NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

MSc THESIS

Evaluating the capabilities of LLMs in geospatial question answering and geospatial reasoning

EVANGELOS EMMANOUIL KARAGIANNIS

Supervisor: **Manolis Koubarakis,** Professor

Co-Supervisor: **Konstantinos Plas,** Associate Researcher

ATHENS

JANUARY 2025

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αξιολογώντας την απόδοση των Μεγάλων Γλωσσικών Μοντέλων (LLMs) σε γεω-χωρικές ερωτήσεις και χωρική αντίληψη

ΕΥΑΓΓΕΛΟΣ ΕΜΜΑΝΟΥΗΛ ΚΑΡΑΓΙΑΝΝΗΣ

Επιβλέπων: **Μανόλης Κουμπαράκης,** Καθηγητής

Συνεπιβλέπων: **Κωνσταντίνος Πλας,** Συνεργαζόμενος Ερευνητής

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2025

MSc THESIS

Evaluating the capabilities of LLMs in geospatial question answering and geospatial reasoning

EVANGELOS EMMANOUIL KARAGIANNIS

S.N.: 7115132200002

SUPERVISOR: **Manolis Koubarakis,** Professor

COSUPERVISOR: **Konstantinos Plas,** Associate Researcher

THESIS COMMITTEE: **Name Surname,** Professor
Name Surname, Professor
Name Surname, Professor

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αξιολογώντας την απόδοση των Μεγάλων Γλωσσικών Μοντέλων (LLMs) σε γεω-χωρικές ερωτήσεις και χωρική αντίληψη

ΕΥΑΓΓΕΛΟΣ ΕΜΜΑΝΟΥΗΛ ΚΑΡΑΓΙΑΝΝΗΣ

Α.Μ.: 7115132200002

ΕΠΙΒΛΕΠΩΝ: **Μανόλης Κουμπαράκης,** Καθηγητής

ΣΥΝΕΠΙΒΛΕΠΩΝ: **Κωνσταντίνος Πλας,** Συνεργαζόμενος Ερευνητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Όνομα Επώνυμο,** Καθηγητής
Όνομα Επώνυμο, Καθηγητής
Όνομα Επώνυμο, Καθηγητής

ABSTRACT

In this thesis, we aim to evaluate LLMs in terms of their ability to correctly answer geospatial questions. For the purposes of the aforementioned evaluation, we used the GeoQuestions1089 benchmark, which includes geospatial questions along with their respective answers and SPARQL or GeoSPARQL queries. In the case where the answers are close to identical, we classify the LLM's answer as (mostly) correct. If the answers are not identical but share some common elements, we consider them partially correct. Conversely, if they share no common elements, the LLM's answer is classified as incorrect.

The second aspect of this assignment relates to the ability of LLMs to understand and process spatial information, as well as to deduce relationships between objects in two-dimensional space. More specifically, given certain graphs and providing the models with the spatial relations between some of their nodes as input, we investigate whether the models can infer other relationships.

After extracting the useful information from GeoQuestions1089 and removing the questions containing polygons, we categorize the remaining questions into Binary, Descriptive, and Quantitative types. This categorization is necessary because each question type requires a different evaluation methodology.

The results showed that the models perform well in correctly evaluating the first and third categories of questions, which are considered easier, but struggle with the second category. Specifically, large models are capable of addressing questions from the second category but not always successfully. Smaller models have greater difficulty with the second category of questions and are more inconsistent, meaning they may provide different answers to the same question.

SUBJECT AREA: Large Language Models

KEYWORDS: Deep Learning, SPARQL, GeoSPARQL, Knowledge Graphs, Large Language Models

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία επιχειρήθηκε η εξέταση διαφόρων Μεγάλων Γλωσσικών Μοντέλων αναφορικά με την ικανότητά τους να απαντούν σωστά ερωτήσεις γεωγραφικού και χωρικού χαρακτήρα (geospatial questions).

Για τα πλαίσια της παραπάνω αξιολόγησης χρησιμοποιήθηκε το dataset, GeoQuestions1089, που εμπεριέχει ερωτήσεις γεω-χωρικού χαρακτήρα, τις αντίστοιχες απαντήσεις μαζί με τις SPARQL ή GeoSPARQL εντολές. Στην περίπτωση που οι απαντήσεις ταυτίζονται ή είναι παρόμοιες, τότε η απάντηση που πήραμε από το LLM χαρακτηρίζεται ως (κυρίως) σωστή. Διαφορετικά, αν υπάρχουν ορισμένα κοινά σημεία, τότε η απάντηση του γλωσσικού μοντέλου θεωρείται μερικώς σωστή. Αντιθέτως, αν δεν υπάρχει σχεδόν κανένα κοινό σημείο μεταξύ των δύο στοιχείων, η απάντηση του LLM θα ερμηνευθεί ως λανθασμένη.

Η δεύτερη πτυχή της εργασίας αφορά την ικανότητα των γλωσσικών μοντέλων να αντιλαμβάνονται, επεξεργάζονται χωρική πληροφορία και να συμπεραίνουν τις σχέσεις αντικειμένων στον χώρο των δύο διαστάσεων. Συγκεκριμένα, δοθέντος κάποιων γράφων και δίνοντας στα μοντέλα την χωρική σχέση μεταξύ κάποιων από των κόμβων τους ως εντολή εισόδου, ρωτάμε το μοντέλο να συμπεράνει κάποιες άλλες σχέσεις.

Μετά την επιτυχημένη εξαγωγή της χρήσιμης πληροφορίας από τα περιεχόμενα του δείκτη GeoQuestions1089 και την διαγραφή των ερωτήσεων που έχουν ως απάντηση πολύγονα, χωρίσαμε όλες τις ερωτήσεις στις κατηγορίες Δυαδικές, Περιγραφικές και Ποσοτικές. Ο λόγος που έγινε αυτή η διαφοροποίηση είναι λόγο του ότι η κάθε κατηγορία ερωτήσεων αξιολογείται με διαφορετικό τρόπο.

Τα αποτελέσματα έδειξαν ότι τα μοντέλα είναι καλά στο να αξιολογούν σωστά την πρώτη και τρίτη κατηγορία ερωτήσεων που θεωρούνται πιο εύκολες αλλά δυσκολεύονται στην δεύτερη. Συγκεκριμένα, τα μεγάλα μοντέλα μπορούν να βρουν και ερωτήσεις της δεύτερης κατηγορίας αλλά όχι πάντα με επιτυχία. Τα μικρά μοντέλα δυσκολεύονται περισσότερο στην δεύτερη κατηγορία των ερωτήσεων και είναι πιο ασταθή, δηλαδή μπορούν να απαντήσουν διαφορετικά στην ίδια ερώτηση.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μεγάλα Γλωσσικά Μοντέλα

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Βαθιά Εκμάθηση , SPARQL, GeoSPARQL, Γραφήματα Γνώσης, Μεγάλα Γλωσσικά Μοντέλα

Στη σελίδα αυτή αναφέρονται οι αφιερώσεις. Η σελίδα αυτή είναι προαιρετική.

ACKNOWLEDGEMENTS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. ΕΙΣΑΓΩΓΗ

1.1 Περιγραφή Προβλήματος

Η ραγδαία τεχνολογική εξέλιξη των τελευταίων χρόνων έχει οδηγήσει στην ανάπτυξη και την διάδοση ενός μεγάλου και σημαντικού κλάδου της μοντέρνας πληροφορικής, στην μηχανική μάθηση και την βαθιά μάθηση. Ο κλάδος αυτός με την σειρά του ανέπτυξε μια τεχνολογία που έχει κυριαρχήσει στον τομέα της πληροφορικής, τα Μεγάλα Γλωσσικά Μοντέλα (Large Language Models - LLMs). Βασισμένα στην αρχιτεκτονική των transformers όπως το BERT [11] και το RoBERTa [17], έχουν προσφέρει σημαντικές βελτιώσεις στην επεξεργασία και ανάλυση της φυσικής γλώσσας. Τα Μεγάλα Γλωσσικά Μοντέλα αξιοποιούνται σε μια μεγάλη γκάμα εφαρμογών, όπως στην μετάφραση [8], στο sentiment analysis [29], στην ανάπτυξη και διόρθωση κώδικα [4], στην κατηγοριοποίηση [3] [21] κτλ. Συγκεκριμένα, η έκδοση του ChatGPT-3.5 [19] είχε ως αποτέλεσμα να στρέψει την προσοχή της επιστημονικής κοινότητας, του κόσμου των επιχειρήσεων καθώς και των απλών ιδιωτών προς αυτού του είδους τα μοντέλα. Ιδίως στην ακαδημία το ενδιαφέρον για τα Μεγάλα Γλωσσικά Μοντέλα είναι τεράστιο αναφορικά με τις ικανότητές τους και την δυνατότητα ενσωμάτωσης τους σε υπάρχοντα συστήματα για την επίλυση διαφόρων ειδών προβλημάτων. Είναι κοινή παρατήρηση ότι αυτά τα μοντέλα φαίνεται να κατέχουν πολύ καλές γλωσσικές ικανότητες. Παρόλα αυτά, οι δεξιότητες τους αναφορικά με την γνώση γεω-χωρικών ερωτήσεων καθώς και την αντίληψη γεωμετρίας και χώρων φαίνεται να είναι περιορισμένη.

Συνεπώς, το πρόβλημα που εξετάζει η παρούσα εργασία είναι η αξιολόγηση ορισμένων μεγάλων γλωσσικών μοντέλων ως προς την ικανότητά τους να απαντούν ορθά σε ερωτήσεις που σχετίζονται με γεωμετρίες και γεωγραφίες καθώς και αν κατέχουν την εγγενή ευφυΐα να αντιλαμβάνονται τις σχέσεις αντικειμένων σε δισδιάστατους χώρους.

1.2 Κίνητρο και Στόχοι

Τα γεω-χωρικά δεδομένα παίζουν έναν πολύ μεγάλο ρόλο σε πολλές σύγχρονες εφαρμογές, από στρατιωτικές, εκπαιδευτικές μέχρι και εμπορικές [18]. Κρίνεται σκόπιμο λοιπόν να διερευνήσουμε με κάποιον συστηματικό τρόπο αν τα LLMs καταλαβαίνουν την έννοια του γεωγραφικού χώρου καθώς και τις ιδιότητες του. Ταυτόχρονα, χρειάζεται να εξετάσουμε αν τα προαναφερόμενα μοντέλα κατέχουν την απαιτούμενη ευφυΐα που σχετίζεται με την κατανόηση αντικειμένων σε δισδιάστατους και τρισδιάστατους χώρους, τις ιδιοτήτων αυτών καθώς και τις μεταξύ τους σχέσεις. Αυτό θα μας επιτρέψει να βοηθήσουμε τα GIS συστήματα (Geographic Information System) να γίνουν πιο αποδοτικά, να επεκτείνουμε την χρήση των LLMs ώστε να ενσωματωθούν αποτελεσματικά σε διαφόρου τύπου εφαρμογές που διαχειρίζονται, επεξεργάζονται και χρησιμοποιούν γεω-χωρικά δεδομένα. Επίσης, η διερεύνηση των ικανοτήτων των LLM ως προς την χωρική συλλογιστική (spatial reasoning) ανοίγει νέες προοπτικές για την ταυτότητα και την δυνατότητα των εν λόγω μοντέλων.

Μέχρι σήμερα τα LLMs θεωρούνται μοντέλα που κατέχουν καλές γλωσσικές ικανότητες αλλά πασχίζουν με προβλήματα που απαιτούν χρήση λογικής, μαθηματικών και χωρικών ικανοτήτων. Συνεπώς, η διερεύνηση αναφορικά με το αν βρίσκεται εγγενή γνώση στα LLMs για γεωγραφίες, γεωμετρίες και ιδιότητες χώρων είναι αναγκαία για να διαπιστωθεί αν είναι ικανά να προσποιούνται ότι κατανοούν έννοιες που σχετίζονται με άλλες πτυχές της ανθρώπινης ευφυΐας, ανεξάρτητα από τις γλωσσικές και την παραγωγή κειμένου.

Παράλληλα, πέρα από το πρακτικό κομμάτι της εφαρμογής των LLMs για την επίλυση προβλημάτων στον πραγματικό κόσμο, χρειάζεται να κατανοηθούν σε βάθος οι δυνατότητες των εν λόγω μοντέλων καθαρά από την θεωρητική πτυχή της τεχνητής νοημοσύνης. Ένα βασικό ερώτημα του εν λόγου επιστημονικού κλάδου που προκύπτει είναι το εξής: Είναι τα LLMs τεχνολογίες που απλά παράγουν κείμενο βάση υπολογιστικών δομών και στατιστικών-μαθηματικών ιδιοτήτων, με τρόπο που διαμορφώθηκε κατά την φάση της προπόνησης τους, ή μπορούν να έχουν και προοπτικές για να χαρακτηριστούν ως πετυχημένα παραδείγματα γενικής τεχνητής νοημοσύνης (AGI);

Ταυτόχρονα, χρειαζόμαστε να μελετήσουμε αν τα LLMs έχουν τις απαιτούμενες γνώσεις να απαντήσουν σε ερωτήσεις γεωγραφικού τύπου, είτε είναι ποσοτικές είτε ποιοτικές. Η εξέταση αυτή θα μπορέσει να αναδείξει την δυνατότητά των LLMs να χρησιμοποιηθούν με αξιοπιστία από διάφορους φορείς προκειμένου να βρίσκουν απάντηση για ερωτήματα τέτοιας φύσεως καθώς και να μπορούν να συγκρίνονται (ως δείκτης αναφοράς) με άλλα συστήματα που έχουν φτιαχτεί με σκοπό να απαντούν αυτόματα σε ερωτήσεις γεωγραφικού και γεω-χωρικού προσανατολισμού.

Τέλος, η δυνατότητα των LLMs να εξειδικευτούν σε ένα αντικείμενο μέσω τεχνικών transfer learning, από τις οποίες η πιο γνωστή και ευρέως διαδεδομένη είναι αυτής του fine-tuning, μας παρέχει άλλο ένα κατάλληλο έναυσμα για την εκπόνηση της παρούσας εργασίας. Αυτό γίνεται για δύο λόγους. Πρώτον, αν δείξουμε ότι τα LLMs χωρίς στοχευμένη προπόνηση κατέχουν ορισμένη γνώση και καταλαβαίνουν την φύση των γεω-χωρικών ερωτήσεων και αυτή της χωρικής αντίληψης, τότε θα μπορούσαν να βελτιωθούν μέσω fine-tuning. Δεύτερον, προκύπτει το ερώτημα πόση προπόνηση, αναφορικά με τον χρόνο υλοποίησης και τους υπολογιστικούς πόρους που απαιτούνται, χρειάζεται για να γίνει πετυχημένα η διαδικασία του fine-tuning.

Συνεπώς, ένας από τους κύριους στόχος της παρούσας εργασίας είναι ο προσδιορισμός και η αποτύπωση της επίδοσης κάποιων μοντέλων LLM σχετικά με την ικανότητα τους να απαντούν σωστά σε ερωτήσεις γεω-χωρικού χαρακτήρα. Η παραπάνω καταγραφή της επίδοσης, πρέπει να γίνει με μια καλώς ορισμένη και εύλογη μεθοδολογία καθώς και οι μετρικές που θα χρησιμοποιηθούν να είναι εύκολο να εφαρμοστούν σε ίδια ή παρόμοια προβλήματα.

