (film), Scott Derrickson, Woodson, Arkansas, Tyler Bates, Ed Wood, Adam Collis, Sinister (film), Conrad Brooks, Doctor Strange (2016 film)

Πίνακας 1 - Παράδειγμα instance από το HotPotQA dataset

Προεπεξεργασία ανά μέθοδο:

1. Naive RAG

Στην απλή εκδοχή, όλες οι προτάσεις από τα context documents αποθηκεύτηκαν ως ξεχωριστά Document objects. Κάθε document είχε ως περιεχόμενο μία πρόταση και metadata (τίτλος άρθρου, sentence id). Αυτό αποσκοπούσε στο να είναι δυνατή η αξιολόγηση του συστήματος χρησιμοποιώντας το evaluation script του Hotpot, καθώς αυτό απαιτεί τα retrieved chunks να είναι ξεκάθαρο από ποια παράγραφο και ποια πρόταση προέρχονται. Ουσιαστικά, ενώθηκαν όλες οι παράγραφοι του dataset σε μια μεγάλη συλλογή προτάσεων, η οποία αποτέλεσε το corpus γνώσης για dense retrieval.
Αυτό έχει το πλεονέκτημα της απλότητας, αλλά οδηγεί σε πολύ μεγάλο πλήθος μικρών τεκμηρίων, χωρίς ιεραρχία ή σύνδεση μεταξύ τους.

2. RAPTOR RAG

Για το RAPTOR, κάθε δείγμα μετατράπηκε σε ένα ενιαίο κείμενο ανά άρθρο, όπου τίτλος και προτάσεις ενώθηκαν σε μία παράγραφο. Δηλαδή, από το context φτιάχτηκε ένα μεγάλο block κειμένου ανά τίτλο. Έτσι, κάθε row δεδομένων (δηλαδή κάθε άρθρο με όλες τις προτάσεις του) πέρασε στη διαδικασία κατασκευής του **RAPTOR tree**.
Το RAPTOR tree δομήθηκε ιεραρχικά (max_levels=3), δημιουργώντας **multi-level embeddings** που επιτρέπουν πιο έξυπνη αναζήτηση (top-down refinement).

3. KG-RAG

Για την εκδοχή με Knowledge Graph RAG, από κάθε πρόταση έγινε προσπάθεια εξαγωγής τριπλετών (h, r, t) ακολουθώντας τη διαδικασία που περιγράφτηκε στην παράγραφο 4.5. Οι τριπλέτες αποθηκεύτηκαν σε JSON αρχεία ανά entity/άρθρο, δημιουργώντας ένα structured sub-knowledge graph.
Έτσι, η βάση γνώσης δεν αποτελείται πλέον από αδόμητο κείμενο, αλλά από σχέσεις μεταξύ οντοτήτων, γεγονός που επιτρέπει ερωτήσεις πιο κοντά σε graph traversal και knowledge-grounded reasoning.

Με αυτόν τον τρόπο, η ίδια πηγή δεδομένων (HotpotQA) χρησιμοποιήθηκε με τρεις διαφορετικούς τρόπους αναπαράστασης:

- Naive → απλό corpus από προτάσεις (flat documents).
- RAPTOR → ιεραρχικά blocks κειμένου με δενδρική δόμηση.
- KG-RAG → γραφο-δομημένη γνώση με τριπλέτες.

5.1.2 Evaluation Script του HotpotQA

Το HotpotQA συνοδεύεται από ένα επίσημο evaluation script το οποίο χρησιμοποιείται εκτενώς στη βιβλιογραφία για τη σύγκριση μεθόδων. Το script αυτό υπολογίζει μια σειρά από μετρικές που επιτρέπουν την αξιολόγηση τόσο της τελικής απάντησης όσο και της ορθότητας των supporting facts επιτρέποντας έτσι την αξιολόγηση τόσο του Retriever όσο και του Augmentor του συστήματός μας.

Οι βασικές μετρικές που παράγει είναι εκείνες του παρακάτω *Πίνακα 2*:

Exact Match (EM)	Ελέγχει αν η απάντηση που δίνει το σύστημα είναι ακριβώς ίδια με την ground-truth απάντηση (μετά από κανονικοποίηση: lowercasing, αφαίρεση σημείων στίξης και άρθρων)
F1 Score	Υπολογίζει τον συνδυασμό **precision** και **recall** σε επίπεδο tokens ανάμεσα στην πρόβλεψη και την πραγματική απάντηση.
Precision	Ελέγχει από όλα όσα προέβλεψε το μοντέλο ως "σωστά", πόσα ήταν πράγματι σωστά
Recall	Ελέγχει από όλα τα πραγματικά σωστά στοιχεία (ground truth), πόσα τα βρήκε το μοντέλο
Supporting Facts Metrics	Αντίστοιχες μετρικές (EM, precision, recall, F1) υπολογίζονται και για τις προτάσεις που δηλώνονται ως τεκμηριωτικές
Joint Metrics	Συνδυασμός των παραπάνω, ώστε να μετράται ταυτόχρονα αν το σύστημα πέτυχε και σωστή απάντηση και σωστά supporting facts

Πίνακας 2 - Κλασσικές Μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση

Η χρήση του script είναι ιδιαίτερα σημαντική καθώς διασφαλίζει αντικειμενική και συγκρίσιμη αξιολόγηση ανάμεσα σε διαφορετικές μεθοδολογίες. Επιπλέον, η ύπαρξη μετρικών για τα supporting facts καθιστά το HotpotQA μοναδικό, καθώς αξιολογεί όχι μόνο το αν βρέθηκε η σωστή απάντηση αλλά και το αν αυτή μπορεί να δικαιολογηθεί με βάση τις παρεχόμενες πηγές.

5.2 DeepEval

Το **DeepEval** είναι ένα open-source framework αξιολόγησης (evaluation framework) για LLM εφαρμογές, σχεδιασμένο ώστε να διευκολύνει unit testing και αυτοματοποιημένη αξιολόγηση.

Μερικά χαρακτηριστικά του:

- επιτρέπει να ορίζεις test cases, δηλαδή είσοδο (ερώτηση), αναμενόμενη έξοδο, και retrieval context, και να τρέχεις μετρικές πάνω σε αυτά.
- Υποστηρίζει διάφορες μετρικές αξιολόγησης, τόσο για τον retrieval (context) όσο και για τη γεννήτρια (answer).
- Μπορεί να ενσωματωθεί με άλλα εργαλεία όπως LlamaIndex ή Haystack για να αξιολογήσεις pipelines RAG.
- Σου επιτρέπει να ορίσεις thresholds για μετρικές και να κάνεις assertions για το αν περνά η εκάστοτε περίπτωση δοκιμής.
- Παρέχει μετρικές σχετικές με contextual precision / recall / relevancy, answer relevancy, faithfulness κ.ά.

Ένα από τα πιο ενδιαφέροντα χαρακτηριστικά του DeepEval είναι ότι αξιοποιεί την προσέγγιση **LLM-as-a-judge**: αντί να βασίζεται αποκλειστικά σε στατιστικές ή ντετερμινιστικές μετρικές (π.χ. BLEU, ROUGE), χρησιμοποιεί ένα άλλο μεγάλο γλωσσικό μοντέλο ως "κριτή" για να αξιολογήσει την ποιότητα της απάντησης.
Με αυτόν τον τρόπο, το σύστημα μπορεί να εκτιμήσει πιο "νοηματικά" στοιχεία, όπως σχετικότητα, πληρότητα, σαφήνεια και πιστότητα στο context, τα οποία οι παραδοσιακές μετρικές δεν μπορούν εύκολα να αποτυπώσουν επιτρέποντας στο DeepEval να παρέχει πιο ανθρώπινα ρεαλιστικές αξιολογήσεις.

Το DeepEval, κατά κάποιον τρόπο, λειτουργεί σαν "pytest για LLMs / RAG" — δηλαδή, να έχεις αυτοματοποιημένες δοκιμές τις οποίες τρέχεις και βλέπεις αν το σύστημα σου περνά ή αποτυγχάνει βάση μετρικών.

5.2.1 Αξιολόγηση RAG συστήματος με DeepEval

Για να αξιολογήσω ένα RAG σύστημα μέσω DeepEval, επέλεξα μετρικές που καλύπτουν τόσο το retrieval όσο και το generation στάδιο.

- **Answer Relevancy**:

Η μετρική Answer Relevancy αξιολογεί πόσο σχετική είναι η απάντηση που παρήγαγε το σύστημα σε σχέση με την ερώτηση (input).

Πιο συγκεκριμένα:

- Χρησιμοποιεί ένα LLM ως κριτή (LLM-as-a-judge) για να αναλύσει την απάντηση του συστήματος και να διασπάσει την πρόταση σε δηλώσεις (statements).
- Έπειτα, για κάθε δήλωση, το ίδιο LLM αποφασίζει αν είναι **σχετική** προς την ερώτηση (input) ή όχι.
- Τέλος, υπολογίζει:

$$Answer\ Relevancy = \frac{Αριθμός\ σχετικών\ δηλώσεων}{Συνολικός\ αριθμός\ δηλώσεων}$$

δηλαδή το ποσοστό των δηλώσεων που είναι "σχετικές" στην ερώτηση

- **Faithfulness**

Η μετρική Faithfulness μετράει το κατά πόσο η απάντηση που παρήγαγε το σύστημα είναι συμβατή / συνεπής με τα τεκμήρια (retrieval context) που επέλεξε.

Πιο αναλυτικά:

- Χρησιμοποιείται **LLM-as-a-judge**, δηλαδή ένα μοντέλο γλώσσας λειτουργεί ως «κριτής», για να αναλύσει την απάντηση (actual output) και να εξάγει **δηλώσεις** (claims) που κάνει η απάντηση.
- Στη συνέχεια, για κάθε δήλωση, το ίδιο LLM ελέγχει αν **αντιφάσκει με τα τεκμήρια** που του έδωσε ως context (retrieval context). Δηλαδή, εάν η δήλωση αντίκειται ή συγκρούεται με πληροφορίες από τα τεκμήρια, η δήλωση θεωρείται **μη αληθής**.
- Η βαθμολογία **Faithfulness** υπολογίζεται ως:

$$Faithfulness\ = \frac{Αριθμός\ δηλώσεων\ που\ κρίθηκαν\ ως\ αληθείς}{Συνολικός\ αριθμός\ δηλώσεων}$$

δηλαδή το ποσοστό των δηλώσεων που δεν έρχονται σε αντίφαση με τα τεκμήρια.