Ο δεύτερος στόχος της παρούσας εργασίας, είναι η διερεύνηση της ικανότητας των μοντέλων να κατανοούν χώρους, τις θέσεις των αντικειμένων που συμπεριλαμβάνονται σε αυτούς καθώς και τις μεταξύ τους σχέσεις. Επίσης, περιλαμβάνεται στον σκοπό η εύρεση του κατάλληλου τρόπου για να εξάγουμε από τα μοντέλα τυχόν πληροφορία που θα δείχνει ότι είναι ικανά για spatial reasoning.

1.3 Μεθοδολογία

Για να επιτευχθεί ο πρώτος στόχος της παρούσας εργασίας θα πρέπει να βρούμε ένα σύνολο δεδομένων που θα εμπεριέχει ερωτήσεις γεω-χωρικού χαρακτήρα και τις αντίστοιχες σωστές απαντήσεις.

Για αυτό τον σκοπό θα χρησιμοποιηθεί ο δείκτης αναφοράς GeoQuestions1089 [15]. Το συγκεκριμένο benchmark εμπεριέχει διάφορες γεω-χωρικές ερωτήσεις ποιοτικού ή ποσοτικού χαρακτήρα. Πέρα από τις ερωτήσεις, το σύνολο δεδομένων εμπεριέχει τις εντολές SPARQL και GeoSPARQL καθώς και τις απαντήσεις που δόθηκαν από την ένωση δύο γνωσιακών γράφων που χρησιμοποιήθηκαν για αυτόν τον σκοπό.

Το GeoQuestions1089 χρησιμοποιείται ως δείκτης αναφοράς για τον έλεγχο της επίδοσης συστημάτων αυτόματης απάντησης ερωτήσεων καθώς και για την αξιολόγηση των LLMs.

Στην παρούσα εργασία, θα χρησιμοποιήσουμε το παραπάνω benchmark για να αξιολογηθούν τρία μικρά LLMs της επιλογής μας κάτω από ορισμένες διαφορετικές μεθοδολογίες και θα παρουσιάσουμε τα αποτελέσματα διαγραμματικά για να συγκρίνουμε την επίδοση τους.

Επιπρόσθετα, θα δημιουργήσουμε τρεις γράφους σε έναν χώρο δύο διαστάσεων και θα περιγράψουμε τις σχέσεις των κόμβων μεταξύ τους. Η προαναφερόμενη περιγραφή θα γίνει στα πλαίσια ενός μαθηματικού φορμαλισμού στον Ευκλείδειο χώρο των δύο διαστάσεων. Το μαθηματικό πλαίσιο και κάποιες από τις ιδιότητές του θα περιγραφούν στο κεφάλαιο 5.

Έπειτα θα εξετάσουμε τα μοντέλα ChatGPT-4o, Llama3.3 70B, Gemma2 9B και το Llama3.1 8B ως προς την επίδοση τους να κατανοούν τον παραπάνω μαθηματικό φορμαλισμό. Αν τα προαναφερόμενα επιτύχουν καλή επίδοση, ιδιαίτερα σε έναν τύπο ερωτήσεων, τότε αυτό θα σημαίνει ότι είναι ικανά να κατανοήσουν κάποιες βασικές πτυχές του spatial reasoning.

Αξίζει να σημειωθεί ότι στα πλαίσια της διπλωματικής χρειάστηκε να χρησιμοποιηθεί κάποια γλώσσα προγραμματισμού για να γίνουν αυτοματοποιημένες διαδικασίες ή να υπολογιστούν οι διάφορες μετρικές. Οι παραπάνω διεργασίες έγιναν σε υπολογιστικό περιβάλλον της Python [1]. Κατά την περιγραφή της μεθοδολογίας, όπου αυτό κριθεί αναγκαίο για σκοπούς σαφήνειας, θα γίνονται αναφορές στις βιβλιοθήκες που χρησιμοποιήθηκαν ή θα παρατίθεται ο κώδικας.

1.4 Δομή της Διπλωματικής

Η παρούσα εργασία χωρίζεται ως εξής: Στο κεφάλαιο 2 περιγράφονται κάποιες θεμελιώδεις έννοιες πάνω στις οποίες θα στηριχτεί το αντικείμενο της εργασίας. Στο κεφάλαιο 3 θα αναλύσουμε άρθρα που σχετίζονται με την παρούσα εργασία και έχουν δουλέψει σε παρόμοια θέματα. Θα περιγράψουμε σύντομα τον σκοπό τους, πολύ περιληπτικά την μεθοδολογία τους και τα αποτελέσματά τους όπου αυτό κριθεί απαραίτητο.

Το κεφάλαιο 4 αναφέρεται στην παραγωγή συνόλου δεδομένων για τις δύο πτυχές της παρούσας εργασίας. Θα αναφερθούμε εν συντομία στο πρόγραμμα που χρησιμοποιήσαμε και τον κώδικα που γράψαμε, προκειμένου το σύνολο δεδομένων να είναι καλύτερο για τα πλαίσια της μετέπειτα ανάλυσης και σύγκρισης του.

Στο κεφάλαιο 5 αναλύουμε την βασική μεθοδολογία που ακολουθήθηκε προκειμένου να φτάσουμε στα τελικά αποτελέσματα. Γίνεται πλήρης περιγραφή των δεδομένων του GeoQuestions1089, των προβλημάτων που αντιμετωπίσαμε καθώς και των τρόπων που επιχειρήσαμε να τα επιλύσουμε. Έπειτα, θέτουμε το πλαίσιο των πειραμάτων, περιγράφουμε τον κώδικα που υλοποιήσαμε και κάποιες βασικές έννοιες, όπου αυτό κρίνεται αναγκαίο.

Στο κεφάλαιο 6 παρουσιάζουμε σε μορφή πίνακα τα αποτελέσματα των πειραμάτων μας και επιχειρούμε μια ανάλυση αυτών προκειμένου να καταλήξουμε σε ορθά συμπεράσματα. Τέλος, στο κεφάλαιο 7 συνοψίζουμε όσα έχουμε κάνει στα πλαίσια της εργασίας μας και έπειτα σχολιάζουμε μελλοντικά βήματα που θα μπορούσαν να γίνουν προκειμένου να επεκταθεί το αντικείμενο που βασίζεται η παρούσα δουλειά, έτσι ώστε να βρεθούν πρόσθετα ευρήματα και να καταλήξουμε σε πιο εύρωστα συμπεράσματα.

[1] https://www.python.org/

2. ΘΕΜΕΛΙΩΔΕΙΣ ΕΝΝΟΙΕΣ

2.1 Σύνοψη Κεφαλαίου 2

Σε αυτήν την ενότητα θα αναλύσουμε τις παρακάτω τέσσερις έννοιες που αποτελούν και την βάση που στηρίζεται η παρούσα διπλωματική εργασία:

- SPARQL
- GeoSPARQL
- Large Language Models
- Spatial Reasoning

Πρωτίστως όμως, θα περιγράψουμε επιγραμματικά κάποιες θεμελιώδεις έννοιες για την καλύτερη κατανόηση των όρων που αναφέρθηκαν παραπάνω.

Μια βάση δεδομένων γραφημάτων (graph database) [5] είναι στην ουσία μια βάση δεδομένων. Η διαφορά με μια συμβατική είναι ότι χρησιμοποιεί γράφους για να αποθηκεύσει και να οργανώσει δεδομένα. Αυτό επιτυγχάνεται μέσω της χρήσης ενός δικτύου κόμβων και ακμών. Οι κόμβοι αυτοί αναπαριστούν οντότητες και οι ακμές τις μεταξύ τους σχέσεις.

Ένας γνωσιακός γράφος (knowledge graph) [12] είναι μια οργανωμένη αναπαράσταση πραγματικών οντοτήτων και των σχέσεων μεταξύ τους. Είναι συνήθως υλοποιημένος ως βάση δεδομένων γραφημάτων που έχει αποθηκευμένες τις σχέσεις ορισμένων οντοτήτων μεταξύ τους. Οι οντότητες σε έναν γνωσιακό γράφο μπορούν να αναφέρονται σε αντικείμενα, γεγονότα, καταστάσεις ή έννοιες. Οι σχέσεις μεταξύ των παραπάνω απεικονίζουν το νόημα και τα συμφραζόμενα του τρόπου που αυτά συνδέονται.

Οι γνωσιακοί γράφοι αποθηκεύουν δεδομένα και τις μεταξύ τους σχέσεις με βάση κάποιο πρότυπο που καθορίζει τις αρχές οργάνωσης των στοιχείων του. Μπορούν να θεωρηθούν ως κανόνες ή κατηγορίες γύρω από δεδομένα που παρέχουν μια ευέλικτη και εννοιολογική οργάνωση με σκοπό να εξαχθούν βαθύτερες σχέσεις των υποκείμενων δεδομένων του. Έτσι, είναι εφικτό να χρησιμοποιηθεί από διάφορους χρήστες προκειμένου να εξάγουν νέα και χρήσιμη πληροφορία στηριζόμενοι στο οργανωτικό πλαίσιο και τους κανόνες πάνω στους οποίους έχει δημιουργηθεί.

Πιο αυστηρά, ως γνωσιακός γράφος ορίζεται ένας κατευθυνόμενος γράφος KG = (V,E) που αποτελείται από ένα σύνολο κόμβων V που αντιπροσωπεύουν οντότητες, τύπους και γεγονότα καθώς και από ένα σύνολο ακμών E που συνδέουν αυτούς τους κόμβους. Οι ακμές περιέχουν ετικέτες και καθορίζουν με ποιόν τρόπο γίνεται η σύνδεση των κόμβων.

Παραθέτουμε επίσης δύο παραδείγματα γνωσιακών γράφων, στο διάγραμμα 2.1 που αντιπροσωπεύουν διαφορετικά θέματα και εμπεριέχουν έναν κόμβο που αντιστοιχεί σε μία ίδια οντότητα του πραγματικού κόσμου.

Μια άλλη έννοια που κρίνεται σκόπιμο να αναφέρουμε είναι αυτή της RDF (Resource Description Framework) [23]. Το RDF είναι μια μέθοδος για την περιγραφή και ανταλλαγή δεδομένων που είναι οργανωμένα σε γράφους. Ουσιαστικά, είναι ένας κατευθυνόμενος γράφος που αποτελείται από τρεις δηλώσεις (s,p,o).

Συγκεκριμένα αποτελείται από: (1) τον κόμβο ενός υποκείμενου s, (2) την ακμή που συνδέει το υποκείμενο με το αντικείμενο p, και (3) τον κόμβο που αντιπροσωπεύει το αντικείμενο (o).

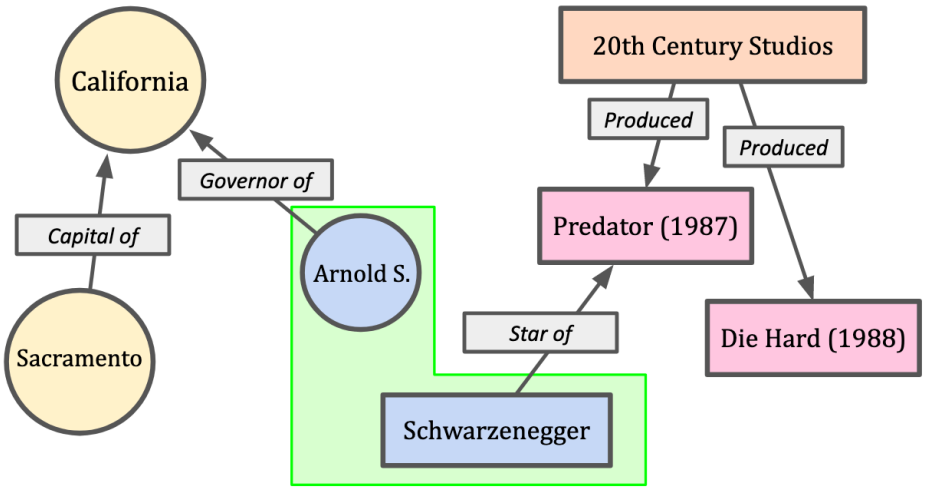

Figure 2.1: Two knowledge graphs

Συνεπώς, για έναν γνωσιακό γράφο που απεικονίζεται σε RDF έχουμε ότι το s,o ανήκουν στο σύνολο των κόμβων (V) και ότι το p στο σύνολο των ακμών (E), όπως ορίστηκαν στον παραπάνω ορισμό.

Τέλος, ως γεω-χωρικά δεδομένα (Geospatial data) [15] είναι πληροφορία που περιγράφει αντικείμενα, γεγονότα ή άλλα χαρακτηριστικά που εντοπίζονται πάνω ή κοντά στην επιφάνεια της γης. Αυτού του τύπου τα δεδομένα, τυπικά συνδυάζουν πληροφορίες για τοποθεσίες (συνήθως συντεταγμένες της γης) και για χαρακτηριστικά (γνωρίσματα αντικειμένων, γεγονότων ή φαινομένων) με χρονική πληροφορία (χρόνος ή χρονική διάρκεια που η τοποθεσία και τα χαρακτηριστικά υπάρχουν).

2.2 Ορισμός SPARQL και GeoSPARQL

Η SPARQL [23] είναι μια RDF query γλώσσα και πρωτόκολλο που χρησιμοποιείται για να ανακτά και να τροποποιεί δεδομένα που είναι αποθηκευμένα σε μορφή RDF. Τυποποιήθηκε από την RDF Data Access Working Group [1] του World Wide Web Consortium [1] και αναγνωρίζεται ως μια από τις βασικές τεχνολογίες του Σημασιολογικός Ιστός (Semantic Web [25]).

Θα μπορούσαμε να πούμε ότι η SPARQL είναι ανάλογη της SQL, με την διαφορά ότι η πρώτη είναι σχεδιασμένη ώστε ο χρήστης να κάνει ερωτήσεις σε δεδομένα που είναι εκφρασμένα σε μορφή γράφου. Αντίθετα, η SQL χρησιμοποιείται για τον ίδιο σκοπό όταν όμως τα δεδομένα είναι αποθηκευμένα σε σχεσιακές βάσεις δεδομένων.

Ένα παράδειγμα εντολής της γλώσσας SPARQL που ανακτά το όνομα και το email ανθρώπων από έναν γράφο είναι το παρακάτω:

```
SELECT ?name ?email
WHERE {
  ?person <http://xmlns.com/foaf/0.1/name> ?name .
  ?person <http://xmlns.com/foaf/0.1/mbox> ?email .
}
```

Επιπρόσθετα, η GeoSPARQL [6] είναι ένα υπόδειγμα για την αναπαράσταση και την

[1] https://www.w3.org/

υποβολή ερωτημάτων γεω-χωρικών συνδεδεμένων δεδομένων από τον Σημασιολογικό Ιστό. Τυποποιήθηκε από την Open Geospatial Consortium [2] ως GeoSPARQL. Στην ουσία, ορίζει ένα λεξιλόγιο για την αναπαράσταση geospatial δεδομένων σε μορφή RDF και αποτελεί μια επέκταση της γλώσσας SPARQL για την επεξεργασία γεω-χωρικών δεδομένων.

2.3 Ορισμός Large Language Model και spatial reasoning

Τα μεγάλα γλωσσικά μοντέλα (LLMs) είναι αλγόριθμοι βαθιάς μηχανικής μάθησης που μπορούν να αναγνωρίζουν, συνοψίζουν, μεταφράζουν, προβλέπουν και να αναπαράγουν περιεχόμενο, κυρίως κείμενο, αφού είναι εκπαιδευμένα σε πολύ μεγάλα σύνολα δεδομένων.

Τα LLMs κυρίως αναπαριστούν μια κλάση αρχιτεκτονικής βαθιάς μάθησης που αποκαλείται transformer [28].

Ουσιαστικά, ένα transformer μοντέλο είναι ένα νευρωνικό δίκτυο που μαθαίνει συμφραζόμενα και νοήματα καταγράφοντας σχέσεις σε σειριακά δεδομένα, όπως είναι οι λέξεις που αποτελούν μια πρόταση.

Η δομή των transformers [11] αποτελείται από πολλαπλά transformers blocks, που αποκαλούνται επίσης και επίπεδα. Παραδείγματος χάριν, ένα tranformer μπορεί να έχει τα ακόλουθα επίπεδα: self-attention, feed-forward και normalization, όπου όλα αυτά δουλεύουν μαζί για την αποκωδικοποίηση των δεδομένων εισαγωγής ώστε να προβλέψουν την ροή που θα εξάγει κατά την διάρκεια του συμπεράσματος.

Τα επίπεδα, έτσι όπως περιγράφηκαν προηγουμένως μπορούν να στοιβαχτούν μαζί για να δημιουργήσουν βαθύτερες δομές transformer και πιο ισχυρά γλωσσικά μοντέλα.

Η αρχιτεκτονική transformer εισήχθηκε πρωτίστως από την Google το 2017 στο άρθρο με τίτλο [28] ”Attention is all you Need”. Υπάρχουν δύο καινοτομίες κλειδιά που κάνουν την εν λόγω αρχιτεκτονική ιδιαίτερα ικανή για τα μεγάλα γλωσσολογικά μοντέλα: positional encodings και self-attention.

Το positional encoding ενσωματώνει την σειρά με την οποία τα δεδομένα εισαγωγής εμφανίζονται μέσα στην δοθείσα πρόταση. Ουσιαστικά, αντί να τροφοδοτούμε λέξεις μέσα σε μία πρόταση διαδοχικά στο νευρωνικό δίκτυο, χάρις στο positional encoding, οι λέξεις μπορούν να τοποθετηθούν χωρίς την προκαθορισμένη σειρά εμφάνισής τους.

Από την άλλη, το self-attention εκχωρεί ένα βάρος σε κάθε κομμάτι των δεδομένων εισαγωγής όταν το επεξεργάζεται. Αυτό το βάρος σηματοδοτεί την σημασία αυτού του στοιχείου εισαγωγής στο πλαίσιο των υπόλοιπων κομματιών των στοιχείων εισαγωγής. Με άλλα λόγια, τα μοντέλα δεν χρειάζεται πλέον να αφιερώνουν την ίδια προσοχή σε όλα τα στοιχεία που εισάγονται σε αυτά και μπορούν να εστιάσουν σε σημεία που έχουν την ύψιστη σημασία. Η αναπαράσταση που αναφέρεται σε ποια κομμάτια το νευρωνικό δίκτυο πρέπει να δώσει περισσότερο προσοχή μαθαίνεται σταδιακά, καθώς το μοντέλο διατρέχει και αναλύει μια σειρά από δεδομένα.

Οι παραπάνω δύο τεχνικές σε συνδυασμό επιτρέπουν να γίνεται ανάλυση των τρόπων και συμφραζομένων στα οποία ξεχωριστά στοιχεία επηρεάζουν και σχετίζονται μεταξύ τους σε μεγάλες αποστάσεις, μη διαδοχικά.

Η ικανότητα της μη-σειριακής επεξεργασίας δεδομένων επιτρέπει την αποσύνθεση πολύπλοκων προβλημάτων σε πολλαπλούς μικρότερους και ταυτόχρονους υπολογισμούς. Από την φύση τους, οι κάρτες γραφικών (Graphical Processing Units-GPUs) είναι εφοδιασμένες

[2] https://www.ogc.org/

για να λύνουν αυτού του είδους τα προβλήματα παράλληλα, επιτρέποντας επεξεργασία μεγάλης κλίμακας σε δεδομένα που δεν είναι επισημασμένα και μεγάλων transformers δικτύων.

Η χωρική συλλογιστική (spatial reasoning) [7] είναι ευρέως διαδεδομένη στις καθημερινές μας αλληλεπιδράσεις με τον κόσμο. Αποτελεί τον ακρογωνιαίο λίθο της ικανότητα μας να σχεδιάσουμε ένα ταξίδι, να προσδιορίσουμε οντότητες στον χώρο και να φανταστούμε αντικείμενα όταν ακούμε περιγραφές των διατάξεων τους σε φυσική γλώσσα.

Για παράδειγμα, μια απλή μορφή χωρικής συλλογιστικής εξαρτάται από παραδοχές που περιγράφουν μια μονοδιάστατη διάταξη αντικειμένων, όπως:

- Ο Α είναι δεξιά του Β.
- Ο C είναι αριστερά του Β.
- Οπότε, ο Α είναι δεξιά του C.

Συνεπώς, θα μπορούσαμε να ορίσουμε γενικότερα το spatial reasoning ως την ικανότητα να κατανοούμε, να απεικονίζουμε και να χειριζόμαστε αντικείμενα στον χώρο. Αυτή η ικανότητα συμπεριλαμβάνει το να σκεφτόμαστε για θέσεις, διαστάσεις, κινήσεις και σχέσεις μεταξύ αντικειμένων σε έναν χώρο δύο, τριών ή περισσότερων διαστάσεων.

3. ΣΧΕΤΙΚΗ ΕΡΓΑΣΙΑ

Στην παρούσα ενότητα θα περιγράψουμε κάποια άρθρα στα οποία στηρίζεται η παρούσα διπλωματική, ως προς τους στόχους και τις μεθοδολογίες που θα ακολουθήσουν.

3.1 Σύστημα Απάντησης Ερωτήσεων και GeoQuestions1089

Πρωτίστως, θα αναφερθούμε σε ένα κομμάτι του άρθρου ”The Question Answering System GeoQA2 and a New Benchmark For its Evaluation” [15]. Η συγκεκριμένη δουλειά παρουσιάζει το GeoQA2, ένα σύστημα ερωτήσεων-απαντήσεων που ανταποκρίνεται σε ερωτήματα χρησιμοποιώντας την ένωση δύο γνωσιακών γράφων. Στην ουσία το GeoQA2 παίρνει σαν είσοδο δεδομένων μια ερώτηση διατυπωμένη σε φυσική γλώσσα, την μεταφράζει σε ένα σύνολο SPARQL/GEOSPARQL αιτημάτων, έπειτα τα ταξινομεί σε αύξουσα σειρά και εκτελεί αυτά που κατατάχθηκαν υψηλότερα, λαμβάνοντας την απάντηση από την ένωση δύο γνωσιακών γράφων. Έπειτα, παρουσιάζεται το benchmark του GeoQuestions1089 που εμπεριέχει απαντήσεις από την ένωση των δύο προαναφερόμενων γράφων. Ακολουθεί η αξιολόγηση του GeoQA2 πάνω στο GeoQuestions1089 και τα αποτελέσματα συγκρίνονται με αυτά ενός παρόμοιου συστήματος, εν ονόματι Hamzei. Στην αναφερόμενη εργασία, επιχειρείται ταυτόχρονα η ανάλυση της επίδοσης του ChatGPT-3.5 πάνω στο benchmark GeoQuestions1089 για να διαπιστωθεί κατά πόσο μπορεί το πιο γνωστό LLM μοντέλο να απαντήσει σωστά. Επίσης, συγκρίνεται η επίδοση του ChatGPT-3.5 με αυτή του συστήματος GeoQA2. Τα αποτελέσματα δείχνουν ότι η GeoQA2 αποδίδει καλύτερα από την Hamzei, παρόλο που και οι δύο μηχανές έχουν περιθώριο βελτίωσης. Το ChatGPT-3.5 είχε καλή επίδοση σε πιο εύκολες ερωτήσεις, κυρίως αυτές που έχουν ως απάντηση ”yes” ή ”no” και τις ερωτήσεις υπερθετικού και συγκριτικού χαρακτήρα. Παρόλα αυτά, όσο πιο δύσκολες γίνονται οι ερωτήσεις, τόσο χειροτερεύουν οι απαντήσεις του ChatGPT. Τέλος, στις ερωτήσεις που συμπεριλαμβάνουν πληθυσμιακές στατιστικές, το LLM ήταν ασυνεπές και ασταθές ως προς τις απαντήσεις του.

3.2 Αξιολόγηση των Ικανοτήτων Χωρικής Λογικής του ChatGPT-4

Στη συνέχεια, θα αναφερθούμε συνοπτικά στην εργασία του ”An Evaluation of ChatGPT-4’s Qualitative Spatial Reasoning Capabilities in RCC-8” [9].

Μέσα σε αυτό άρθρο εξετάζεται η ικανότητα του ChatGPT-4o να μπορεί να επεξεργάζεται και να εξάγει σωστά συμπεράσματα στον χώρο του Qualitative Spatial Reasoning χρησιμοποιώντας το mereotopological calculus, RCC-8.

Ουσιαστικά, το Qualitative Spatial Reasoning (QSR) [9] είναι ένας κλάδος μελέτης που εστιάζει στην αναπαράσταση και ανάλυση σχετικά με χωρικές σχέσεις μεταξύ αντικειμένων, περιοχών ή οντοτήτων με έναν μη-αριθμητικό τρόπο. Αντί να εξαρτάται από ακριβείς μετρήσεις όπως αποστάσεις, γωνίες και συντεταγμένες, το QSR χρησιμοποιεί συμβολικές περιγραφές ενός χωρικού σχήματος, όπως ”δίπλα σε”, ”μέσα”, ”αλληλοεπικαλύπτεται” και ”μακριά από”.

Το RCC-8 (Region Connection Calculus) [9] είναι ένας συγκεκριμένος φορμαλισμός μέσα στον mereotopological στοχασμό. Ορίζει οκτώ θεμελιώδεις σχέσεις μεταξύ δύο χωρικές περιοχές βάση των τοπολογικών ιδιοτήτων, αναφορικά με το πόσο αλληλεπικαλύπτονται ή πόσο είναι συνδεδεμένες.

Ο ανωτέρω φορμαλισμός χρησιμοποιείται ευρέως για την αναπαράσταση και την αποτύπωση προβλημάτων που εμπίπτουν στην περιοχή Qualitative Spatial Reasoning (QSR).

Στην εξεταζόμενη εργασία διενεργούνται μια σειρά από πειράματα ως εξής: αρχικά περιγράφονται στο LLM οι σχέσεις και η δομή του RCC-8 calculus, μετά υποβάλλονται διάφορες ερωτήσεις που απαιτούν να εφαρμοστεί ο εν λόγω λογισμός για να βρεθούν κάποιες πιθανές νέες σχέσεις περιοχών που έχουν δοθεί και στην συνέχεια, εξετάζονται και αξιολογούνται οι απαντήσεις που λαμβάνονται από το μοντέλο.

Το πρώτο πείραμα αφορά το Compositional Reasoning in RCC-8, το δεύτερο το Preferred Compositions in RCC-8 και το τρίτο το Spatial Continuity.

Μαζεύοντας τις απαντήσεις από το LLM στα πειράματα βάση των ερωτήσεων που δόθηκαν στο μοντέλο, το άρθρο καταλήγει ότι τα αποτελέσματα υποστηρίζουν την ευρέως αποδεκτή άποψη ότι τα LLMs δυσκολεύονται να προβούν σε ορθή συλλογιστική χρησιμοποιώντας τον υπό εξέταση φορμαλισμό.

Η επίδοση που πέτυχε το ChatGPT-4 για να υπολογίσει ολόκληρο τον composition πίνακα για το RCC-8 είναι 71,94%, το οποίο είναι πολύ καλύτερο από το αν απάνταγε τυχαία και υποδεικνύει μια υποτυπώδη ικανότητα να πραγματοποιεί τέτοιους υπολογισμούς.

Μια λεπτομερή ανάλυση των παραγόμενων απαντήσεων του ChatGPT-4 φανερώνει την ικανότητα του να πραγματοποιεί χωρική λογική (spatial reasoning), αλλά πολλές φορές αποτυγχάνει, κάνοντας σε ορισμένες περιπτώσεις βασικά λάθη.

Επίσης, δείχνει μια ασυνέπεια στο ότι μπορεί να αναλύσει μια σχέση αλλά όχι και την αντίστροφή της. Ορισμένες φορές μάλιστα μπερδεύει μια σχέση με την αντίστροφη της.

3.3 Εξετάζοντας την χωρική αντίληψη των Μεγάλων Γλωσσικών Μοντέλων

Το επόμενο άρθρο με τίτλο ”Testing spatial reasoning of Large Language Models: the case of tic-tac-toe” [16] που στηρίχθηκε κομμάτι της μεθοδολογίας της παρούσας εργασίας, εξετάζει την χωρική αντίληψη (spatial reasoning) των μεγάλων γλωσσικών μοντέλων μέσω του παιχνιδιού tic-tac-toe.

Συγκεκριμένα, εξετάζεται ο τρόπος που τα LLMs μπορούν να επιλέξουν κινήσεις στο εν λόγω παιχνίδι προκειμένου να αξιολογηθούν οι νοητικές ικανότητες τους όταν η πληροφορία για την εξαγωγή συμπερασμάτων εμπίπτει σε ένα χωρικό περιεχόμενο.

Για την ευόδωση αυτού του σκοπού στα πλαίσια της αναφερόμενης εργασίας επιστρατεύτηκαν μια σειρά από μοντέλα και τους ανατέθηκε να παίξουν παρτίδες του προαναφερόμενου παιχνιδιού έναντι του γνωστού αλγορίθμου minimax algorithm [16].

Ο παραπάνω είναι ένας αλγόριθμος επιλογής απόφασης και λειτουργεί επιλέγοντας την καλύτερη πιθανή κίνηση για έναν παίκτη σε ένα παίγνιο μηδενικού αθροίσματος δύο παικτών, όπου η νίκη του ενός είναι ισοδύναμη με την ήττα του άλλου. Περιληπτικά, η εν λόγω στρατηγική κοιτάζει όλες τις πιθανές κινήσεις σε ένα παιχνίδι και ανακαλύπτει την καλύτερη κίνηση λαμβάνοντας υπόψη το χειρότερο πιθανό σενάριο.