Έτσι, η μετρική αυτή προσφέρει έλεγχο κατά πόσο το σύστημα «ψάχνει» από τα τεκμήρια και δεν «φαντάζεται» στοιχεία που δεν στηρίζονται σε αυτά — δηλαδή αν η απάντηση είναι πιστή στην τεκμηριωμένη γνώση που παρήχθη.

- **Contextual Precision**

Η μετρική Contextual Precision στο μετράει πόσο καλά το σύστημα ταξινομεί / τοποθετεί τα τεκμήρια (nodes) που ανακτά — δηλαδή πόσο "καθαρό" είναι το ανακτημένο περιεχόμενο όσον αφορά στη σχετικότητα με την ερώτηση και την αναμενόμενη απάντηση.

Πιο αναλυτικά:

- Χρησιμοποιείται LLM ως κριτής (LLM-as-a-judge) για να αποφασίσει, για κάθε node (τεκμήριο / κομμάτι context) που επέλεξε το σύστημα, αν αυτό είναι σχετικό ή όχι ως προς την είσοδο (input) και την αναμενόμενη έξοδο (expected output).
- Κατόπιν, η Contextual Precision υπολογίζεται με έναν σταθμισμένο συσσωρευτικό τρόπο (weighted cumulative precision):
    - Έστω η retrieval_context έχει n nodes, διατεταγμένα σε σειρά.
    - Για κάθε θέση k ($1 \leq k \leq n$), το σύστημα εξετάζει πόσα από τα πρώτα k nodes είναι σχετικά.
    - Καθεμία από αυτές τις "ανακτήσεις μέχρι τη θέση k" παίρνει βάρος / συμβολή, έτσι ώστε οι πρώτες θέσεις (τα κορυφαία τεκμήρια) να μετράνε περισσότερο. deepeval.com
- Ο τύπος δίνει έμφαση στο ότι τα πιο πάνω-ranked τεκμήρια πρέπει να είναι τα πιο σχετικά, δηλαδή δεν αρκεί να έχεις πολλά σωστά τεκμήρια· πρέπει να τα "βρίσκεις" πρώτα. deepeval.com
- Το αποτέλεσμα είναι ένας βαθμός (score) μεταξύ 0 και 1, όπου υψηλό σκορ δείχνει ότι το σύστημα κατάφερε να ταξινομεί τα σχετικά τεκμήρια πρώτα (υψηλή "καθαρότητα" του retrieved context).

Παράδειγμα:

Ερώτηση	Ποιο είναι το όνομα του σκηνοθέτη της ταινίας *A Kiss for Corliss*;
Απαντηση	Richard Wallace
Retrieved Context	1. "A Kiss for Corliss is a 1949 American comedy film directed by Richard Wallace and written by Howard Dimsdale." 2. "Shirley Temple was named United States ambassador to Ghana and to Czechoslovakia." 3. "Shirley Temple was an American actress, singer, dancer, businesswoman, and diplomat."

Σε αυτό το παράδειγμα, το σύστημα επέλεξε τρία τεκμήρια (nodes). Το πρώτο τεκμήριο περιέχει ξεκάθαρα την πληροφορία "directed by Richard Wallace" και είναι **σχετικό** με την ερώτηση. Τα άλλα δύο δεν σχετίζονται με τον σκηνοθέτη της ταινίας· δεν δίνουν πληροφορία για το "directed by" και είναι άσχετα για τον σκοπό της ερώτησης. Άρα, σε αυτό το παράδειγμα,

η **Contextual Precision = 1.0**, διότι το μόνο σχετικό τεκμήριο ήταν πρώτο — δηλαδή ο "καλός" κόμβος μπήκε πρώτος.

- **Contextual Recal**

Η μετρική Contextual Recall μετράει το πόσο καλά το σύστημα retriever ανασύρει όλες τις απαραίτητες πληροφορίες (τεκμήρια / nodes) που απαιτούνται για να στηριχθεί η αναμενόμενη απάντηση.

Πιο αναλυτικά, τι κάνει:

- Χρησιμοποιεί **LLM ως "κριτή"** (LLM-as-a-judge) για να διασπάσει την **αναμενόμενη απάντηση (expected_output)** σε δηλώσεις (statements).
- Για κάθε μία από αυτές τις δηλώσεις, το μοντέλο κρίνει αν μπορεί να "αποδοθεί" (attributed) σε κάποιο node (τεκμήριο) από το **retrieval_context** που επέλεξε ο retriever. Δηλαδή: «Μπορεί αυτή η δήλωση να υποστηριχτεί από κάποιο από τα ανακτηθέντα τεκμήρια;»
- Το Contextual Recall υπολογίζεται ως:

$$Contextual\ Recall\ = \frac{Αριθμός\ δηλώσεων\ στα\ τεκμήρια}{Συνολικός\ αριθμός\ δηλώσεων}$$

- Όσο μεγαλύτερο το σκορ, τόσο καλύτερα ο retriever έχει καλύψει τις δηλώσεις που χρειάζονται για την απάντηση — δηλαδή έχει **πλήρη ανάκτηση των απαραίτητων πληροφοριών** από τα τεκμήρια.

- **Contextual Relevancy**

Η μετρική Contextual Relevancy μετράει πόσο σχετικό είναι το σύνολο των πληροφοριών (τεκμηρίων / context) που επέλεξε το σύστημα προς την ερώτηση (input).

Αναλυτικότερα:

- Ο "κριτής" (LLM-as-a-judge) εξάγει **δηλώσεις (statements)** από τα τεκμήρια που ανακτήθηκαν στο retrieval_context.
- Στη συνέχεια, το ίδιο LLM αξιολογεί για κάθε δήλωση αν είναι **σχετική** προς την ερώτηση (input).
- Η βαθμολογία υπολογίζεται ως:

$$Contextual\ Relevancy = \frac{Αριθμός\ σχετικών\ δηλώσεων}{Συνολικός\ αριθμός\ δηλώσεων\ στα\ τεκμήρια}$$

Δηλαδή, το ποσοστό των δηλώσεων μέσα στο context που κρίνονται ως ουσιαστικά συνεισφέροντα προς την ερώτηση.

- **G-Eval**

Η G-Eval είναι μια γενική μετρική αξιολόγησης βασισμένη σε LLM, που επιτρέπει την ορισμό κριτηρίων (criteria) για το τι σημαίνει «καλή απάντηση» σε ένα συγκεκριμένο context. Είναι ιδανική για περιπτώσεις όπου οι παραδοσιακές μετρικές (όπως F1, EM) δεν αποτυπώνουν πλήρως την ποιότητα, ιδίως όσον αφορά την συνέπεια, την ακρίβεια και την πληρότητα της πληροφορίας στην απάντηση.

Στη δικιά μας περίπτωση θα χρησιμοποιήσουμε την πιο δημοφιλή G-Eval του DeepEval η οποία είναι η **Correctness**. Η συγκεκριμένη μετρική αξιολογεί το κατά πόσο η απάντηση του Generator (actual output) είναι όντως σωστή με βάση την αναμενόμενη απάντηση (expected output).

Αρκετά ενδιαφέρον και σημαντικό είναι το γεγονός ότι οι μετρικές του DeepEval είναι **self-explaining**, δηλαδή μαζί με το σκορ παράγουν και ένα "λόγο / αιτιολόγηση" (reason) από το LLM για το γιατί έδωσε αυτό το σκορ. Αυτή η ιδιότητα είναι πολύ χρήσιμη για την κατανόηση της συμπεριφοράς του συστήματος μας.

5.3 Αποτελέσματα

5.3.1 Πειράματα σε μικρό μέρος του Dataset

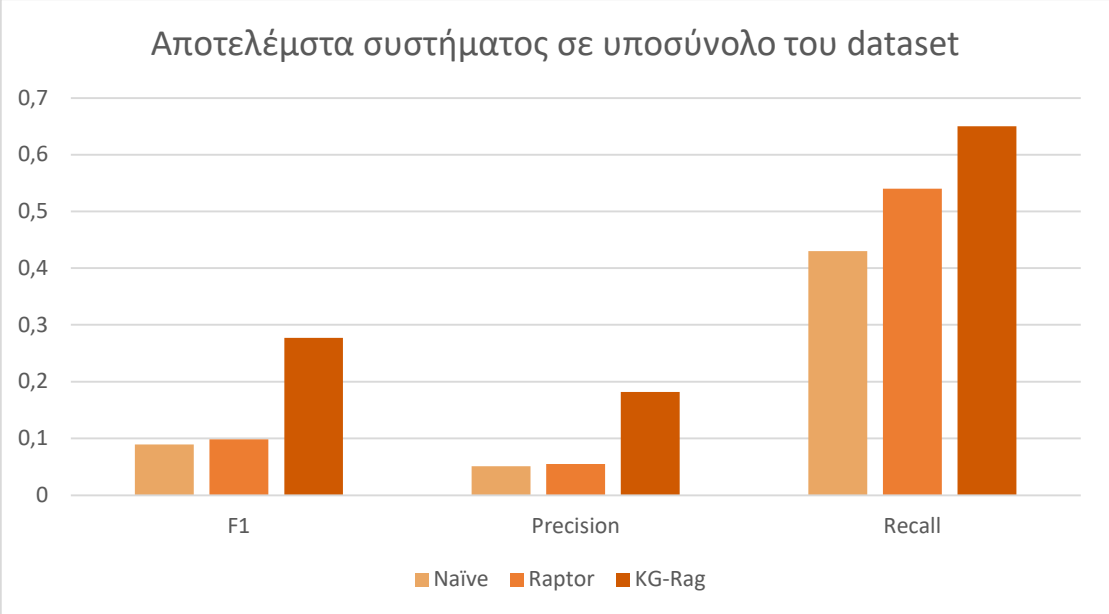
Έτσι ξεκινήσαμε με τον πειραματισμό σε ένα μικρότερο σύνολο των δεδομένων ώστε να μπορούμε πιο άμεσα και γρήγορα να αποκτήσουμε μια εποπτεία του τι συμβαίνει στα συστήματα μας και πως ανταποκρίνονται οι μέθοδοι μας καθώς και να μπορούμε να κάνουμε τις κατάλληλες αλλαγές μέχρι να επέλθουμε στο επιθυμητό αποτέλεσμα και να συνεχίσουμε με τα πειράματα σε ολόκληρο το dataset. Οι αλλαγές αυτές είχαν να κάνουν κυρίως με τη σωστή αποθήκευση και μορφή που έπρεπε να επιβάλουμε στις προβλέψεις μας ώστε να είναι σε θέση μετά να αξιολογηθούν. Να σημειώσουμε βέβαια ότι κάθε instance του dataset περιείχε πληθώρα πληροφοριών (περίπου 10 διαφορετικές παράγραφοι σε κάθε instance) οπότε το σύνολο της γνώσης που δημιουργούταν ήταν αρκετό για να μπορέσουμε να βγάλουμε κάποια αρχικά συμπεράσματα για τη συμπεριφορά κάθε μεθόδου.