Το πείραμα στο πλαίσιο που διεξάγεται [16], είναι μη-τετριμμένο, αφού περιλαμβάνει την αναγνώριση χαρακτήρων και την ικανότητα να εξάγονται συμπεράσματα βάση σημείων-θέσεων σε έναν χώρο δύο διαστάσεων. Τα αποτελέσματα υποστηρίζουν την ευρέα αποδεκτή άποψη ότι τα LLMs δεν είναι τόσο καλά στην χωρική αντίληψη, αν και τονίζεται ότι θα πρέπει να γίνουν περισσότερα πειράματα αυτού του είδους.

3.4 Αξιολόγηση διαλεκτικού γλωσσικού μοντέλου

Μια επιπρόσθετη εργασία που επιχειρεί να εξετάσει την ικανότητα χωρικής αντίληψης των αναφερόμενων μοντέλων είναι από τους Anthony G Cohn και Jose Hernandez-Orallo ” [10]. Η εν λόγω εργασία θεωρεί ότι πολλά από τα μεγάλα και τυποποιημένα benchmarks που χρησιμοποιούνται για την αξιολόγηση της επίδοσης των LLMs δεν αποτυπώνουν σωστά την πραγματικότητα.

Αυτό γίνεται διότι πολλοί από αυτούς τους δείκτες αναφοράς εμπεριέχουν πολύ πληροφορία που ενδέχεται να έχει συναντήσει το μοντέλο κατά την διαδικασία της προπόνησης του. Επίσης, πολλές από τις ερωτήσεις σε αυτούς τους δείκτες, είναι πολλαπλής επιλογής, αυξάνοντας την αναξιοπιστία του ελέγχου.

Συνεπώς, στο αναφερόμενο άρθρο επιλέγεται να εξετάσουν κάποια LLMs, όπως το ChatGPT-4, προκρίνοντας μια πιο διαλεκτική μέθοδο. Αυτό μπορεί να επιτευχθεί επειδή τα LLMs μπορούν να δεχθούν ερωτήσεις σειριακά, με την απάντηση της τελευταίας ερώτησης να επηρεάζεται από την ροή της συζήτησης που έχει ήδη γίνει. Παρατηρείται κακή απόδοση από την πλευρά των μοντέλων στην διαλεκτική μέθοδο του πειράματος, κάνοντας συχνά βασικά λάθη κατά την διαδικασία της απάντησης.

3.5 Χωρική αντίληψη στα Visual-LLMs

Το επόμενο άρθρο που θα παρουσιάσουμε συνοπτικά [24] εξετάζει τον τρόπο με τον οποίο θα μπορούσαν τα V-LLMs, δηλαδή μοντέλα που παράγουν κείμενο έχοντας λάβει περιγραφές και εικόνες ως στοιχεία εισόδου, θα μπορούσαν να βελτιώσουν την απόδοσή τους.

Ο λόγος που εξετάζεται κάποιος πιθανός τρόπος είναι η κακή επίδοση στην χωρική αντίληψη που έχουν αυτά τα μοντέλα. Συγκεκριμένα, αναφέρεται πως τα V-LLMs μπορούν να δώσουν περιγραφικές λεξιλογικές απαντήσεις, όμως η χωρική του αντίληψη χαρακτηρίζεται υπο-ανεπτυγμένη αφού ορισμένες φορές μπερδεύουν το αριστερά με το δεξιά.

Στην παρούσα τους δουλειά, εξερευνούν συγκεκριμένες οδηγίες σχετικά με χωρική πληροφορία, προκειμένου να βρεθεί η κατάλληλη διεργασία fine-tuning ώστε να μπορέσουν τα V-LLMs να αναλύσουν ουσιαστικά και να παράξουν συντεταγμένες από εικόνες.

Στην ουσία, η εργασία βασίζεται στην υπόθεση ότι η επεξεργασία τοποθεσίας αντικειμένων που παρουσιάζεται σε φυσική γλώσσα θα οδηγήσει τα μοντέλα να διαλέξουν τις κατάλληλες περιοχές, σε αντίθεση από το να στηρίζονται σε χαρακτηριστικά των περιοχών που παρέχονται από την αρχιτεκτονική τους. Η μεθοδολογία που ακολουθούν, η οποία χαρακτηρίζεται από 4 διαφορετικά στάδια προπόνησης, έχει ως αποτέλεσμα τα V-LLMs να μπορούν να αναλύουν καλύτερα σχετικά με την σύνθεση της εικόνας χρησιμοποιώντας τις συντεταγμένες της τελευταίας που έχουν δοθεί ως κείμενο εισόδου.

3.6 Προτροπή σχετικής τοποθεσίας σε LLMs

Το τελευταίο άρθρο που σχετίζεται με το αντικείμενο της παρούσας εργασίας [22] αποσκοπεί στο να βρει μια στρατηγική prompting για να αυξήσει την επίδοση των LLMs στο κομμάτι του spatial reasoning.

Το πρόβλημα που θα χρησιμοποιηθεί αφορά ένα σενάριο που περιλαμβάνει ένα 5x5

πλέγμα όπου καθορίζονται τυχαία σε αυτό το σημείο εκκίνησης, το σημείο προορισμού και 3 σημεία που εμπεριέχονται εμπόδια.Το παιχνίδι θεωρείται επιτυχές όταν ο παίκτης καταφέρει να φτάσει από το σημείο εκκίνησης στο σημείο προορισμού το πολύ σε 10 γύρους.

Στο LLM περιγράφονται οι κανόνες, το περιβάλλον και ο στόχος του παιχνιδιού και δίνουμε σε κάθε γύρο το εν λόγω μοντέλο να κινηθεί ένα βήμα στο πλέγμα σε μια εκ εξής κατευθύνσεων: αριστερά, δεξιά, πάνω και κάτω.

Η τεχνική prompting που ακολουθήθηκε λειτουργεί με το να ενημερώνεται σε κάθε κίνηση το LLM με την σχετική του τοποθεσία αναφορικά με τον στόχο. Το παραπάνω αποκαλείται Relative location prompting. Στην συνέχεια, συγκρίνει αυτή την μέθοδο με την βασική στην οποία το LLM δεν λαμβάνει την σχετική θέση του παίκτη αναφορικά με τον στόχο και μια τρίτη στην οποία κινείται τυχαία σε κάθε γύρο μέχρι να βρει τον στόχο.

Τα αποτελέσματα δείχνουν ότι η Relative location prompting πετυχαίνει την καλύτερη επίδοση με 68%, η τυχαία μέθοδος με 19% και η βασική μέθοδος prompting 15%.

4. ΠΑΡΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΗΝ ΔΙΕΞΑΓΩΓΗ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ

4.1 Σύνοψη κεφαλαίου 4

Στα πλαίσια αυτού του κεφαλαίου θα περιγράψουμε την διαδικασία παραγωγής συνόλου δεδομένων για τα δύο υπο-εξέταση θέματα που επικεντρώνεται η παρούσα εργασία. Πρωτίστως, θα χρειαστούμε τις απαντήσεις των LLMs στις αντίστοιχες ερωτήσεις από το δείκτη αναφοράς GeoQuestions1089 που θα χρησιμοποιηθούν για την υλοποίηση των πειραμάτων μας.

Για το δεύτερο στόχο της εργασίας η παραγωγή ενός συνόλου γράφων είναι τετριμμένη διεργασία και δεν χρειάζεται να χρησιμοποιήσουμε κάποιο ειδικό software μιας και οι γράφοι είναι μικροί σε πλήθος και σε μέγεθος. Ταυτόχρονα, θα τους περιγράψουμε συνοπτικά στα LLMs λαμβάνοντας υπόψη μας την κατεύθυνση των κόμβων μεταξύ τους βάση ενός μαθηματικού μοντέλου που περιγράφεται στο κεφάλαιο 5.

4.2 Παραγωγή απαντήσεων των LLMs στο Geoquestions1089

Αρχικά, αποφασίσαμε να εξετάσουμε τα μοντέλα llama 3.1 [2], mistral [13] και gemma2 9B [27]. Το Llama 3.1 είναι ένα open-source LLM που αναπτύχθηκε από την Meta και κυκλοφόρησε τον Ιούλιο 2024. Τρεις τύποι ως προς το μέγεθος των παραμέτρων είναι διαθέσιμοι. Το πρώτο είναι το μικρό που έχει 8 δις παραμέτρους, το μεσαίο με 70 δις και τέλος το μεγάλο με 405 δις.

Στα πλαίσια της παρούσας ανάλυσης χρησιμοποιήθηκε το μικρό μοντέλο καθώς είναι σχεδιασμένο για να μπορεί να επιστρατευτεί αποδοτικά σε μικρής κλίμακας υπολογιστικά συστήματα.

Το mistral είναι μια σειρά από open-source LLMs που έχουν αναπτυχθεί από την Mistral AI, μια ευρωπαϊκή AI start-up. Το mistral που χρησιμοποιήσαμε έχει 7 δις παραμέτρων.

Το τελευταίο open-source μεγάλο γλωσσικό μοντέλο είναι το gemma2 που έχει αναπτυχθεί από την Google. Έχει τρεις εκδοχές ως προς το μέγεθος της αρχιτεκτονικής του:

- Gemma 2 των 2 δις παραμέτρων
- Gemma 2 των 9 δις παραμέτρων
- Gemma 2 των 27 δις παραμέτρων

Στην εν λόγω εργασία θα αξιολογηθεί η δεύτερη εκδοχή του μοντέλου, ήτοι των 9 δις, καθότι το πρώτο κρίθηκε πολύ μικρό για να μπορέσει να έχει κάποια αξιόλογη επίδοση και το τρίτο είναι πολύ μεγάλο μοντέλο απαιτεί την ύπαρξη πολλών υπολογιστικών πόρων, γεγονός που δεν ανταποκρίνεται βάση των περιορισμών που αντιμετωπίζουμε στην παρούσα εργασία.

Για την ευόδωση του παραπάνω σκοπού εγκαταστάθηκε το πρόγραμμα ollama τοπικά στο υπολογιστικό σύστημα που στηριχθήκαμε. Το ollama είναι μια πλατφόρμα για την εγκατάσταση και την διαχείριση μεγάλων γλωσσικών μοντέλων. Επίσης, παρέχει και το κατάλληλο API σε Python και Javascript περιβάλλον για να είναι εφικτή η αποδοτική ενσωμάτωση των LLMs σε διάφορες εφαρμογές που αποσκοπούν να ενσωματώσουν generative AI στις δυνατότητες τους.

Τα παραπάνω λοιπόν, μπορούν να γίνουν εφικτά τοπικά σε ένα απλό υπολογιστικό σύστημα με την βοήθεια του ollama.

Πρωτίστως, εγκαθιστούμε τοπικά στον υπολογιστή την εφαρμογή του ollama και κατεβάζουμε από το αποθετήριο της επίσημης ιστοσελίδας τα μοντέλα που μας ενδιαφέρουν. Μετά εγκαθιστούμε το αντίστοιχο API για την ενσωμάτωση τους στο Python περιβάλλον που δουλεύουμε. Το module που φορτώνουμε στο αρχείο Python λέγεται ollama και διαλέγοντας το κατάλληλο μοντέλο ως όρισμα καθώς και την ερώτηση που μας ενδιαφέρει από το GeoQuestions1089 παίρνουμε την αντίστοιχη απάντηση του LLM.

Η διεργασία αυτή γίνεται μέσω της μεθόδου generate. Παρακάτω δίνεται ο κώδικας για την παραγωγή απαντήσεων για το μοντέλο mistral:

```
import json, ollama, re
MODELS = ["llama3.1:8b", "mistral", "gemma2"]
new_data = dict()
with open(pathFile, 'r') as file:
    data = json.load(file)
    for key in tqdm(data.keys()):
        ques = data[key]['Question']
        response = ollama.generate("mistral", prompt=ques)
        x = response['response']
        new_data[key] = {'Question': ques, 'Answer': x}
```

4.2.1 Εξερευνώντας τις απαντήσεις των LLMs

Αφού πήραμε τα δεδομένα και τα αποθηκεύσαμε, είδαμε ότι εμπεριέχουν χαρακτήρες όπως αστερίσκους "*", emojis και αλλά σύμβολα που δεν είναι χρήσιμα στα πλαίσια της ανάλυσης μας. Αυτά αφαιρέθηκαν με την μέθοδο sub του module re:

- cleaned_text = re.sub(r' * *' , '', cleaned_text)

Αντίστοιχα, αφαιρέσαμε τα emojis με την μέθοδο compile του module re.

Μια από τις βασικές απορίες που δημιουργήθηκαν ήταν να ελέγξουμε την έκταση της περιττολογίας που εμφανίζουν οι απαντήσεις των μοντέλων. Αυτό θα εξεταστεί σε αναφορά με τις απαντήσεις που υπάρχουν στο benchmark GeoQuestions1089. Κατά αυτόν τον τρόπο, ορίσαμε τον αριθμοδείκτη verbosity ως εξής:

$$verbosity = \frac{len(ansLlm)}{len(ansGeo)}$$

όπου το len(ansLlm) αναφέρεται στον αριθμό των λέξεων που υπάρχουν στην απάντηση του LLM και το len(ansGeo) στο πλήθος της απάντησης αναφοράς.

Ουσιαστικά, αν έχουμε verbosity = 6, για μια δεδομένη ερώτηση, τότε για κάθε μια λέξη της σωστής απάντησης το LLM απάντησε με έξι. Προφανώς, όσο πιο μεγάλη είναι η μετρική, τόσο πιο πολύ περιτολλογεί το μοντέλο.

Έπειτα, υπολογίσαμε αυτή την μετρική για όλες τις ερωτήσεις και για κάθε μοντέλο, τις συγκεντρώσαμε σε μια λίστα και μέσω του module matplotlib.pyplot [1] και seaborn [2] υπολογίσαμε

[1] https://matplotlib.org/

[2] https://seaborn.pydata.org/

και απεικονίσαμε τα ιστογράμματα και εκτιμήσαμε τις κατανομές μέσω του Kernel Density Estimation.

Παρουσιάζουμε τα αποτελέσματα για τις απαντήσεις του Llama 3.1 για τις τρεις κατηγορίες ερωτήσεων στα διαγράμματα 4.1, 4.2, 4.3.

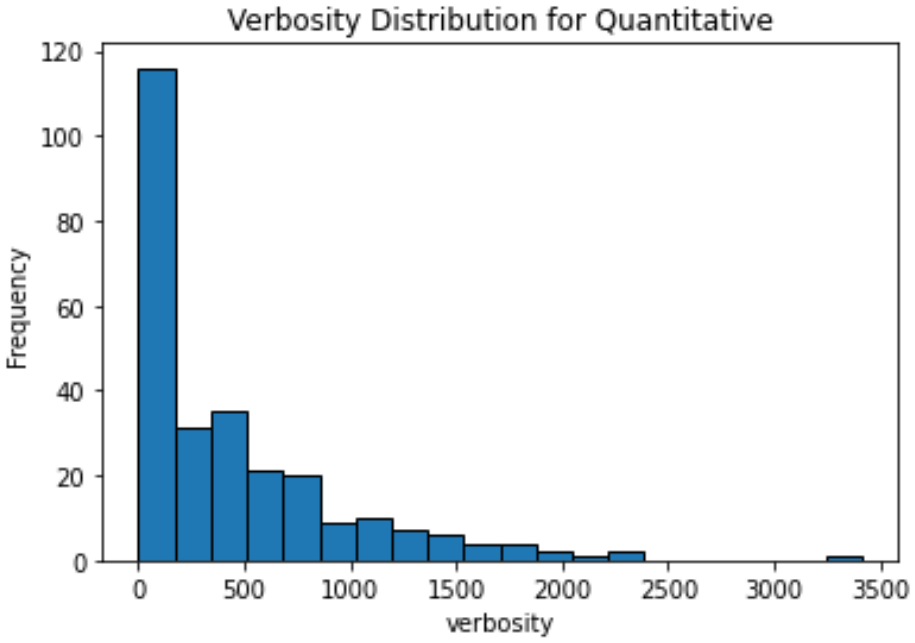

Figure 4.1: Ιστόγραμμα για ποσοτικές ερωτήσεις

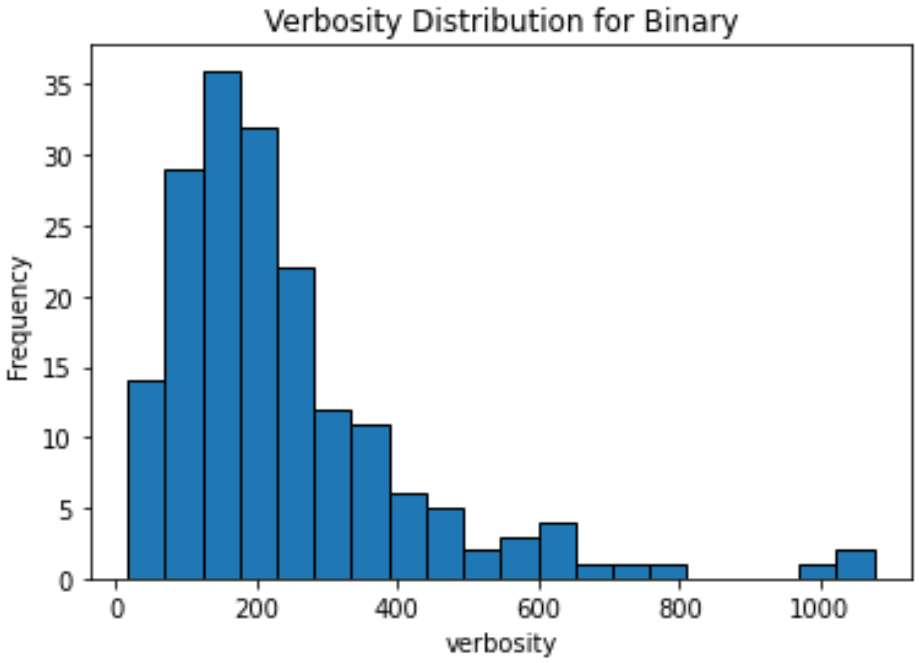

Figure 4.2: Ιστόγραμμα για Δυαδικές ερωτήσεις

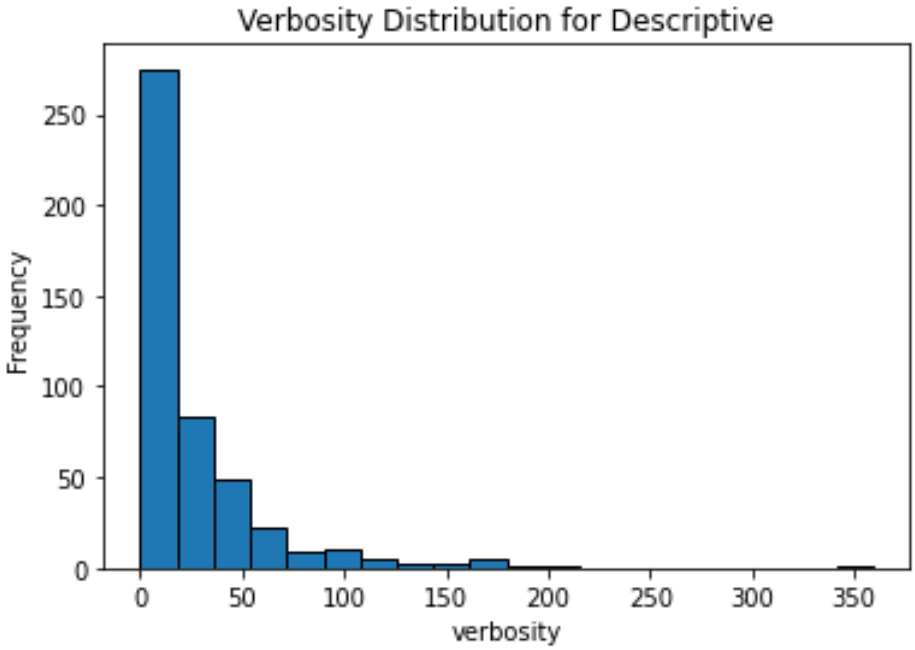

Figure 4.3: Ιστόγραμμα για περιγραφικές ερωτήσεις

Δεδομένου των παραπάνω ιστογραμμάτων θα εκτιμήσουμε τις αντίστοιχες κατανομές πυκνότητας πιθανότητας με την μέθοδο Kernel Density Estimation (KDE) [20].

Η εν λόγω μέθοδος είναι ένας μη-παραμετρικός τρόπος να εκτιμήσουμε την συνάρτηση πιθανότητα πυκνότητας (PDF) μιας τυχαίας μεταβλητής. Σε αντίθεση με τα ιστογράμματα, όπου είναι διακριτά και βασίζονται σε προκαθορισμένους κάδους (εύρη τιμών), η μέθοδος KDE δημιουργεί μία ομαλή και συνεχή αναπαράσταση της κατανομής των δεδομένων.

Το παραπάνω γίνεται εφικτό εφαρμόζοντας μια kernel συνάρτηση (ομαλή καμπύλη) για κάθε σημείο των δεδομένων και αθροίζοντας τα. Η kernel συνάρτηση που χρησιμοποιήθηκε για τον παραπάνω σκοπό είναι η Gaussian.

Αυτό υλοποιήθηκε μέσω της συνάρτησης kdeplot του module seaborn. Παρατίθεται κομμάτι του κώδικα παρακάτω:

```
verbosity = dict()
Descr = typeKeys['Descr']
Quant = typeKeys['Quant']
Binary = typeKeys['Binary']
verbosity['Descr'] = list()
verbosity['Quant'] = list()
verbosity['Binary'] = list()

for key in clGeoAns.keys():

    gans = clGeoAns[key]['Answer']
    lans = clLlmAns[key]['Answer']
    lengan = len(''.join(gans).replace('"', ''))
    lenlans = len(lans.replace('*', '').replace('.',\
        '').replace(',', '').strip())
    if lengan != 0:
        # verbosity.append(lenlans/lengan)
        pass
```

```
        else:
            continue
        if key in Descr:
            verbosity['Descr'].append(lenlans/lengan)
        elif key in Quant:
            verbosity['Quant'].append(lenlans/lengan)
        else:
            verbosity['Binary'].append(lenlans/lengan)

#Kernel Density Estimation (Smooth Distribution)
sns.kdeplot(verbosity['Descr'], fill=True)
plt.title('Kernel Density Estimation for Descriptive')
plt.xlabel('verbosity')
plt.ylabel('Density')
plt.show()

plt.hist(verbosity['Descr'], bins=20, edgecolor='black')
plt.xlabel('verbosity')
plt.ylabel('Frequency')
plt.title('Verbosity Distribution for Descriptive')
plt.show()
```

Με την προαναφερόμενη συνάρτηση δημιουργούμε την εκτίμηση της κατανομής πιθανότητας πυκνότητας και την απεικονίζουμε διαγραμματικά με την βοήθεια της pyplot:

Τα διαγράμματα 4.4, 4.5, 4.6 παρουσιάζουν τα αποτελέσματα των παραπάνω εντολών για τους τρεις τύπους ερωτήσεων:

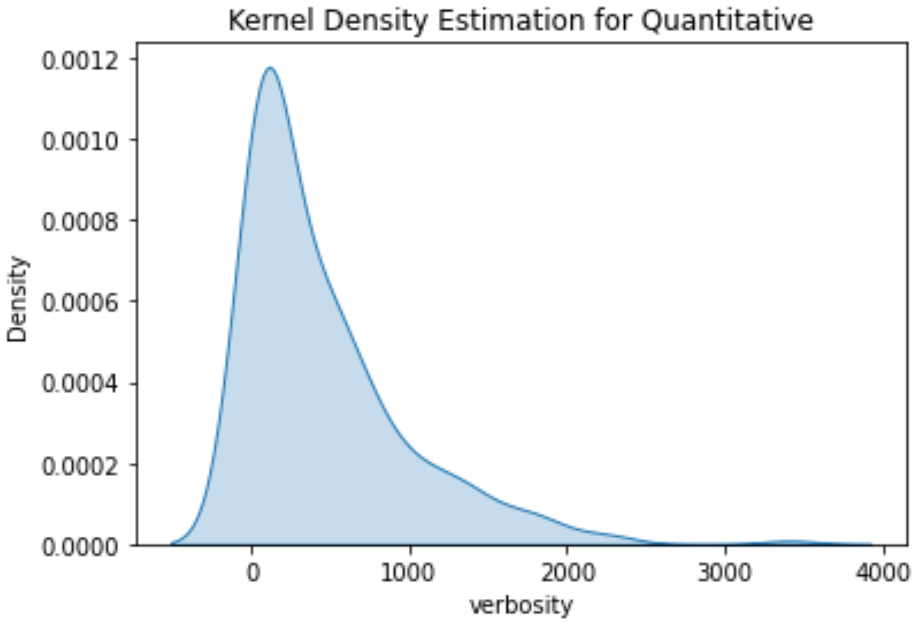

Figure 4.4: Κατανομή πυκνότητας πιθανότητας για ποσοτικές ερωτήσεις

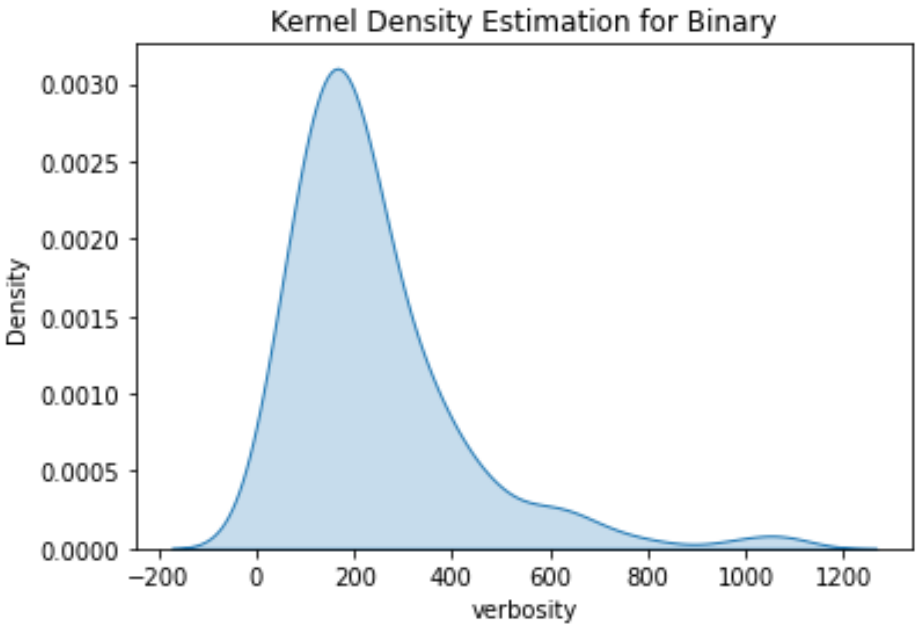

Figure 4.5: Κατανομή πυκνότητας πιθανότητας για δυαδικές ερωτήσεις

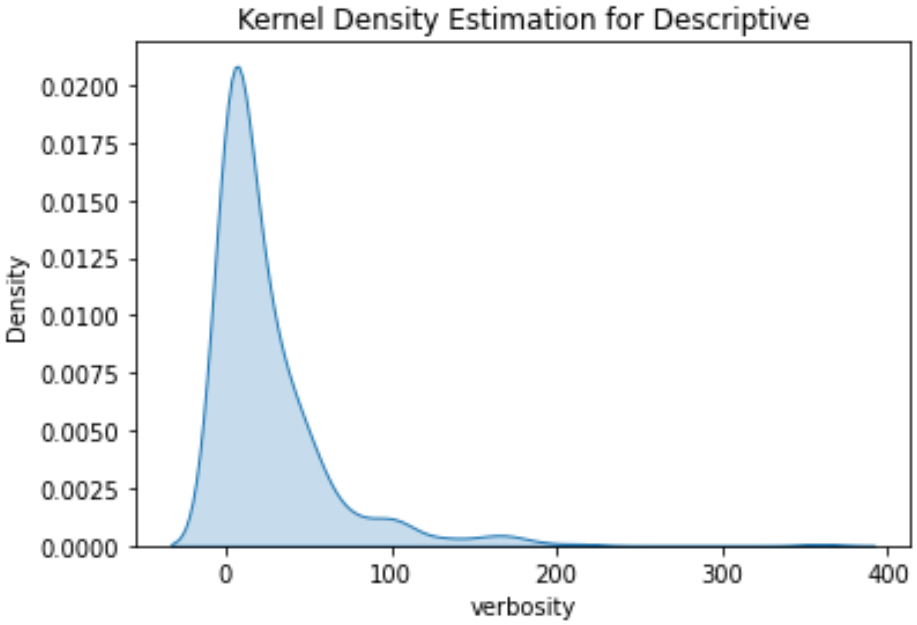

Figure 4.6: Κατανομή πυκνότητας πιθανότητας για περιγραφικές ερωτήσεις

Από τα παραπάνω διαγράμματα μπορούμε να διαπιστώσουμε ότι το verbosity είναι αρκετά μεγάλο και για τις τρεις κατηγορίες ερωτήσεων για το Llama 3.1.

Επιπρόσθετα, μπορούμε να δούμε ότι για τις ερωτήσεις Binary και Quantitative η μετρική verbosity είναι ιδιαίτερα αυξημένη σε σχέση με τις ερωτήσεις του τύπου Descriptive.

Συγκεκριμένα, η πλειοψηφία των τιμών verbosity για τις Descriptive ερωτήσεις συγκεντρώνεται μεταξύ των κάδων που αντιστοιχούν στα εύρη τιμών (0-20), (20-40) και (40-60).

Αντίθετα, για τις Binary ερωτήσεις οι περισσότερες τιμές συγκεντρώνονται στα μεταξύ του εύρους (0-400) και αντίστοιχα για τις ποσοτικές ερωτήσεις (0-1000).

Αυτό είναι εύλογο, μιας και οι απαντήσεις του γνωσιακού γράφου για τις δυαδικές και ποσοτικές ερωτήσεις είναι μια ή δύο. Αντίθετα, για του περιγραφικού τύπου, οι απαντήσεις

αναφοράς εμπεριέχουν πολλές λέξεις με αποτέλεσμα να αυξάνεται ο παρανομαστής της μετρικής verbosity και συνεπώς αυτή να μειώνεται.