5.3.1.1 Αποτελέσματα F1, Precision, Recall

Παρακάτω στους *Πίνακες 3* και στην *Εικόνα 14* παραθέτουμε τα αποτελέσματα των 3 μεθόδων στα πρώτα πειράματα μας με μικρό μέρος του Dataset. Με έντονα γράμματα σημειώνονται τα καλύτερα αποτελέσματα για κάθε μετρική.:

ΕΠΙΠΕΔΟ	ΜΕΘΟΔΟΙ	EM	F1	PRECISION	RECALL
GENERATOR	Naive	0.5	0.607	0.633	0.64
	Raptor	0.6	0.657	0.7	0.64
	KG-Rag	**0.7**	**0.757**	**0.8**	**0.74**
RETRIEVER	Naïve	-	0.124	0.0702	0. 656
	Raptor	-	0.133	0.073	0.8
	KG-Rag	-	**0. 354**	**0.227**	**0.877**
SYSTEM	Naïve	-	0.089	0.051	0. 43
	Raptor	-	0.098	0.055	0.54
	KG-Rag	-	**0. 277**	**0.182**	**0.65**

Πίνακας 3- Αποτελέσματα κλασσικών metrics στο μικρό μέρος του dataset



Εικόνα 14 - Σχηματική αναπαράσταση των metrics F1, Precision και Recall σε υποσύνολο του dataset

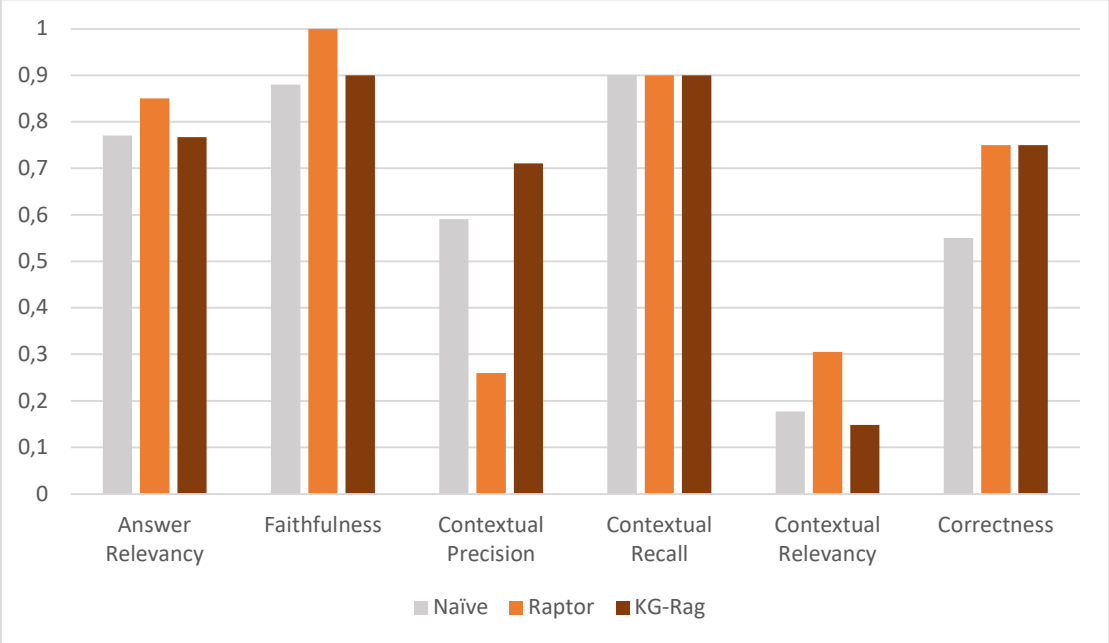
Από τα αποτελέσματα βλέπουμε ότι οι υλοποιήσεις **RAPTOR** και **KG-RAG** ξεπερνούν την **Naive** σε όλες τις μετρικές. Η αύξηση του F1 και ειδικά του precision υποδηλώνει ότι οι πιο προηγμένες μέθοδοι επιτυγχάνουν πιο καθαρές και συναφείς απαντήσεις.

Οι τιμές recall υποδηλώνουν ότι ενώ η Naive κι η RAPTOR ανακτούν σημαντικό ποσοστό των σωστών πληροφοριών (0.64), η KG-RAG φέρνει ακόμη μεγαλύτερο ποσοστό (0.74), αποτελεσματικά μειώνοντας τις "ελλείψεις".

Επιπλέον, αν εξετάσουμε τα αποτελέσματα που σχετίζονται με την συμπεριφορά του **Retriever**, η KG-RAG δείχνει σημαντικά ανώτερη από τις άλλες (F1 ≈ 0.354), αποκαλύπτοντας ότι ο retriever της επιλέγει πολύ πιο κατάλληλα τεκμήρια. Αυτό ενισχύει την υπόθεση ότι οι γράφοι γνώσης (knowledge-graphs) βοηθάνε στο να αποφύγεις άσχετα τεκμήρια που θα "μπερδέψουν" τον generator.

Συνολικά, τα αποτελέσματα καταδεικνύουν ότι η χρήση πιο δομημένων μεθόδων (RAPTOR, KG-RAG) προσφέρει σαφή πλεονεκτήματα σε σχέση με τη Naive, ειδικά στο κομμάτι της καθαρότητας και της λογικής στήριξης των απαντήσεων.

5.3.1.2 Αποτελέσματα μετρικών του DeepEval



Εικόνα 15 - Αποτελέσματα των DeepEval metrics σε υποσύνολο του dataset

Εδώ τα αποτελέσματα της *Εικόνας 15* δεν είναι τόσο ξεκάθαρα και δεν δείχνουν την ίδια συμπεριφορά που ξεκάθαρα παρατηρήσαμε στις προηγούμενες μετρικές (δηλαδή ότι η Naive ήταν η χειρότερη σε απόδοση μέθοδος με καλύτερή της την Raptor και ακόμα καλύτερη την KG-RAG). Βέβαια αυτή η συμπεριφορά μπορεί να παρατηρηθεί στην μετρική **Correctness**, η

οποία μας δείχνει πόσες σωστές απαντήσεις παράγει η κάθε μέθοδος, στην οποία Raptor και KG-RAG ισοβαθμούν παράγοντας 7,5/10 σωστές απαντήσεις σε σύγκριση με τη Naive που παράγει 5,5/10 σωστές.

Το DeepEval προσφέρει την πολύτιμη δυνατότητα να παρέχει εξήγηση για κάθε βαθμολογία που παράγει οπότε μπορούμε να ερευνήσουμε πιο αναλυτικά τι συμβαίνει στο σύστημά μας.

Όσον αφορά την **Answer Relevancy**:

Στις περιπτώσεις που παρατηρείται μειωμένο σκορ έχουν δοθεί οι εξής επεξηγήσεις:

Test case 1:

Question	*What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?*
Predicted Answer	Ambassador to Ghana.
Score	0.67
Reason	*"The score is 0.67 because while the response may have provided some relevant information, it included irrelevant details such as the word 'to', which did not address the specific question about the government position held by the woman who portrayed Corliss Archer in the film Kiss and Tell."*

Πίνακας 4 - Test case 1

Δηλαδή σε αυτή την περίπτωση του *Πίνακα 4* το σκορ μειώθηκε κατά 1/3 απλώς και μόνο επειδή υπήρχε κι η λέξη "to" στην απάντηση.

Test case 2:

Question	*Are the Laleli Mosque and Esma Sultan Mansion located in the same neighborhood?*
Predicted Answer	No.
Expected Answer	no
Score	0.0

Reason	*"The score is 0.00 because the response 'No.' does not address the specific locations of Laleli Mosque and Esma Sultan Mansion, making it irrelevant to the input question."*

Πίνακας 5 - Test case 2

Δηλαδή αυτή η περίπτωση του *Πίνακα 5* βαθμολογήθηκε με 0, ενώ στην πραγματικότητα είναι μια σωστή απάντηση.

Συμπέρασμα:

Και στις δύο παραπάνω περιπτώσεις είχαμε ένα χαμηλό σκορ το οποίο όμως δεν αντανακλά μια πραγματικά κακή απόδοση του συστήματος. Αν οι απαντήσεις δεν ήταν τόσο σύντομες και περιεκτικές τα σκορ θα ήταν ανεβασμένα, διότι θα περιέχονταν περισσότερες πληροφορίες που θα έδιναν στον LLM judge evaluator να καταλάβει ότι η απάντηση είναι πράγματι σχετική με την ερώτηση. Οι σύντομες και περιεκτικές απαντήσεις από τον Generator είναι κάτι αναμενόμενο, καθώς τον έχουμε προτρέψει να απαντάει με αυτό τον τρόπο εμπεριέχοντας στον prompt το εξής:

"give a short factoid answer (as few words as possible)"

Test case 3:

Question	*The director of the romantic comedy "Big Stone Gap" is based in what New York city?*
Predicted Answer of KG-RAG	Greenwich Village.
Predicted Answer of Raptor	Greenwich Village
Predicted Answer of Naive	Greenwich Village
Score KG-RAG	0.0
Score Raptor	1.0
Score Naive	1.0

Reason	*"The score is 0.00 because the actual output did not address the question at all, failing to provide any information about the director or the city where they are based."*

Πίνακας 6 - Test case 3

Δηλαδή σε αυτή την περίπτωση του *Πίνακα 6* η ίδια ερώτηση με την ίδια απάντηση και από τις 3 μεθόδους στις περιπτώσεις των Naive και Raptor βαθμολογήθηκαν με 1 και στην περίπτωση του KG-RAG βαθμολογήθηκε με 0.

Συμπέρασμα:

Οι μετρικές του DeepEval χρησιμοποιούν όπως έχουμε πει ένα LLM-as-a-judge για να πάρει αποφάσεις για την βαθμολογία κάθε περίπτωσης (test case). Αυτό οδηγεί σε μια μη ντετερμινιστική συμπεριφορά καθώς δεν μπορούμε να ελέγχουμε τον τρόπο που καθορίζονται οι απόψεις του judge κι έτσι είναι πιθανό να υπάρχουν κάποια λανθασμένα και μη αντιπροσωπευτικά σκορ.

Σύμφωνα με τις παραπάνω παρατηρήσεις και συμπεράσματα και συνυπολογίζοντας το μικρό δείγμα δεδομένων που έχουμε αξιολογήσει σε αυτά τα πειράματα, καταλήγουμε στο ότι είναι εύλογο να υπάρχουν αποκλίσεις στις βαθμολογίες και έχοντας όλες τις μεθόδους να πετυχαίνουν σκορ που είναι αρκετά κοντινά μεταξύ τους δεν μπορούμε να βγάλουμε κάποιο σαφές συμπέρασμα για το ποια μέθοδος υπερέχει. Σημασία έχει που όλες οι μετρικές είναι αρκετά υψηλές και μας δείχνουν μια γενική καλή συμπεριφορά όλων των Generator μας στο να παράγουν απαντήσεις σχετικές με τις ερωτήσεις.

Όσον αφορά την **Faithfulness**:

Όλες οι μέθοδοι έχουν σχεδόν άριστες βαθμολογίες ως προς αυτή τη μετρική, γεγονός που σημαίνει ότι οι Generator τα καταφέρνουν πολύ καλά να αποφεύγουν τα hallucinations και να μην παράγουν πληροφορίες που δεν σχετίζονται με το context που έχουν κάνει retrieve. Άρα κι οι 3 μέθοδοι είναι αξιόπιστες.

Όσον αφορά την **Contextual Precision**:

Καλύτερη επίδοση παρουσιάζει η KG-RAG, δηλαδή καταφέρνει με τον καλύτερο τρόπο να κάνει retrieve και να ταξινομεί τις πληροφορίες από την πιο σημαντική στην πιο ασήμαντη. Παρατηρούμε, επίσης, ότι η Raptor δεν τα πηγαίνει καλά σε αυτό το metric. Αυτό είναι αναμενόμενο αν αναλογιστούμε τον τρόπο που δουλεύει ο retriever της Raptor ο οποίος

ξεκινάει την αναζήτηση από την κορυφή του raptor tree και ύστερα εμβαθύνει στα χαμηλότερα επίπεδα, στα παιδιά κάθε φύλλου, και καταλήγει στη βάση του δέντρου που βρίσκονται οι πληροφορίες αυτές καθ΄ αυτές. Άρα οι πιο λεπτομερείς και κατάλληλες για την απάντηση πληροφορίες βρίσκονται στη βάση του δέντρου και ανακτώνται τελευταίες.

Όσον αφορά την **Contextual Recall**:

Παρατηρούμε ισοβαθμία των τριών μεθόδων με πολύ καλό σκορ (0,9) γεγονός που δείχνει ότι όλες οι μέθοδοι κάνουν πολύ καλό retrieve, όμως βρισκόμαστε στα πειράματα σε μικρό μέρος του Dataset κι έτσι είναι ακόμα μικρό το knowledge base μας με αποτέλεσμα να μην είναι τόσο δύσκολη η ανάκτηση πληροφοριών. Αναμένουμε να δούμε τη συμπεριφορά των μεθόδων σε μεγαλύτερο όγκο δεδομένων.

Όσον αφορά την **Contextual Relevancy**:

Σε όλες τις μεθόδους παρατηρούμε αρκετά χαμηλό σκορ στη συγκεκριμένη μετρική. Αυτό σημαίνει πως οι Retrievers μας μαζί με τις χρήσιμες πληροφορίες ανακτάνε και πολλές μη ωφέλιμες και εισάγουν θόρυβο στο σύστημα. Αυτό είναι κάτι αναμενόμενο αφού δεν υπάρχει κάποιο «φίλτρο» που να εφαρμόζεται πχ στο τέλος και να κρατάει μόνο τα πραγματικά χρήσιμα αλλά αντί για αυτό προσπαθούμε να ανακτούμε το k πιο χρήσιμα έγγραφα. Στο Dataset όμως που έχουμε οι πληροφορίες είναι πολύ διαφορετικές μεταξύ τους και μόνο ένα πολύ μικρό κομμάτι είναι όντως ωφέλιμο για την δημιουργία της απάντησης.

Ταυτόχρονα, στο KG-RAG παρατηρείται ακόμα πιο χαμηλή τιμή σε αυτή τη μετρική. Αυτό συμβαίνει εξαιτίας της τεχνικής που ακολουθεί η μέθοδος μας για να κάνει ανάκτηση πληροφοριών κατά την οποία κάνουμε επέκταση των σημασιολογικά σχετικών πληροφοριών με βάση τον γράφο γνώσης μας. Αυτό προσθέτει πληροφορίες στο context οι οποίες με μια 1η ματιά δεν φαίνονται σημασιολογικά συναφείς, έχει αποδειχθεί όμως, και φαίνεται και από τα δικά μας πειράματα, ότι τελικά βοηθάει στη σύνθεση της τελικής απάντησης.

Τελικό συμπέρασμα:

Συνολικά, το reasoning που παρέχουν οι μετρικές του DeepEval είναι ένα πάρα πολύ πολύτιμο εργαλείο που μπορεί να σε βοηθήσει να κατανοήσεις σε βάθος τι συμβαίνει στο σύστημα σου και να κάνεις τις απαραίτητες τροποποιήσεις μέχρι να καταλήξεις στο επιθυμητό αποτέλεσμα. Σημαντικό είναι να τονιστεί πως πρέπει να χρησιμοποιείται ένα δυνατό και ικανό LLM για το Evaluation. Τέλος, οι μετρικές του DeepEval είναι καλό να χρησιμοποιούνται συνδυαστικά με άλλες πιο απόλυτες μετρικές όπως το F1, precision, recall.

5.3.2 Πειράματα σε μεγαλύτερο μέρος του Dataset

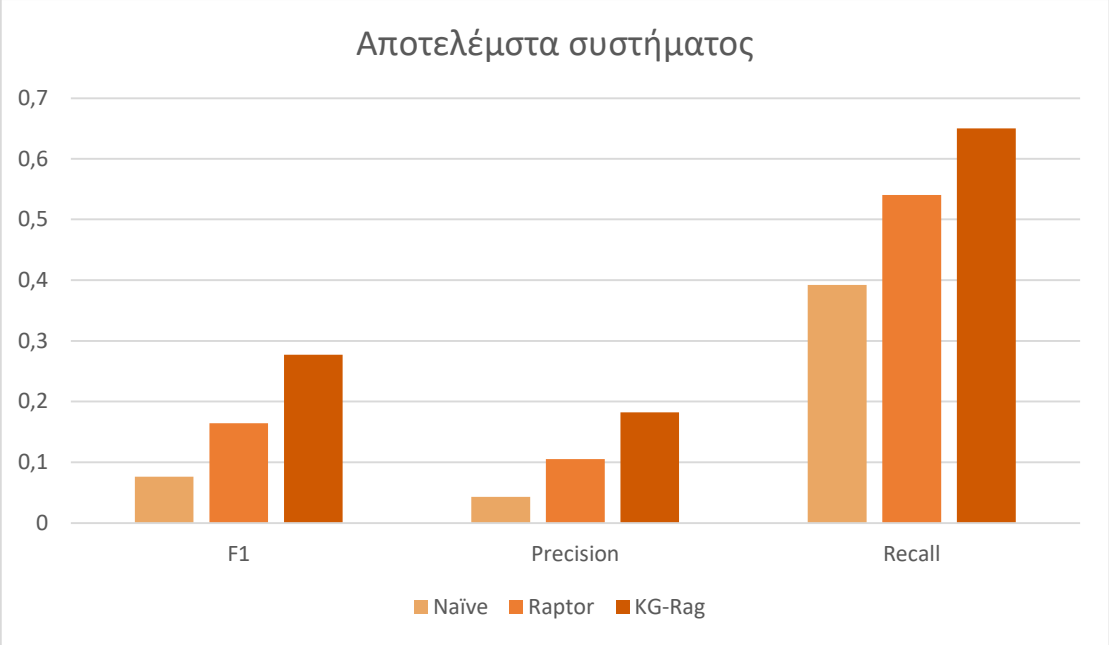
Στη συνέχεια διεξάγαμε πειράματα σε ένα αρκετά μεγαλύτερο μέρος του Dataset για να μπορέσουμε να έχουμε μια πιο ολιστική εποπτεία της ανταπόκρισης και του τρόπου λειτουργίας της κάθε μεθόδου.