4.3 Δημιουργία γράφων

Για τον σκοπό της δεύτερης πτυχής της παρούσας εργασίας δημιουργήθηκαν απλοί γράφοι για τους οποίους θα περιγράψουμε την γεωγραφική σχέση των κόμβων μεταξύ τους με βάση ένα μαθηματικό μοντέλο που θα το εισάγουμε στην ενότητα 5.

Πρώτα θα φτιάξουμε κάποιους μικρούς γράφους και θα ορίσουμε τους αντίστοιχους κόμβους τους. Θα διατυπώσουμε τον μαθηματικό φορμαλισμό που θα περιγραφούν οι κόμβοι και έπειτα θα ορίσουμε τις σχέσεις αυτών των κόμβων με τον εν λόγω συμβολισμό.

Όλα τα παραπάνω θα δοθούν ως prompt στα αντίστοιχα LLMs και θα ζητήσουμε βάση αυτών των μαθηματικών σχέσεων, το μοντέλο να συμπεράνει τις υπόλοιπες σχέσεις.

Αναλύοντας την απαντήσεις που θα μας δοθούν θα εξετάσουμε αν τα εν λόγω μοντέλα μπορούν να καταλάβουν την μαθηματικό φορμαλισμό των γεωγραφικών σχέσεων των αντικειμένων.

Παραδείγματα των γράφων που θα περιγράψουμε στα LLMs βάση του μοντέλου των cardinals παρουσιάζονται στην εικόνα 4.7.

Ο πρώτος γράφος έχει τρεις κόμβους για να εξεταστούν οι σχέσεις βόρεια και νότια και κατά πόσο τα LLMs μπορούν να τα κατανοήσουν βάση του φορμαλισμού που θα διατυπωθούν.

Figure 4.7: Πρώτος υπό εξέταση γράφος

Με την ίδια λογική θα δημιουργήσουμε και τον δεύτερο γράφο που θα αποτελείται από τρεις κόμβους προκειμένου να αξιολογηθεί η ικανότητα των γλωσσικών μοντέλων να κατανοούν τις σχέσεις ανατολικά και δυτικά.

Ο παρακάτω γράφος θα περιγραφτεί για να εξεταστεί η ικανότητα των LLMs να κατανοεί τις εναπομείνασες κατευθύνσεις.

Figure 4.8: Δεύτερος υπό εξέταση γράφος

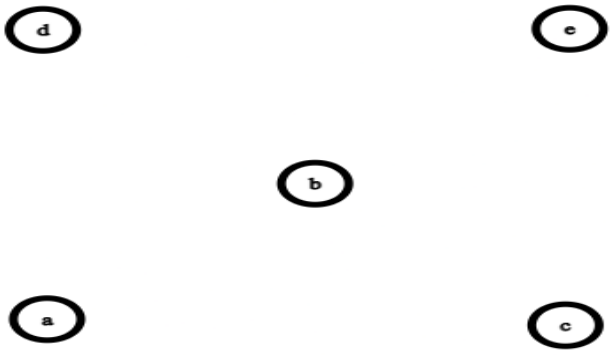

Figure 4.9: Τρίτος υπό εξέταση γράφος

5. ΜΕΘΟΔΟΛΟΓΊΕΣ ΑΞΙΟΛΌΓΗΣΗΣ ΤΩΝ ΜΕΓΆΛΩΝ ΓΛΩΣΣΙΚΏΝ ΜΟΝΤΈΛΩΝ

5.1 Περίληψη Ενότητας

Στην παρούσα ενότητα θα περιγράψουμε την βασική μεθοδολογία που ακολουθήθηκε για την αξιολόγηση των ερωτήσεων του benchmark GeoQuestions1089, καθώς και τα προβλήματα και τις λύσεις που προέκυψαν κατά την διάρκεια της εν λόγω εργασίας. Στην συνέχεια θα αναλύσουμε τον τρόπο εργασίας για την αξιολόγηση των LLM ως προς την ικανότητά τους να μπορούν να κατανοήσουν ή να συμπεράνουν γεω-τοπολογικές σχέσεις δεδομένου ενός γράφου και την σχέση των επιμέρων κόμβων τους.

5.2 Αξιολόγηση του GeoQuestions1089

Με την ραγδαία άνοδο των Μεγάλων Γλωσσικών Μοντέλων και την ευρεία χρήση τους από ακαδημαϊκούς, επαγγελματίες της αγοράς και καθημερινούς ανθρώπους είναι φυσικό να αναρωτηθούμε κατά πόσο μπορούν να απαντήσουν σωστά και σε τι βαθμό σε διάφορες ερωτήσεις.

Όπως αναφέραμε και στο κεφάλαιο 1 το εν λόγω benchmark χρησιμοποιείται για την αξιολόγηση αυτόματων συστημάτων ερωτήσεων-απαντήσεων αναφορικά με την γνώση που κατέχουν και κατά πόσο σωστές απαντήσεις δίνουν. Ουσιαστικά αποτελείται από ένα σύνολο ερωτήσεων γεωγραφικού και χωρικού χαρακτήρα, οι οποίες χωρίζονται σε κάποιες βασικές κατηγορίες.

Ένας από τους βασικούς στόχους της παρούσας εργασίας είναι η αξιολόγηση της επίδοσης των LLMs πάνω στο εν λόγω benchmark.

Στην ουσία, όσο πιο κοντά είναι οι απαντήσεις του LLM σε σχέση με αυτές του δείκτη αναφοράς, τόσο καλύτερη κρίνεται η επίδοση των μοντέλων. Συνεπώς, θα χρειαστούμε να δώσουμε τις ερωτήσεις ως prompt στα αντίστοιχα LLMs, να συλλέξουμε τις απαντήσεις τους και να δούμε πως θα μπορέσουμε να τις συγκρίνουμε με αυτές του δείκτη αναφοράς με τον πιο κατάλληλο τρόπο.

5.2.1 Εξερευνώντας τις ερωτήσεις του benchmark GeoQuestions1089.

Μια από τις πρώτες διεργασίες που κρίθηκε αναγκαίο να υλοποιηθεί ήταν η κατανόηση των ερωτήσεων GeoQuestions1089 και του τρόπου που είναι αποτυπωμένες οι αντίστοιχες απαντήσεις τους, συμπεριλαμβανομένου και της μορφή τους.

Παρακάτω παρατίθενται τα διαγράμματα 5.1, 5.2, 5.3, 5.4:

Στο GeoQuestions1089 εμπεριέχονται 1089 ερωτήσεις σε πρότυπο όμοιο με τα παραπάνω διαγράμματα. Από μια απλή ματιά είναι εμφανές ότι ο τύπος των απαντήσεων και η αντίστοιχη τροποποίηση διαφέρει από την μορφή των ερωτήσεων. Σύμφωνα με την δομή τους μπορούμε να χωρίσουμε τις ερωτήσεις του GeoQuestions1089 σε τέσσερις τύπους.

Πρώτες είναι οι ερωτήσεις που ξεκινούν με το “where” και αφορούν τοποθεσίες (locational). Όλες αυτές οι ερωτήσεις έχουν ως απάντηση δομές που αναφέρονται ως “POLYGONS” και “MULTIPOLYGONS”.

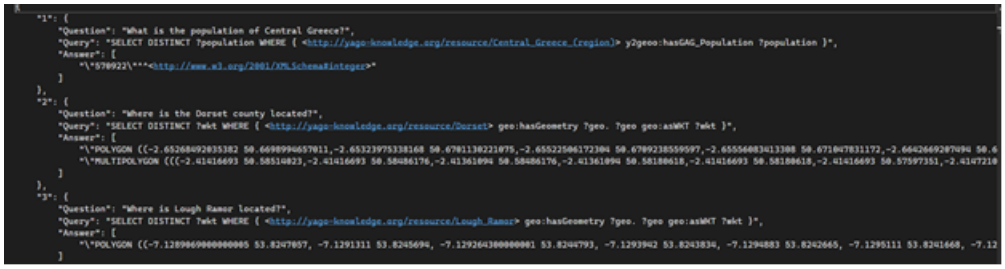

Figure 5.1: Παραδείγματα του benchmark

Figure 5.2: Παραδείγματα του benchmark

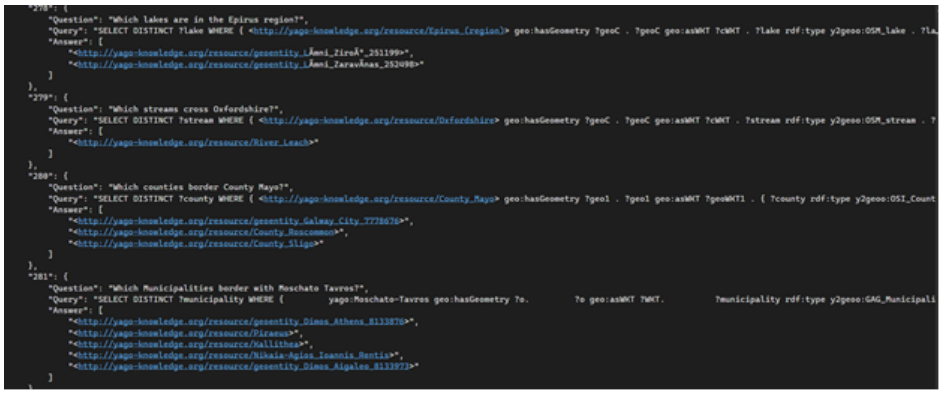

Figure 5.3: Παραδείγματα του benchmark

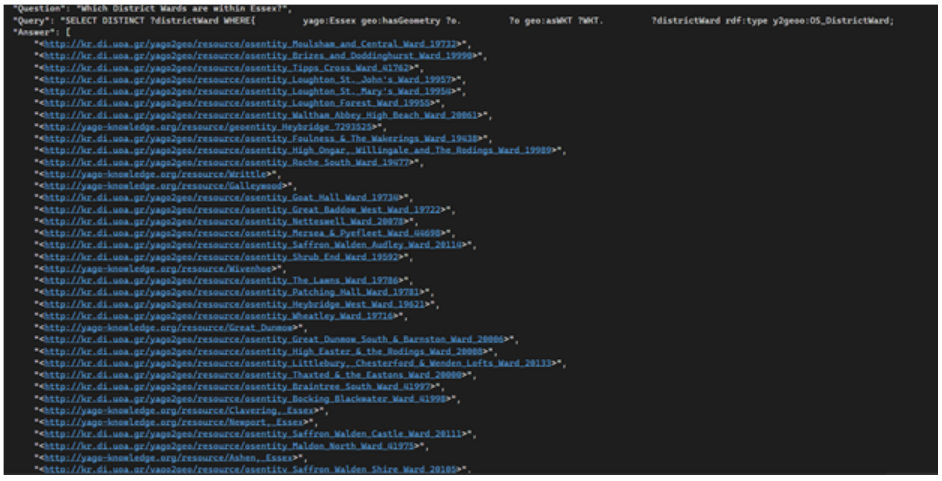

Figure 5.4: Παραδείγματα του benchmark

Οι συγκεκριμένες ερωτήσεις εξαιρούνται από την παρούσα ανάλυση καθώς είναι μια ιδιαίτερη δομή που δίνεται από τους γνωσιακούς γράφους, η οποία δεν βρίσκεται στις απαντήσεις των LLMs.

Οι συγκεκριμένες ερωτήσεις υπολογίστηκαν στις 89. Αφού εξαιρέσουμε τον προαναφερόμενο τύπο, μένει να χωρίσουμε τις υπόλοιπες σε τρεις σημαντικούς τύπους.

Η δεύτερος τύπος είναι οι ερωτήσεις που ξεκινούν με “Is”, “Are”, “Does”, “Do” και χαρακτηρίζονται ως Binary, διότι η απάντηση μπορεί να είναι ένα «Ναι» ή ένα «Όχι» ή μεταξύ δύο επιλογών.

Παράδειγμα τέτοιου τύπου ερωτήσεων είναι:

- ’Is Belfast closer to the capital of the Republic of Ireland or the capital of Scotland?’
- ’Is Southampton located south of Oxford?’
- ’Which is the bigger city in density, Athens or Dublin?’

Η πρώτη ερώτηση, που αποτελεί την μειοψηφία των ερωτήσεων αυτού του τύπου, έχει την ακόλουθη απάντηση σε μορφή URI: [’<http://yago-knowledge.org/resource/Dublin>’]. Από την άλλη πλευρά, η δεύτερη ερώτηση που είναι και η επικρατούσα μορφή αυτού του τύπου έχει ως απάντηση: [’1’].

Η παρουσία του ψηφίου ‘1’ υποδηλώνει θετική απάντηση, ενώ αν αυτή ήταν [’0’], τότε θα ήταν αρνητική. Μια εξαίρεση σε αυτού του τύπου τις απαντήσεις αποτελεί η ακόλουθη: ’Which is the bigger city in density, Athens or Dublin?’, η οποία θα έχει ως απάντηση την σωστή πόλη σε μια λίστα.

Ως τρίτος τύπος ερωτήσεων ορίζουμε τις περιγραφικές ερωτήσεις. Στην ουσία πρόκειται για ερωτήσεις ορισμού ή ερωτήσεις που έχουν ως απάντηση επιγραμματικές πληροφορίες, είτε αναφέρονται σε μια οντότητα, είτε σε πολλές.

Παραδείγματα αυτού του τύπου αποτελούν οι παρακάτω:

- 914: What is Skye?
- 1002: What is the longest river in the United States?
- 941: Find all the museums in Athens.
- 925: Where is Buckingham Palace located?
- 857: Which is the longest canal in England?

Οι αντίστοιχες απαντήσεις είναι στην εξής μορφή:

- 914: yago:OSM_island
- 1002: yago:Alum_Creek
- 925: yago:Westminster
- 857: yago:geoentity_Chelmer_and_Blackwater_Canal_2653269

Μπορούμε να δούμε ότι οι ερωτήσεις είναι μια λίστα με URIs τα οποία πέρα από κάποιες μικρές αλλαγές έχουν την ίδια δομή. Συνεπώς, μέσω αυτού του μοτίβου είναι εύκολο να βρούμε τον κατάλληλο αλγόριθμο για να εξάγουμε την σωστή πληροφορία που έχει νόημα για τα πλαίσια της ανάλυσης μας. Θα αναλύσουμε και παρακάτω τον αλγόριθμο που χρησιμοποιήθηκε για να εξάγουμε τις λέξεις που έχουν νόημα από τα αντίστοιχα URIs που μας έδωσε ο γράφος.

Τέλος, η τέταρτη κατηγορία ορίζεται ως αυτή των ποσοτικών και περιλαμβάνει ερωτήσεις που έχουν αριθμητικά δεδομένα ως απάντηση. Χαρακτηριστικά παραδείγματα αυτού του τύπου ερωτήσεων είναι:

- 4: What is the population of the Municipality of Moschato Tavros?
- 14: What is the total area of Cambridgeshire?
- 903: How many people live in LA?
- 901: What is the population of Athens?