5.3.2.1 Αποτελέσματα F1, Precision, Recall

Παρακάτω, στον *Πίνακα 7* και στην *Εικόνα 16*, παραθέτουμε τα αποτελέσματα των 3 μεθόδων στα πειράματα σε εκτεταμένο μέρος του Dataset:

ΕΠΙΠΕΔΟ	ΜΕΘΟΔΟΙ	EM	F1	PRECISION	RECALL
GENERATOR	Naive	0.43	0.551	0.572	0.559
	Raptor	0.3	0.412	0.439	0.431
	KG-Rag	**0.7**	**0.757**	**0.8**	**0.74**
RETRIEVER	Naive	-	0.113	0.064	0. 562
	Raptor	-	0.305	0.189	0.79
	KG-Rag	-	**0. 354**	**0.227**	**0.877**
SYSTEM	ει	-	0.076	0.043	0. 392
	Raptor	-	0.164	0.105	0.409
	KG-Rag	-	**0. 277**	**0.182**	**0.65**

Πίνακας 7 - Αποτελέσματα κλασσικών metrics



Εικόνα 16 - Σχηματική αναπαράσταση των metrics F1, Precision και Recall

Σύμφωνα με τα παραπάνω αποτελέσματα παρατηρούμε διάφορα ενδιαφέροντα πράγματα:

5.3.2.1.1 Ως προς την παραγωγή απαντήσεων:

Η πρώτη ξεκάθαρη και αδιαμφισβήτητη παρατήρηση είναι ότι η μέθοδος **KG-RAG** υπερέχει σημαντικά στην παραγωγή απαντήσεων, με **F1 = 0.76**, **precision = 0.80** και **recall = 0.74**. Αυτό δείχνει ότι οι απαντήσεις που παράγει είναι ταυτόχρονα ακριβείς (λιγότερος θόρυβος) και πλήρεις (περιλαμβάνουν το μεγαλύτερο μέρος της ζητούμενης πληροφορίας). Η μέθοδος **Naive** αποδίδει μέτρια (F1 = 0.55), με ισορροπία ανάμεσα σε precision και recall, αλλά με αρκετά χαμηλότερη ποιότητα απαντήσεων σε σχέση με την KG-RAG. Αντίθετα, η **RAPTOR** εμφανίζει τη χαμηλότερη επίδοση (F1 = 0.41), γεγονός που δείχνει ότι, παρά την ιεραρχική δόμηση του corpus, δυσκολεύεται να συνθέσει σωστές απαντήσεις.

5.3.2.1.2 Ως προς την ανάκτηση σχετικών τεκμηρίων:

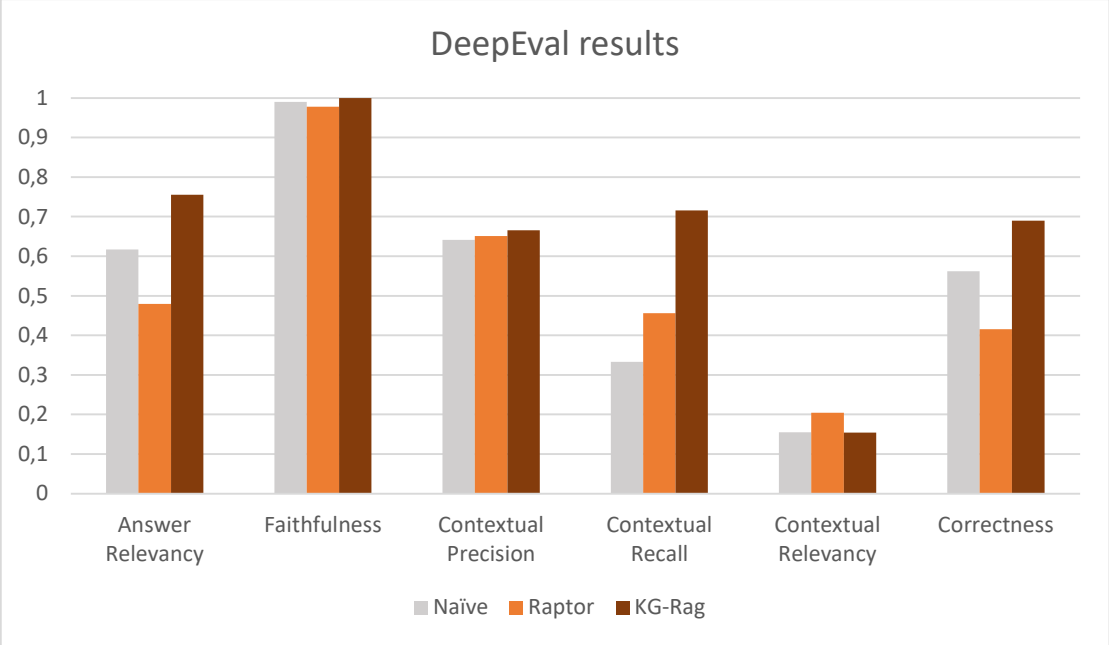
Στις μετρικές υποστηρικτικών τεκμηρίων παρατηρείται ότι το **recall** είναι υψηλό για τη **RAPTOR (0.79)** και ακόμη υψηλότερο για την **KG-RAG (0.88)**, γεγονός που σημαίνει ότι οι μέθοδοι αυτές ανακτούν σχεδόν όλα τα απαραίτητα τεκμήρια για την απάντηση. Ωστόσο, το **precision** είναι χαμηλό σε όλες τις περιπτώσεις, με την Naive να εμφανίζει την πιο αδύναμη απόδοση (0.06) και την KG-RAG να υπερέχει ελαφρώς (0.23). Αυτό δείχνει ότι, ενώ οι retrievers φέρνουν πολλά σχετικά τεκμήρια, ταυτόχρονα συνοδεύονται από σημαντικό ποσοστό άσχετης πληροφορίας, κάτι που μειώνει την καθαρότητα του context.

5.3.2.1.3 Ως προς όλο το σύστημα:

Στις joint μετρικές, οι οποίες μετρούν την ταυτόχρονη ορθότητα της απάντησης και των υποστηρικτικών τεκμηρίων, παρατηρούμε μια κλιμακωτή βελτίωση της απόδοσης από μέθοδο σε μέθοδο. Νικήτρια και πάλι στέφεται η KG-RAG μέθοδος, η οποία καταφέρνει πιο συχνά να δώσει όχι μόνο τη σωστή απάντηση αλλά και να ανακτήσει τα κατάλληλα supporting facts για να την τεκμηριώσει.

5.3.2.2 Αποτελέσματα μετρικών του DeepEval

Παρακάτω στην *Εικόνα 17* ακολουθούν τα αποτελέσματα των μετρικών του DeepEval:



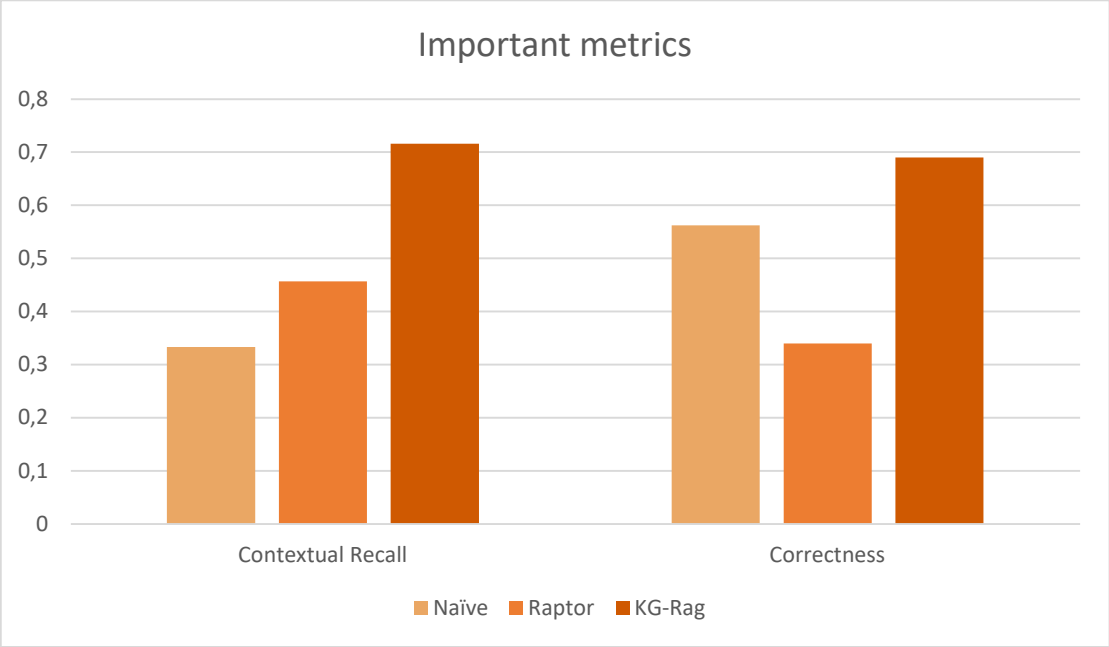
Εικόνα 17 - Αποτελέσματα των DeepEval metrics

Αντίστοιχα με την ανάλυση που πραγματοποιήσαμε παραπάνω για τις μετρικές του DeepEval μπορούμε να παρατηρήσουμε τα εξής:

- Όλες οι μέθοδοι παρουσιάζουν πολύ καλά σκορ στις Answer Relevancy και Faithfulness, γεγονός που δείχνει την καλή ποιότητα των Generator.
- Σχεδόν σε όλες τις μετρικές η KG-RAG δείχνει τα καλύτερα αποτελέσματα τόσο στην ανάκτηση πληροφοριών όσο και στην παραγωγή απαντήσεων.
- Η μόνη μετρική στην οποία δεν παρουσιάζει τόσο καλά αποτελέσματα σε σχέση με τις άλλες δύο μεθόδους είναι η Contextual Relevancy. Η μετρική αυτή δείχνει πόσο σημασιολογικά συναφή είναι το ανακτημένο context με την ερώτηση που τέθηκε από το χρήστη. Η χαμηλότερη επίδοση εδώ πέρα δεν είναι κάτι ανησυχητικό, αντιθέτως είναι αναμενόμενη αν αναλογιστούμε την τεχνική της KG-RAG. Βασικό βήμα αυτής

της τεχνικής είναι η επέκταση των σημασιολογικά κοντινών πληροφοριών με βάση το γράφο γνώσης με τις επιπλέον κοντινές τριπλέτες που μπορούν να εξαχθούν. Το συνολικό επεκταμένο context μπορεί να μην είναι τόσο σημασιολογικά συναφές με την ερώτηση, είναι όμως το κλειδί από ότι φαίνεται για να βρεθούν όχι εμφανείς αλλά χρήσιμες πληροφορίων για τη δημιουργία της απάντησης, γεγονός που αποδεικνύεται και από την καλύτερη απόδοση στην παραγωγή σωστών απαντήσεων.

- Γενικά βέβαια το Contextual Relevancy είναι χαμηλό και θα μπορούσε να αποτελέσει αντικείμενο μελέτης στο μέλλον, ώστε να μειωθούν οι μη σχετικές πληροφορίες που ανακτώνται και τελικά ίσως να είχαμε και καλύτερη συνολική απόδοση.



Εικόνα 18 - Αποτελέσματα Contextual Recall και Correctness

Οι πιο σημαντικές μετρικές οι οποίες μας προσφέρουν τις πιο πολύτιμες πληροφορίες είναι οι **Correctness** κι η **Contextual Recall**, οι οποίες απεικονίζονται στην *Εικόνα 18***.** Η Correctness, όπως έχουμε αναφέρει και παραπάνω, μας δείχνει κατά πόσο η απάντηση που παρήγαγε το σύστημά μας (actual output) είναι όντως η αναμενόμενη (expected output), το οποίο είναι και το ζητούμενο κι εκείνο που τελικά είναι ο σκοπός μας. Παρατηρούμε λοιπόν ότι το σύστημα που ανταποκρίνεται καλύτερα και έχει τις περισσότερες σωστές απαντήσεις είναι για άλλη μια φορά το KG-RAG. Ύστερα ακολουθεί η απλούστερη μέθοδος, η Naive, και τελευταία είναι η τεχνική Raptor.

Η Contextual Recall μας δείχνει την ικανότητα του συστήματος να ανακτά τις κατάλληλες πληροφορίες για την απάντηση της ερώτησης που θέτουμε. Εδώ παρατηρούμε μια κλιμακωτή αύξηση της απόδοσης από μέθοδο σε μέθοδο με την Naive να είναι η χειρότερη στην ανάκτηση των κατάλληλων πληροφοριών, η Raptor να είναι καλύτερη και η KG-RAG ακόμα καλύτερη.

Εδώ, λοιπόν, παρατηρείται ένα **παράδοξο** αφού η Raptor, η οποία είναι πιο σύνθετη από τη Naive, και αναμέναμε να συμπεριφέρεται καλύτερα, δεν καταφέρνει να παράγει αρκετές σωστές απαντήσεις

$$Correctness_{Rapto} < Correctness_{Naive}$$

ενώ βλέπουμε ότι οι πληροφορίες που ανακτά είναι καλύτερες από εκείνες που ανακτά η Naive.

$$Contectual\ Recall_{Rap} > Contectual\ Recall_{Naive}$$

Αυτό σημαίνει ότι ο Generator του Raptor δεν ανταποκρίνεται τόσο καλά. Παρακάτω παραθέτουμε ένα παράδειγμα μιας ερώτησης και της ανταπόκρισης του Naive και του Raptor.

Ερώτηση (input)	*What science fantasy young adult series, told in first person, has a set of companion books narrating the stories of enslaved worlds and alien species?*
Αναμενόμενη απάντηση	*Animorphs*
Απάντηση Naive	*Animorphs*
Απάντηση Raptor	*I don't have that information.*
Ανακτημένο κείμενο Naive (retrieved context)	*"Science Fantasy" ceased publication the following year, when Roberts & Vinter came under financial pressure after their printer went bankrupt. Animorphs is a science fantasy series of young adult books written by Katherine Applegate and her husband Michael Grant, writing together under the name K. A. Applegate, and published by Scholastic. It is told in first person, with all six main characters taking turns narrating the books through their own perspectives. Horror, war, dehumanization,* *"Left Behind: The Kids (stylized as LEFT BEHIND >THE KIDS<)" is a series written by Jerry B. Jenkins, Tim LaHaye, and Chris Fabry. The series consists of 40 short novels aimed primarily at the young adult market based on the adult series Left Behind also written by Jerry B. Jenkins. It follows a core group of teenagers as they experience the rapture and*

	tribulation, based on scriptures found in the Bible, and background plots introduced in the adult novels. Like the adult series, the books own perspectives. Horror, war, dehumanization, sanity, morality, innocence, leadership, freedom and growing up are the core themes of the series. The series has sold over 11 million copies worldwide. Dozens of Square Enix companion books have been produced since 1998, when video game developer Square began to produce books that focused on artwork, developer interviews, and background information on the fictional worlds and characters in its games rather than on gameplay details. The first series of these books was the "Perfect Works" series, written and published by Square subsidiary DigiCube. They produced three books between 1998 and "Wild Ink: A Grownups Guide to Writing Fiction for Teens".
Ανακτημένο κείμενο Raptor (retrieved context)	*Animorphs* *I don't have information about a "science fantasy young adult series" that is told in first-person with a set of companion books narrating the stories of enslaved worlds and alien species.* *I don't have information about a "science fantasy young adult series" that fits your description.* *The context provided does not contain information about a science fantasy young adult series told in first person with a set of companion books narrating the stories of enslaved worlds and alien species.*

Πίνακας 8 - Παράδειγμα συμπεριφοράς της Naive και της Raptor όταν τους τέθηκε η ίδια ερώτηση (με μπλε είναι σημειωμένη η πληροφορία που ανέκτησαν οι retrievers των δύο μεθόδων και χρειαζόταν για την απάντηση της ερώτησης)

Στο παραπάνω παράδειγμα του *Πίνακα 8* φαίνεται ότι η Raptor απαντάει και λάθος και λανθασμένα. Απαντάει λάθος απάντηση, όχι “Animorphs”, και επίσης απαντάει λανθασμένα ότι «Δεν έχει αυτή την πληροφορία» παρά το γεγονός ότι στο ανακτημένο κείμενο υπάρχει η κατάλληλη πληροφορία. Στο ανακτημένο κείμενο όμως εκτός από την πολύτιμη για την απάντηση πληροφορία υπάρχουν άλλες 3 δηλώσεις ότι «Δεν έχω αυτή την πληροφορία»

γεγονός που μπερδεύει τον Generator και τον οδηγεί στο να παράγει λάθος απαντήσεις. Για αυτό το γεγονός ευθύνονται δύο πράγματα:

1. Ο Contextual Compression Retriever της Raptor

Κατά τη διάρκεια της εκτέλεσης του Raptor pipeline, στα στάδιο της ανάκτησης των τεκμηρίων υπάρχουν δύο βήματα. Αρχικά, με έναν βασικό retriever, βρίσκουμε τα πιο κοντινά έγγραφα από το FAISS index με βάση την ομοιότητα των embeddings. Στη συνέχεια αυτά τα «πιο όμοια» κείμενα περνάνε από τον contextual compression retriever ώστε να συμπιεστούν. Στον contextual compression retriever έχουμε ορίσει το εξής prompt:

"Given the following context and question, extract only the relevant information for answering the question:"

Δηλαδή λέμε στο LLM *«Πάρε το context και την ερώτηση, και κράτησε μόνο τις πληροφορίες που είναι σχετικές για την απάντηση».* Έτσι, αντί να περάσει όλο το context, φτιάχνεται ένα φίλτρο που εξάγει μόνο τα χρήσιμα σημεία. Το αποτέλεσμα είναι ότι από τα κείμενα που δεν περιέχουν σχετική με την ερώτηση πληροφορία παίρνουμε σαν context το ότι «Δεν έχω κάποια πληροφορία για αυτή την ερώτηση» γεγονός που οδηγεί πολλές φορές τον Generator σε αποτυχία.

2. Η Raptor μέθοδος δεν ενδείκνυται για το συγκεκριμένο dataset

Το φαινόμενο της πτώσης της απόδοσης κατά την εφαρμογή της Raptor στο dataset HotPotQA μπορεί να εξηγηθεί τόσο από τη φύση του ίδιου του dataset όσο και από τη σχεδιαστική φιλοσοφία του RAPTOR. Συγκεκριμένα, το HotPotQA αποτελείται από σχετικά σύντομα αποσπάσματα κειμένου, στα οποία κάθε πρόταση περιέχει μία αυτόνομη και πλήρη πληροφορία. Σε τέτοια σενάρια, η λογική του **clustering** και της **αφαιρετικής περίληψης (abstractive summarization)** που εφαρμόζει η RAPTOR δεν προσφέρει ουσιαστικό πλεονέκτημα, καθώς δεν υπάρχει μεγάλος όγκος κειμένου που να χρειάζεται συμπύκνωση. Αντιθέτως, η διαδικασία περίληψης μπορεί να οδηγήσει σε **απώλεια κρίσιμων λεπτομερειών** ή και σε **παραμόρφωση** της πληροφορίας, με αποτέλεσμα χαμηλότερη ποιότητα απαντήσεων.

Όπως επισημαίνεται και στο RAPTOR paper [35], η μέθοδος είναι ιδιαίτερα αποτελεσματική όταν η ζητούμενη πληροφορία εκτείνεται σε **εκτενή κείμενα** και

απαιτείται ιεραρχική οργάνωση μέσω clustering και summarization ώστε να αναδειχθεί το πιο χρήσιμο περιεχόμενο.

Συνεπώς, η χαμηλή απόδοση του RAPTOR στο HotPotQA δεν οφείλεται σε αδυναμία της μεθόδου καθαυτής, αλλά στην **ανακολουθία ανάμεσα στη φύση του dataset** και στο είδος προβλήματος για το οποίο σχεδιάστηκε. Η συγκεκριμένη μέθοδος παραμένει χρήσιμη σε περιβάλλοντα με εκτενή κείμενα (π.χ. επιστημονικά άρθρα, reports, βιβλία), αλλά λιγότερο κατάλληλη για benchmarks με μικρά, αυτοτελή αποσπάσματα όπως το HotPotQA.