Οι αντίστοιχες απαντήσεις των ερωτήσεων είναι οι εξής:

- $[''38116'' \wedge \wedge xsd:integer']$
- $[''0.40279582142829895'' \wedge \wedge xsd:integer,'' 0.6051532626152039'' \wedge \wedge xsd:double$
- $[''3833995'' \wedge \wedge xsd:integer]$
- $[''745514'' \wedge \wedge xsd:integer]$

5.2.2 Αλγόριθμος για κατηγοριοποίηση ερωτήσεων GeoQuestions1089

Παρατηρώντας τα ακόλουθα, είναι εύκολο να φτιάξουμε έναν αλγόριθμο που να συγκεντρώνει όλες τις ερωτήσεις και να τις χωρίζει στις ανωτέρω κατηγορίες εξαιρώντας αυτές που έχουν ως απάντηση POLYGONS ή MULTIPOLYGONS.

Συγκεκριμένα ο αλγόριθμος περιγράφεται ως εξής:

Πρώτα διαβάζουμε το json αρχείο που βρίσκονται οι ερωτήσεις των GeoQuestions1089. Μετά ορίζουμε την μεταβλητή TYPESBIN που είναι μια λίστα που περιέχει τις τέσσερις λέξεις που ξεκινούν οι ερωτήσεις του τύπου BINARY.

Συγκεκριμένα, αυτές είναι ”Is”, ”Do”, ”Does”, ”Are”. Ορίζουμε την συνάρτηση cleanDescr η οποία λαμβάνει το URI ενός στοιχείου που έχει δοθεί από τον γνωσιακό γράφο ως απάντηση.

Στην συνέχεια, χωρίζει την απάντηση βάση του χαρακτήρα whitespace (” ”). Αυτό γίνεται γιατί σε ορισμένες απαντήσεις των descriptives έχουμε την απάντηση μέσω του URI ακολουθούμενο από έναν αριθμό ή και ένα αντίστοιχο URI. Ως παραδείγματα της προαναφερόμενης περίπτωσης δίνονται τα στοιχεία που αντιστοιχούν σε κλειδιά 990, 991, 882.

- ’<http://yago-knowledge.org/resource/Little_Rock,_Arkansas> 191930’,
- ’<http://yago-knowledge.org/resource/Fort_Smith,_Arkansas> 80268’,

- ’<http://yago-knowledge.org/resource/Fayetteville,_Arkansas> 73372’

Μιας και μας ενδιαφέρει μόνο το πρώτο στοιχείο θα έχουμε την ακόλουθη εντολή el.split(” ”)[0]. Από την διερεύνηση του dataset προκύπτει ότι τα στοιχεία των descriptives απαντήσεων θα περιέχουν στο URI μετά την λέξη ”resource” μια από τις λέξεις osentity, geoentity, osientity, osnientity, gagentity, gadmentity ή απλά η απάντηση με τις λέξεις θα χωρίζονται από τον χαρακτήρα underscore.

Όταν δεν υπάρχει καμία από αυτές τις λέξεις χωρίζουμε το string σε υπομέρους στοιχεία, με βάση τον underscore χαρακτήρα.

- <http://yago-knowledge.org/resource/Little_Rock,_Arkansas> 191930
- <http://yago-knowledge.org/resource/Fort_Smith,_Arkansas> 80268
- <http://yago-knowledge.org/resource/Fayetteville,_Arkansas> 73372

Όταν υπάρχει η λέξη gadmentity κάνουμε το ίδιο, ήτοι αφαιρούμε την εν λόγω λέξη καθώς και τις 2 τελευταίες αφού είναι αριθμοί που δεν δίνουν κάποια ουσιαστική πληροφορία.

Για παράδειγμα έχουμε την εξής απάντηση:

- <http://kr.di.uoa.gr/yago2geo/resource/gadmentity_Wicklow_IRL.26_1>
- <http://kr.di.uoa.gr/yago2geo/resource/gadmentity_Meath_IRL.17_1>
- <http://kr.di.uoa.gr/yago2geo/resource/gadmentity_Kildare_IRL.9_1>

Αντίστοιχα όταν υπάρχει μια από τις υπόλοιπες λέξεις τότε κάνουμε το ίδιο απλά αφαιρούμε την πρώτη λέξη που είναι μια από τις προαναφερόμενες και την τελευταία που είναι αριθμός. Παραδείγματος χάριν,

- <http://yago-knowledge.org/resource/geoentity_Twin_River_5276583>

Στην εξεταζόμενη κατηγορία ερωτήσεων υπάρχει και η περίπτωση όπου το URI έχει την λέξη ”ontology” αντί για ”resource”. Σε αυτή την περίπτωση κάνουμε το ίδιο απλά αφαιρούμε το ακρώνυμο ”OSI” που βρίσκεται μπροστά από την απάντηση.

- <http://kr.di.uoa.gr/yago2geo/ontology/OSI_City_Council>

Αφού εξάγουμε την χρήσιμη πληροφορία, χρησιμοποιούμε την unidecode [1] συνάρτηση για να μετατρέψουμε γλωσσικούς χαρακτήρες στην αντίστοιχη λατινική τους μορφή και μετά αντικαθιστούμε τους χαρακτήρες ’%26’ και ’%27” με το ’&’ και ” ’ ” ανάλογα, όπου αυτό είναι απαραίτητο.

Ειδικότερα, η unidecode λαμβάνει χαρακτήρες σε μορφή Unicode και τα μετατρέπει σε AS-CII. Στις περισσότερες περιπτώσεις θα μπορούσαμε να απεικονίσουμε Unicode χαρακτήρες ως ”???” ή `\15BA\15A0\1610`, ως παράδειγμα δύο ακραίων περιπτώσεων. Παρόλα αυτά, δεν είναι χρήσιμη αυτή η προηγούμενη απεικόνιση χαρακτήρων για κάποιον που επιθυμεί να διαβάσει το κείμενο όπου εμπεριέχονται αυτοί.

[1] https://docs.python.org/3/howto/unicode.html

Αυτό που παρέχει η Unidecode είναι έναν ενδιάμεσο δρόμο: η συνάρτηση unidecode() λαμβάνει Unicode δεδομένα και τα αναπαριστά σε ASCII χαρακτήρες (δηλαδή την ευρέως αποδεκτή απεικόνιση χαρακτήρων μεταξύ 0x00 και 0x7F), όπου οι συμβιβασμοί που γίνονται μεταξύ δύο συνόλων από χαρακτήρες επιλέγονται έτσι ώστε να είναι εγγύτερα στο τι ένας χρήστης με ένα πληκτρολόγιο ρυθμισμένο στην αγγλική γλώσσα θα επέλεγε.

Χαρακτηριστικά παραδείγματα της εν λόγω συνάρτησης περιλαμβάνουν:

- unidecode(u'ko\u017eu\u0161\u010dek) —> 'kozuscek'
- unidecode(u"\u5317\u4EB0) —> 'Bei Jing '

Μετά τον ορισμό της προαναφερόμενης συνάρτησης, παίρνουμε κάθε κλειδί του dictionary GeoAns, και παίρνουμε την αντίστοιχη ερώτηση και απάντηση που δόθηκε από τον γνωσιακό γράφο, θέτοντας boolean μεταβλητές για κάθε τύπο ερωτήσεων ως False.

Επίσης, σπάμε την ερώτηση σε 2 strings με βάση το κόμμα για να συμπεριλάβουμε τις ερωτήσεις που δεν ξεκινάνε με ερωτηματικές λέξεις αλλά αντί αυτού βρίσκονται μετά το κόμμα. Αυτό γίνεται για την κατηγοριοποίηση των Binary ερωτήσεων, μιας και μόνο για αυτές θα χρησιμοποιήσουμε τις ερωτήσεις για να τις κατατάξουμε.

Ως παράδειγμα, η ερώτηση "In Greece, is there a village that is next to a nature reserve?", η λέξη "is" υποδηλώνει ότι η ερώτηση είναι Binary. Επομένως, μετά την εντολή subques = ques.split(", ") θα μας δώσει ['In Greece', 'is there a village that is next to a nature reserve?'].

Εν συνεχεία για οποιαδήποτε περίπτωση εκ των δύο (είτε η ερώτηση εμπεριέχει κόμμα είτε όχι), ψάχνουμε αν υπάρχει κάποια λέξη εκ των ["Is", "Do", "Does", "Are"] στην στις υποκείμενες προτάσεις.

Αν βρεθεί ότι η ερώτηση είναι όντως Binary, τότε θέτουμε την Boolean μεταβλητή IsitBin σε True και προσθέτουμε το αντίστοιχο κλειδί σε ένα νέο dictionary που αντιστοιχεί την τιμή "Binary" σε μια λίστα όπου εμπεριέχονται τα κλειδιά των ερωτήσεων που εντοπίζονται να είναι αυτού του τύπου.

Αναφορικά με τις ερωτήσεις Binary παρατηρούμε ότι οι απαντήσεις είναι στην μορφή ["0"] ή ["1"], όποτε τις αντικαθιστούμε με τις απαντήσεις ["No"] ή ["Yes"] αντίστοιχα.

Αν δεν είναι Binary, τότε ελέγχουμε αν στα URIs της απάντησης του γράφου υπάρχουν οι λέξεις "POLYGON" και "MULTIPOLYGON". Αν βρεθούν αυτές οι λέξεις τότε η ερώτηση χαρακτηρίζεται ως "Locational" και συνεχίζουμε στην επόμενη ερώτηση, μιας και αυτού του τύπου οι ερωτήσεις δεν θα αξιολογηθούν.

Εν συνεχεία, αν δεν είναι ούτε "Locational", τότε η ερώτηση θα χαρακτηρίζεται είτε ως "Quantitative" είτε ως "Descriptive" και η εν λόγω κατηγοριοποίηση θα γίνει βάση της δομής της απάντησης, όπως εμφανίζεται στο δείκτη αναφοράς.

Γενικά, αν υπάρχει ο χαρακτήρας "^^" στο URI, τότε η απάντηση είναι ποσοτική, δηλαδή εμπεριέχει αριθμό. Αν υπάρχει η λέξη "resource" τότε αυτή είναι απάντηση ερώτησης τύπου "Descriptive". Παρόλα αυτά, υπάρχουν και ερωτήσεις που εμπεριέχουν και τα δύο στοιχεία στο URI.

- <http://yago-knowledge.org/resource/Gregg_County,_Texas> "40" ^^ xsd:integer

Συνεπώς, αν εντοπιστούν τα 2 αυτά στοιχεία, χωρίζουμε το URI με βάση τον χαρακτήρα Whitespace και κρατάμε το πρώτο κομμάτι. Αν η λέξη "resource" βρίσκεται σε αυτό το

κομμάτι του URI τότε καλούμε την συνάρτηση cleanDescr() για να πάρουμε το χρήσιμο κομμάτι της απάντησης και θέτουμε την Boolean μεταβλητή IsitDescr σε True.

Διαφορετικά, η ερώτηση είναι ποσοτική και κρατάμε το κομμάτι μετά τους χαρακτήρες ”^^” αφαιρώντας τον χαρακτήρα ””’. Έπειτα, θέτουμε το IsitQuant ίσο με True για να προσθέσουμε το κλειδί στην λίστα που αντιστοιχεί στο πεδίο ”Quantitative” στην δομή dictionary cleanData.

Η επόμενη περίπτωση που μπορεί να συναντήσουμε είναι να το URI να εμπεριέχει μόνο τα σύμβολα ”^^”, οπότε είναι ποσοτική και κάνουμε την ίδια διαδικασία για να πάρουμε τον αριθμό ή τους αριθμούς και να προσθέσουμε το κλειδί στο σωστό πεδίο στο dictionary cleanData.

Ένα άλλο σενάριο είναι το URI να εμπεριέχει μια εκ των λέξεων ”resource” ή ”ontology” οπότε θα είναι απάντηση Descriptive ερώτησης. Άρα ακολουθούμε την προηγούμενη διαδικασία για να πάρουμε το χρήσιμο κομμάτι της απάντησης και για να προσθέσουμε το κλειδί της ερώτησης στο πεδίο ”Descriptive” του dictionary.

Μας μένουν δύο ιδιάζουσες περιπτώσεις στις μορφές που μπορούν να πάρουν οι ερωτήσεις. Η μία είναι η απάντηση της Descriptive ερώτησης να μην δίνεται σε URI όπως στις προηγούμενες περιπτώσεις, άρα να μην έχει κάποιες από τις λέξεις ”resource” ή ”ontology”.

- Question’: ’Which city has higher population, Athens or New York?
- ’Answer’: [’New York’]

Η άλλη αφορά ποσοτικές ερωτήσεις στις οποίες ο γράφος δεν έδωσε URI, αλλά έναν αριθμό και μια συντομογραφία που υποδηλώνει τετραγωνικά μέτρα ή κάποιο αντίστοιχο μέτρο. Παραδείγματος χάριν, η απάντηση μπορεί να είναι ’98.0#m’.

Συνεπώς, αν φτάσει στην τελευταία περίπτωση τότε πρώτα θα κοιτάξουμε αν υπάρχουν οι λέξεις ”Which” ή ”which” στην ερώτηση. Αν αυτές υπάρχουν, τότε θα είναι ”Descriptive” και θα ακολουθήσουμε παρόμοια διαδικασία εξαγωγής πληροφορίας και θα θέσουμε σε True την κατάλληλη μεταβλητή για να την εκχωρήσουμε στην λίστα που αντιστοιχεί στο ”Descriptive”.

Αν σε αυτή την τελευταία περίπτωση δεν υπάρχουν οι λέξεις ”Which” και ”which”, τότε χαρακτηρίζουμε την ερώτηση ως Quantitative και αφαιρούμε ότι υπάρχει μετά την ”#” για να πάρουμε τον καθαρό αριθμό και θέτουμε την Boolean μεταβλητή ως IsitQuant σε True για να εκχωρήσουμε το κλειδί στην σωστή λίστα στην δομή dictionary cleanData.

Το σύνολο δεδομένων που φορτώσαμε και περιλαμβάνει όλες τις ερωτήσεις από το benchmark του GeoQuestions1089 καθώς και τις απαντήσεις του γνωσιακού γράφου, έχει κάποιες ερωτήσεις που έχουν απαντήσεις same-as.

Πρόκειται για ερωτήσεις που είναι διπλές ή είναι διατυπωμένες με διαφορετικό τρόπο αλλά έχουν τις ίδιες απαντήσεις. Αυτές αντιστοιχούν στα κλειδιά 1018 έως 1089. Ο λόγος ύπαρξης αυτών των ερωτήσεων είναι για να αυξηθεί η ευρωστία (robustness) των συστημάτων που οι απαντήσεις τους αξιολογούνται πάνω στον δείκτη αναφοράς.

Στα πλαίσια αυτής της εργασίας οι ερωτήσεις αυτές εξαιρέθηκαν καθώς δεν προσδίδουν κάτι ιδιαίτερο ως προς την αξιολόγηση των μεγάλων γλωσσικών δεδομένων.

Ο παραπάνω αλγόριθμος είχε ως αποτέλεσμα να κατατάξει όλες τις ερωτήσεις σε τρεις κατηγορίες και να τις εκχωρήσει στο dictionary που αρχικοποιήθηκε με όνομα cleanData.