5.3.2.3 Αποτελέσματα μετρήσεων χρονικής απόκρισης

Μέθοδος	*Χρόνος (s)*
Naïve	17
Raptor	273,3
KG-RAG	~3000

Πίνακας 9 - Χρόνος δημιουργίας βάσης γνώσης

Μέθοδος	*Χρόνος (s)*
Naïve	0,565
Raptor	6,091
KG-RAG	13

Πίνακας 10 - Μέσος χρόνος δημιουργίας απάντησης

Στον *Πίνακα 9* παρουσιάζεται ο χρόνος που απαιτήθηκε για τη δημιουργία της βάσης γνώσης σε κάθε μία από τις τρεις μεθόδους. Παρατηρούμε ότι η Naive μέθοδος ολοκληρώνει τη διαδικασία πολύ γρήγορα (17s), καθώς περιορίζεται σε μία απλή μετατροπή του dataset σε συλλογή προτάσεων χωρίς περαιτέρω επεξεργασία. Αντίθετα, ο RAPTOR εμφανίζει σαφώς μεγαλύτερο χρόνο (273,3s), γεγονός που οφείλεται στο επιπλέον στάδιο clustering και περίληψης μέσω LLM, τα οποία εισάγουν σημαντική υπολογιστική επιβάρυνση. Η πιο απαιτητική μέθοδος είναι η KG-RAG, η οποία χρειάστηκε περίπου 3000s .Αυτό εξηγείται από το γεγονός ότι η διαδικασία περιλαμβάνει εξαγωγή τριπλετών (triplets) από το κάθε context με τη χρήση LLM, κάτι που είναι ιδιαίτερα κοστοβόρο χρονικά και υπολογιστικά.

Στον *Πίνακα 10* παρουσιάζεται ο μέσος χρόνος απόκρισης για τη δημιουργία απάντησης ανά ερώτηση. Η Naive μέθοδος είναι ξανά η ταχύτερη (0,565s), καθώς ανακτά και περνά στο LLM το context χωρίς κάποια ιδιαίτερη επεξεργασία. Ο RAPTOR αυξάνει σημαντικά τον χρόνο (6,091s), αφού εκτελεί το επιπλέον στάδιο contextual compression. Ο KG-RAG εμφανίζει τον υψηλότερο χρόνο (13s), καθώς εκτός από την ανάκτηση context, απαιτείται και χρήση του knowledge graph με τριπλέτες, το οποίο προσθέτει επιπλέον latency στη διαδικασία παραγωγής απάντησης.

Γενικότερα, η ανάλυση δείχνει ένα σαφές **trade-off ανάμεσα στην ακρίβεια και στον χρόνο απόκρισης**. Η Naive μέθοδος είναι η πιο γρήγορη αλλά με χαμηλότερη ποιότητα απαντήσεων. Ο RAPTOR και κυρίως ο KG-RAG απαιτούν πολλαπλάσιο χρόνο, όμως προσφέρουν υψηλότερη ακρίβεια και καλύτερη τεκμηρίωση. Ειδικά στην περίπτωση του KG-RAG, το υπολογιστικό κόστος είναι πολύ υψηλό και καθιστά δύσκολη την εφαρμογή του σε μεγάλης κλίμακας συστήματα σε πραγματικό χρόνο. Ωστόσο, μπορεί να αποδειχθεί ιδιαίτερα χρήσιμο σε σενάρια όπου η ποιότητα και η πιστότητα της απάντησης είναι πιο κρίσιμες από την ταχύτητα.

6

Συμπεράσματα και Μελλοντική Εργασία

6.1 Συμπεράσματα

Η παρούσα εργασία είχε ως στόχο την υλοποίηση, αξιολόγηση και συγκριτική μελέτη τριών διαφορετικών προσεγγίσεων **Retrieval-Augmented Generation (RAG)**: της **Naive RAG**, της **RAPTOR RAG** και της **KG-RAG**. Μέσα από τη μεθοδολογική ανάλυση και την πειραματική αξιολόγηση στο dataset **HotpotQA** (Distractor set) αναδείχθηκαν σημαντικά συμπεράσματα για τα πλεονεκτήματα και τους περιορισμούς κάθε μεθόδου.

Η μέθοδος **Naive RAG** αποδείχθηκε η απλούστερη και ταχύτερη, με ελάχιστο υπολογιστικό κόστος τόσο κατά την κατασκευή της βάσης γνώσης όσο και κατά την απόκριση σε ερωτήσεις. Ωστόσο, η απλότητα αυτή συνοδεύτηκε από περιορισμένη ακρίβεια και υψηλά επίπεδα θορύβου στο ανακτηθέν context, γεγονός που μείωσε την ποιότητα των απαντήσεων.

Η μέθοδος **RAPTOR** εισήγαγε μια πιο σύνθετη διαδικασία οργάνωσης των δεδομένων μέσω clustering και αφαιρετικής περίληψης. Αν και η λογική αυτή είναι ιδιαίτερα χρήσιμη σε περιπτώσεις μεγάλων και εκτενών κειμένων, τα αποτελέσματα στο HotpotQA δεν ήταν τόσο καλά. Η φύση του dataset (σύντομα αποσπάσματα με αυτόνομες πληροφορίες) δεν ευνοεί τη συμπύκνωση, με αποτέλεσμα απώλεια κρίσιμων λεπτομερειών και χαμηλή απόδοση στις τελικές απαντήσεις.

Η μέθοδος **KG-RAG** κατέγραψε την καλύτερη συνολική απόδοση, με υψηλότερες τιμές στις μετρικές F1, precision και recall, καθώς και καλύτερη ποιότητα στα υποστηρικτικά τεκμήρια. Η χρήση γραφο-δομημένης γνώσης προσέφερε μεγαλύτερη ακρίβεια και πιο πιστές απαντήσεις. Ωστόσο, το σημαντικό μειονέκτημα ήταν το υπέρογκο υπολογιστικό κόστος και ο μεγάλος χρόνος απόκρισης, που καθιστούν δύσκολη την άμεση εφαρμογή της σε σενάρια πραγματικού χρόνου.

Συνολικά, τα αποτελέσματα καταδεικνύουν το **trade-off ανάμεσα στην ποιότητα και στον χρόνο απόκρισης**: οι πιο σύνθετες μέθοδοι (RAPTOR, KG-RAG) προσφέρουν υψηλότερη

ακρίβεια, αλλά απαιτούν πολλαπλάσιο χρόνο και πόρους. Αντίθετα, η Naive μέθοδος είναι εξαιρετικά ταχεία αλλά θυσιάζει σε ποιότητα.

6.2 Περιορισμοί της παρούσας εργασίας

Η εργασία αυτή εστιάστηκε σε ένα μόνο dataset (HotpotQA Distractor set), το οποίο, παρότι αποτελεί καθιερωμένο benchmark για multi-hop reasoning, δεν αντικατοπτρίζει πλήρως πιο σύνθετα ή domain-specific σενάρια.

Ένας δεύτερος περιορισμός αφορά το **υπολογιστικό κόστος**. Ειδικά στην περίπτωση της KG-RAG, η διαδικασία εξαγωγής τριπλετών και η χρήση LLM σε μεγάλο όγκο δεδομένων αποδείχθηκε εξαιρετικά χρονοβόρα, περιορίζοντας την πρακτικότητα της μεθόδου.

Επιπλέον, οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση περιλαμβάνουν και **LLM-as-a-judge** προσεγγίσεις μέσω του DeepEval. Παρότι αυτές προσφέρουν λεπτομερή εικόνα για τη faithfulness και relevancy, εισάγουν τον κίνδυνο μεροληψίας, καθώς βασίζονται σε κρίσεις άλλων LLMs.

6.3 Μελλοντική Εργασία

Μελλοντικές κατευθύνσεις για περαιτέρω έρευνα και βελτίωση περιλαμβάνουν:

- **Βελτίωση αποδοτικότητας**: Ανάπτυξη πιο αποδοτικών τεχνικών για triplet extraction και summarization (π.χ. χρήση μικρότερων εξειδικευμένων LLMs ή rule-based extractors) με στόχο τη μείωση χρόνου και κόστους.
- **Επέκταση πειραμάτων**: Δοκιμή σε διαφορετικά datasets (Natural Questions, TriviaQA, domain-specific corpora) ώστε να διερευνηθεί η συμπεριφορά των μεθόδων σε διαφορετικά είδη ερωτήσεων και context.
- **Υβριδικές προσεγγίσεις**: Συνδυασμός RAG με fine-tuned LLMs για συγκεκριμένους τομείς γνώσης
- **Explainability και εμπιστοσύνη**: Ανάπτυξη διεπαφών που εμφανίζουν στον χρήστη όχι μόνο την απάντηση αλλά και τα supporting facts ή τις διαδρομές στο knowledge graph που την τεκμηριώνουν, ενισχύοντας την ερμηνευσιμότητα και τη διαφάνεια.
- **Βελτίωση retriever filtering**: Ενσωμάτωση τεχνικών reranking (π.χ. με cross-encoders) ή contrastive learning ώστε να βελτιωθεί η contextual precision και η contextual relevancy και να μειωθεί ο θόρυβος.

- **Επέκταση σε πολυτροπικά δεδομένα (multi-modal RAG)**: Διερεύνηση της εφαρμογής των μεθόδων σε δεδομένα που συνδυάζουν κείμενο με εικόνες, πίνακες ή άλλες μορφές δομημένης πληροφορίας.

7 Βιβλιογραφία

[1] Q. Zhang *et al.*, "A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models," Jan. 21, 2025, *arXiv*: arXiv:2501.13958. doi: 10.48550/arXiv.2501.13958.

[2] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying Large Language Models and Knowledge Graphs: A Roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, July 2024, doi: 10.1109/TKDE.2024.3352100.

[3] R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg, "DENDRAL: A case study of the first expert system for scientific hypothesis formation," *Artif. Intell.*, vol. 61, no. 2, pp. 209–261, June 1993, doi: 10.1016/0004-3702(93)90068-M.

[4] E. H. Shortliffe, "A rule-based computer program for advising physicians regarding antimicrobial therapy selection," in *Proceedings of the 1974 annual conference on XX - ACM '74*, Not Known: ACM Press, 1974, p. 739. doi: 10.1145/1408800.1408906.

[5] T. Teubner, C. M. Flath, C. Weinhardt, W. van der Aalst, and O. Hinz, "Welcome to the Era of ChatGPT et al.," *Bus. Inf. Syst. Eng.*, vol. 65, no. 2, pp. 95–101, Apr. 2023, doi: 10.1007/s12599-023-00795-x.

[6] "(PDF) Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.," *ResearchGate*, Aug. 2025, Accessed: Oct. 06, 2025. [Online]. Available: https://www.researchgate.net/publication/220546096_Judea_Pearl_Probabilistic_Reasoning_in_Intelligent_Systems_Networks_of_Plausible_Inference

[7] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition and Applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022, doi: 10.1109/TNNLS.2021.3070843.

[8] X. Li *et al.*, "From Matching to Generation: A Survey on Generative Information Retrieval," Mar. 04, 2025, *arXiv*: arXiv:2404.14851. doi: 10.48550/arXiv.2404.14851.

[9] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, "Challenges and Applications of Large Language Models," July 19, 2023, *arXiv*: arXiv:2307.10169. doi: 10.48550/arXiv.2307.10169.

[10] J. Li *et al.*, "Fundamental Capabilities of Large Language Models and their Applications in Domain Scenarios: A Survey," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Dec. 2024, pp. 11116–11141. doi: 10.18653/v1/2024.acl-long.599.

[11] A. Vaswani *et al.*, "Attention Is All You Need," Aug. 02, 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[12] W. Fan *et al.*, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," June 17, 2024, *arXiv*: arXiv:2405.06211. doi: 10.48550/arXiv.2405.06211.

[13] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," Mar. 27, 2024, *arXiv*: arXiv:2312.10997. doi: 10.48550/arXiv.2312.10997.

[14] Z. Jing *et al.*, "When Large Language Models Meet Vector Databases: A Survey," June 23, 2025, *arXiv*: arXiv:2402.01763. doi: 10.48550/arXiv.2402.01763.

[15] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," July 22, 2020, *arXiv*: arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165.

[16] T. Munkhdalai, M. Faruqui, and S. Gopal, “Leave No Context Behind: Efficient Infinite Context Transformers with Infini-attention,” Aug. 09, 2024, *arXiv*: arXiv:2404.07143. doi: 10.48550/arXiv.2404.07143.
[17] S. Chen, S. Wong, L. Chen, and Y. Tian, “Extending Context Window of Large Language Models via Positional Interpolation,” June 28, 2023, *arXiv*: arXiv:2306.15595. doi: 10.48550/arXiv.2306.15595.
[18] Z. Wang, Z. Chu, T. V. Doan, S. Ni, M. Yang, and W. Zhang, “History, Development, and Principles of Large Language Models-An Introductory Survey,” Sept. 23, 2024, *arXiv*: arXiv:2402.06853. doi: 10.48550/arXiv.2402.06853.
[19] Z. Xu, S. Jain, and M. Kankanhalli, “Hallucination is Inevitable: An Innate Limitation of Large Language Models,” Feb. 13, 2025, *arXiv*: arXiv:2401.11817. doi: 10.48550/arXiv.2401.11817.
[20] H. Zhao *et al.*, “Explainability for Large Language Models: A Survey,” Nov. 28, 2023, *arXiv*: arXiv:2309.01029. doi: 10.48550/arXiv.2309.01029.
[21] G. Mentzas, M. Fikardos, K. Lepenioti, and D. Apostolou, “Exploring the landscape of trustworthy artificial intelligence: Status and challenges,” *Intell. Decis. Technol.*, vol. 18, no. 2, pp. 837–854, May 2024, doi: 10.3233/IDT-240366.
[22] N. F. Liu *et al.*, “Lost in the Middle: How Language Models Use Long Contexts,” Nov. 20, 2023, *arXiv*: arXiv:2307.03172. doi: 10.48550/arXiv.2307.03172.
[23] J. Boye and B. Moell, “Large Language Models and Mathematical Reasoning Failures,” Feb. 21, 2025, *arXiv*: arXiv:2502.11574. doi: 10.48550/arXiv.2502.11574.
[24] X. Li, W. Wang, M. Li, J. Guo, Y. Zhang, and F. Feng, “Evaluating Mathematical Reasoning of Large Language Models: A Focus on Error Identification and Correction,” June 02, 2024, *arXiv*: arXiv:2406.00755. doi: 10.48550/arXiv.2406.00755.
[25] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *ArXiv*, May 2020, Accessed: Oct. 07, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Retrieval-Augmented-Generation-for-NLP-Tasks-Lewis-Perez/659bf9ce7175e1ec266ff54359e2bd76e0b7ff31
[26] S. Shrestha, M. Kim, and K. Ross, “Mathematical Reasoning in Large Language Models: Assessing Logical and Arithmetic Errors across Wide Numerical Ranges,” Feb. 12, 2025, *arXiv*: arXiv:2502.08680. doi: 10.48550/arXiv.2502.08680.
[27] D. Edge *et al.*, “From Local to Global: A Graph RAG Approach to Query-Focused Summarization,” Feb. 19, 2025, *arXiv*: arXiv:2404.16130. doi: 10.48550/arXiv.2404.16130.
[28] R. Chen, “Retrieval-Augmented Generation with Knowledge Graphs: A Survey,” presented at the Computer Science Undergradaute Conference 2025 @ XJTU, Mar. 2025. Accessed: July 23, 2025. [Online]. Available: https://openreview.net/forum?id=ZikTuGY28C
[29] X. Wang *et al.*, “Searching for Best Practices in Retrieval-Augmented Generation,” July 01, 2024, *arXiv*: arXiv:2407.01219. doi: 10.48550/arXiv.2407.01219.
[30] Y. Gao, Y. Xiong, M. Wang, and H. Wang, “Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks,” July 26, 2024, *arXiv*: arXiv:2407.21059. doi: 10.48550/arXiv.2407.21059.
[31] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, “Seven Failure Points When Engineering a Retrieval Augmented Generation System,” Jan. 11, 2024, *arXiv*: arXiv:2401.05856. doi: 10.48550/arXiv.2401.05856.
[32] C. Niu *et al.*, “RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models,” May 17, 2024, *arXiv*: arXiv:2401.00396. doi: 10.48550/arXiv.2401.00396.
[33] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, “Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely,” Sept. 23, 2024, *arXiv*: arXiv:2409.14924. doi: 10.48550/arXiv.2409.14924.

[34] "RAG systems have a recall problem, not a hallucination one," Bogdan Buduroiu. Accessed: Oct. 07, 2025. [Online]. Available: https://buduroiu.com/blog/rag-llm-recall-problem/
[35] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval," Jan. 31, 2024, *arXiv*: arXiv:2401.18059. doi: 10.48550/arXiv.2401.18059.
[36] N. Choudhary and C. K. Reddy, "Complex Logical Reasoning over Knowledge Graphs using Large Language Models," Mar. 31, 2024, *arXiv*: arXiv:2305.01157. doi: 10.48550/arXiv.2305.01157.
[37] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering," Dec. 13, 2022, *arXiv*: arXiv:2104.06378. doi: 10.48550/arXiv.2104.06378.
[38] X. Zhu, Y. Xie, Y. Liu, Y. Li, and W. Hu, "Knowledge Graph-Guided Retrieval Augmented Generation," Feb. 08, 2025, *arXiv*: arXiv:2502.06864. doi: 10.48550/arXiv.2502.06864.
[39] D. Wu, Y. Yan, Z. Liu, Z. Liu, and M. Sun, "KG-Infused RAG: Augmenting Corpus-Based RAG with External Knowledge Graphs," June 11, 2025, *arXiv*: arXiv:2506.09542. doi: 10.48550/arXiv.2506.09542.
[40] W. Zhang, J. Zhang, W. Zhang, and J. Zhang, "Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review," *Mathematics*, vol. 13, no. 5, Mar. 2025, doi: 10.3390/math13050856.
[41] P. Béchard and O. M. Ayala, "Reducing hallucination in structured outputs via Retrieval-Augmented Generation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, 2024, pp. 228–238. doi: 10.18653/v1/2024.naacl-industry.19.
[42] M. Cheng *et al.*, "A Survey on Knowledge-Oriented Retrieval-Augmented Generation," Mar. 17, 2025, *arXiv*: arXiv:2503.10677. doi: 10.48550/arXiv.2503.10677.
[43] N. Pipitone and G. H. Alami, "LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain," Aug. 19, 2024, *arXiv*: arXiv:2408.10343. doi: 10.48550/arXiv.2408.10343.
[44] "Leveraging RAG For Domain-Specific Knowledge Retrieval And Generation." Accessed: Oct. 07, 2025. [Online]. Available: https://www.protecto.ai/blog/leveraging-rag-for-domain-specific-knowledge-retrieval-and-generation/
[45] T. Şakar and H. Emekci, "Maximizing RAG efficiency: A comparative analysis of RAG methods," *Nat. Lang. Process.*, vol. 31, no. 1, pp. 1–25, Jan. 2025, doi: 10.1017/nlp.2024.53.
[46] F. Cuconasu *et al.*, "The Power of Noise: Redefining Retrieval for RAG Systems," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2024, pp. 719–729. doi: 10.1145/3626772.3657834.
[47] Y. Tang and Y. Yang, "MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries," Jan. 27, 2024, *arXiv*: arXiv:2401.15391. doi: 10.48550/arXiv.2401.15391.
[48] NirDiamant, *NirDiamant/Controllable-RAG-Agent*. (Oct. 08, 2025). Jupyter Notebook. Accessed: Oct. 08, 2025. [Online]. Available: https://github.com/NirDiamant/Controllable-RAG-Agent
[49] "Advanced RAG Techniques: What They Are & How to Use Them." Accessed: Oct. 08, 2025. [Online]. Available: https://falkordb.com/blog/advanced-rag/
[50] F. Khan, "Optimizing RAG with the Smarter Indexing RAPTOR Pipeline," Medium. Accessed: Oct. 07, 2025. [Online]. Available: https://levelup.gitconnected.com/optimizing-rag-with-the-smarter-indexing-raptor-pipeline-65bb946526ad
[51] H. Q. Yu and F. McQuade, "RAG-KG-IL: A Multi-Agent Hybrid Framework for Reducing Hallucinations and Enhancing LLM Reasoning through RAG and Incremental Knowledge

Graph Learning Integration," Mar. 14, 2025, *arXiv*: arXiv:2503.13514. doi: 10.48550/arXiv.2503.13514.
[52] X. Zhu, Y. Xie, Y. Liu, Y. Li, and W. Hu, "Knowledge Graph-Guided Retrieval Augmented Generation," Feb. 08, 2025, *arXiv*: arXiv:2502.06864. doi: 10.48550/arXiv.2502.06864.
[53] Z. Yang *et al.*, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," Sept. 25, 2018, *arXiv*: arXiv:1809.09600. doi: 10.48550/arXiv.1809.09600.