Υπάρχουν όμως δύο ερωτήσεις που δεν κατηγοριοποιήθηκαν σωστά. Πρόκειται για τις ερωτήσεις που αντιστοιχούν στα κλειδιά ’907’ και ’1005’. Αυτές είναι οι εξής:

- ’Which is the length of River Liffey?’
- ’Which settlement has the biggest population in France?’

με απαντήσεις:

- ’125000.0’
- ’7127’

Αυτές εμπίπτουν στην τελευταία περίπτωση του αλγορίθμου (else), όπου δεν είναι Binary και οι απαντήσεις δεν είναι σε μορφή URI. Επειδή είναι ποσοτικές ερωτήσεις και έχουν την λέξη ”Which” στην ερώτηση θα κατηγοριοποιηθούν λανθασμένα ως Descriptive ενώ είναι Quantitative.

Αυτό που κάνουμε είναι να αφαιρέσουμε αυτά τα κλειδιά από την λίστα με πεδίο Descriptive στο dictionary που δουλεύουμε και να τα προσθέσουμε στην λίστα που αντιστοιχεί στο πεδίο Quantitative. Οι εντολές που κάνουν την προαναφερόμενη διεργασία είναι: cleanData[’Descr’].remove(key) cleanData[’Quant’].append(key) για keys = 907, 1005.

Μετά την ανωτέρω κατηγοριοποίηση των κλειδιών, πρέπει να χωρίσουμε τις υπό εξέταση ερωτήσεις σε κατηγορίες όπως αυτές διαμορφώθηκαν βάση του συνόλου δεδομένων που δουλεύουμε. Ειδικότερα, οι ερωτήσεις κατανέμονται στις κατηγορίες ”A”, ”B”, ”C”, ”D”, ”E”, ”F”, ”H”, ”I”.

Θα χρειαστούμε αυτές τις κατηγορίες γιατί θα παρουσιάσουμε την αξιολόγηση των ερωτήσεων βάση αυτών των κατηγοριών καθώς και βάση των τύπων ερωτήσεων που κάναμε σε προηγούμενη ανάλυση, δηλαδή ”Binary”, ”Descriptive” και ”Quantitative”.

Το εύρος των κλειδιών των ερωτήσεων που ανήκει σε μια συγκεκριμένη κατηγορία βρίσκεται από ένα άλλο αρχείο, όπου συγκεντρώνει τις ερωτήσεις μαζί με το γράμμα της κατηγορίας.

Συνεπώς, αφού ελέγχθηκαν όλες οι κατηγορίες ερωτήσεων συγκεντρώσαμε για κάθε κατηγορία όλα τα κλειδιά που αντιστοιχούν σε αυτήν και τα αποθηκεύσαμε σε ένα dictionary.

Το σύνολο των κλειδιών της κάθε κατηγορίας δίνεται παρακάτω:

- A: 1-142 και 895-925
- B: 144-277 και 927-931
- C: 276-416 και 419-432 και 932-954
- D: 433-452 και 588-590 και 612
- E: 433-587 και 955
- F: 591-611 και 956-958
- I: 873-894 και 1015-1017

Μετά την δημιουργία του dictionary που αντιστοιχεί τα παραπάνω σύνολα κλειδιών στις αντίστοιχες ομάδες αποθηκεύουμε την εν λόγω δομή σε ένα json αρχείο με όνομα ”keysGeoCat”.

5.3 Μεθοδολογία αξιολόγησης ερωτήσεων

Στην συγκεκριμένη ενότητα θα περιγράψουμε την βασική διαδικασία που ακολουθήθηκε προκειμένου να αξιολογηθεί η απάντηση που δόθηκε από το LLM ως σωστή, λάθος ή μερικώς σωστή. Η ορθότητα της απάντησης του LLM κρίνεται με βάση της αντίστοιχης του benchmark.

Για παράδειγμα, στην ερώτηση ”Is Crete an island?” θα θεωρηθεί η απάντηση του LLM ως σωστή μόνο αν υπάρχει θετική απάντηση (”yes” ή τα αντίστοιχα συνώνυμα) αφού η απάντηση του γράφου είναι καταφατική.

Επιπρόσθετα, στην ερώτηση ”Which is the capital of Ireland?” αν ο γράφος έδωσε την απάντηση ”Dublin”, τότε για να θεωρηθεί ως σωστή η απάντηση του LLM θα πρέπει να έχει απαντήσει το ίδιο, ανεξάρτητα με το πόσο περιττολογία εμπεριείχε.

Ακριβώς η ίδια φιλοσοφία ισχύει και για τις ποσοτικές ερωτήσεις τύπου ”What is the population of USA?”. Δεδομένου των αριθμών που δίνει η απάντηση του γνωσιακού γράφου θα θεωρήσουμε αν αυτή του LLM είναι σωστή, μερικώς σωστή ή λάθος.

Το πρώτο πράγμα που ελέγχουμε στις απαντήσεις του LLM είναι αν εμπεριέχεται η πρόταση ”I couldn’t find any information” στην απάντηση του LLM. Αυτό γίνεται για όλες τις ερωτήσεις ανεξαρτήτου του τύπου που ανήκουν.

Αν εμφανίζεται αυτή η φράση τότε αυτό σημαίνει ότι το LLM δεν κατάφερε να απαντήσει στην ερώτηση που του τέθηκε. Αν δεν υπάρχει εξετάζουμε σε ποιόν τύπο ανήκει.

Για τους σκοπούς αυτής της ανάλυσης χρησιμοποιήθηκαν διάφορες βιβλιοθήκες ή modules για να κάνουμε καλύτερη γλωσσολογική ανάλυση στις απαντήσεις που πρέπει να αξιολογήσουμε.

Πιο συγκεκριμένα, εφαρμόζουμε tokenization [14] στο string της απάντησης μέσω της συνάρτησης word_tokenize. Δεδομένου δηλαδή, ενός string όπως lans = ’the population of Greece has dwindled over the years.’, η αντίστοιχη κλήση της εντολής tokens = word_tokenize(lans) θα επιστρέψει μια λίστα με όλες τις λέξεις της υποκείμενης πρότασης. Επομένως, θα πάρουμε την λίστα [’the’, ’population’, ’of’, ’greece’, ’has’, ’dwindled’, ’over’, ’the’, ’years’, ’.’].

Έχοντας κάνει την παραπάνω διεργασία, μπορούμε να χρησιμοποιήσουμε την συνάρτηση pos_tag της nltk [2] η οποία δίνει για κάθε λέξη ένα tuple με αυτήν και μια ετικέτα που αναφέρει το μέρος του λόγου της λέξης.

Στο παράδειγμά μας, η κλήση της pos_tag(tokens) μας δίνει [(’the’, ’DT’), (’population’, ’NN’), (’of’, ’IN’), (’greece’, ’NN’), (’has’, ’VBZ’),(’dwindled’, ’VBN’), (’over’, ’IN’), (’the’, ’DT’), (’years’, ’NNS’), (’.’, ’.’)].

Οι ετικέτες που ξεκινούν με ’N’ δηλώνουν ουσιαστικά, ενώ αυτές που ξεκινούν με ’V’ είναι ρήματα. Αντίστοιχα, τα άρθρα έχουν ’DT’ ετικέτες ενώ οι αριθμοί ’CD’, ακόμα και όταν είναι εκφρασμένα σε φυσική γλώσσα. Δίνεται ως παράδειγμα η παρακάτω λίστα με tuples:

[(’the’, ’DT’), (’population’, ’NN’), (’of’, ’IN’), (’greece’, ’NN’), (’is’, ’VBZ’), (’close’, ’RB’), (’to’, ’TO’), (’10’, ’CD’), (’million’, ’CD’), (’.’, ’.’)].

Επιπρόσθετα, χρησιμοποιούμε από την βιβλιοθήκη word2number [3] το w2n. Το συγκεκριμένο module μας βοηθάει να μετατρέψουμε αριθμούς που είναι εκφρασμένοι σε φυσική γλώσσα

[2] https://www.nltk.org/

[3] https://pypi.org/project/word2number/

στην αντίστοιχη αριθμητική τους αναπαράσταση. Αναφέρονται τα παρακάτω παραδείγματα:

- print w2n.word_to_num(”two million three thousand nine hundred and eighty four”) —> 2003984
- print(w2n.word_to_num(’two point three’)) —> 2.3
- print(w2n.word_to_num(’112’)) —> 112
- print(w2n.word_to_num(’point one’)) —> 0.1
- print(w2n.word_to_num(’one hundred thirty-five’)) —> 135

Επίσης, μέσω της κλήσης των ακόλουθων εντολών nltk.download(’stopwords’), from nltk.corpus import stopwords, stop_words = set(stopwords.words(’english’)), stop_words.remove(”no”), stop_words.remove(”not”), παίρνουμε όλα τα stop words σε ένα σύνολο και από αυτό αφαιρούμε τις λέξεις ”no” και ”not” καθώς τα δύο τελευταία θα μας χρησιμεύσουν στην ανάλυση των Binary ερωτήσεων.

Τέλος, θα χρησιμοποιήσουμε στην σύγκριση των ερωτήσεων την τεχνική lemmatization [14]. Αυτός ο μετασχηματισμός στην ουσία παίρνει μια λέξη και την φέρνει σε μια αρχική μορφή χρησιμοποιώντας την ρίζα της.

Για παράδειγμα, οι λέξεις ”church” και ”churches” έχουν το ίδιο νόημα και αναφέρονται στο ίδιο αντικείμενο. Παρόλα αυτά, τα υπολογιστικά συστήματα δεν μπορούν να καταλάβουν ότι αναφέρεται στην ίδια οντότητα.

Όταν λοιπόν εφαρμόζουμε lemmatization και οι δύο λέξεις θα αποκτήσουν την ίδια μορφή και εν τέλει θα αναγνωριστούν ως όμοιες έννοιες. Συγκεκριμένα από την μέθοδο lemmatize έχουμε τα παρακάτω αποτελέσματα:

- lemmatizer.lemmatize(”churches”) —>’church’
- lemmatizer.lemmatize(”church”) —> ’church’
- lemmatizer.lemmatize(”are”, pos=”v”) —> ’be’
- lemmatizer.lemmatize(”is”, pos=”v”) —> ’be’
- lemmatizer.lemmatize(”does”, pos=”v”) —> ’do’
- lemmatizer.lemmatize(”do”, pos=”v”) —> ’do’

5.3.1 Αλγόριθμος αξιολόγησης για Binary ερωτήσεις

Αν η ερώτηση είναι Binary μορφής, τότε θα επεξεργαστούμε την απάντηση του LLM με την συνάρτηση preprocessBinLlmAns. Ουσιαστικά, το πρώτο που κάνουμε είναι να αντιστοιχήσουμε λέξεις όπως isn’t, hasn’t, aren’t, can’t, couldn’t, didn’t, doesn’t ή παρόμοιες με is not, has not, are not, can not, could not, did not, does not κ.ο.κ.

Με άλλα λόγια θέλουμε να εξάγουμε την άρνηση not, διότι αυτή η λέξη δηλώνει αρνητική απάντηση. Για την επίτευξη αυτού του σκοπού φτιάχνουμε ένα dictionary που αντιστοιχεί κάθε λέξη τύπου ”isn’t” στην πλήρη μορφή του ”is not”.

Μέσω της εντολής contractions_re = re.compile('(%s)' % '|'.join(CONTRACTIONS.keys())) και την βοήθεια της βιβλιοθήκης re φτιάχνουμε το κατάλληλο μοτίβο (re.Pattern), ήτοι re.compile(r"(isn't|hasn't|aren't|can't|couldn't|didn't|doesn't|don't|hadn't|haven't)", re.UNICODE).

Έπειτα καλούμε την συνάρτηση expand_contractions, όπου λαμβάνει το προαναφερόμενο dictionary και για κάθε λέξη που εντοπίζει από το παραπάνω μοτίβο, το αντικαθιστά με την πλήρη μορφή της. Παραδείγματος χάριν, ας θεωρήσουμε την απάντηση ans_test = "Athens isn't in Germany.". Καλώντας την εντολή expand_contractions(ans_test) παίρνουμε 'Athens is not in Germany.'.

Στην συνέχεια, μετατρέπουμε όλους τους χαρακτήρες της υποκείμενης απάντησης σε πεζά γράμματα, αφαιρούμε σημεία στίξης (punctuations) και χρησιμοποιούμε την συνάρτηση word_tokenize της nltk βιβλιοθήκης για να πάρουμε μια λίστα που κάθε στοιχείο της είναι μια μεμονωμένη λέξη από την εξεταζόμενη ερώτηση.

Έπειτα, μέσω list comprehension αφαιρούμε όλα τα stop words, όπως έχουν τα τελευταία οριστεί από την βιβλιοθήκη nltk. Παράλληλα, στο script "EvaluateAns.py" έχουμε φορτώσει το module "metricFuncs4Eval.py" το οποίο εμπεριέχει τρεις συναρτήσεις, μια για κάθε τύπο ερωτήσεων.

Θα χρησιμοποιήσουμε την πρώτη συνάρτηση από το εν λόγω module metricBin, όπου λαμβάνει την απάντηση του LLM και του γράφου ως ορίσματα. Οι απαντήσεις είναι στην μορφή λίστας και εμπεριέχει τις λέξεις της κάθε απάντησης.

Με τον ορισμό της μεταβλητής synonyms = "yes": ["correct", "true", "indeed"], "no": ["false", "wrong", 'not'] θα επιχειρήσουμε να βρούμε τα συνώνυμα των θετικών και αρνητικών απαντήσεων. Δηλαδή, αν η απάντηση αναφοράς είναι θετική και παρόλα αυτά η αντίστοιχη του LLM δεν έχει την λέξη "yes" θα αναζητήσουμε τις λέξεις ["correct", "true", "indeed"]. Αν αυτές βρεθούν στην απάντηση του μοντέλου, τότε αυτή χαρακτηρίζεται ως θετική, ενώ αντίστοιχα αν βρεθούν οι λέξεις που αντιστοιχίζονται με το "no" , στην περίπτωση που δεν υπάρχει αυτή η λέξη στην απάντηση του μοντέλου, τότε αυτή χαρακτηρίζεται ως αρνητική.

Μετά τον ορισμό της μεταβλητής synonyms, αρχικοποιούμε μια λίστα και εκχωρούμε την απάντηση αναφοράς. Ανάλογα με το ποια είναι αυτή, "yes" ή "no", προσθέτουμε σε αυτήν και τα αντίστοιχα συνώνυμα της. Αν η απάντηση του γράφου δεν είναι κάποια από αυτές τις λέξεις, τότε δεν προσθέτουμε κανένα συνώνυμο στην λίστα.

Το επόμενο βήμα είναι να αρχικοποιήσουμε μια μεταβλητή με το όνομα count και να της δώσουμε την τιμή "0". Για κάθε λέξη της λίστας που φτιάχτηκε προηγουμένως, ψάχνουμε να βρούμε αν υπάρχει στην απάντηση που έδωσε το LLM. Αν βρεθεί κάποια από αυτές, η τιμή του count γίνεται "1" και την επιστέφουμε μέσω της συνάρτησης.

Συνεπώς, αν η συνάρτηση επιστρέψει την τιμή "0", τότε η απάντηση του LLM θεωρείται λανθασμένη. Αντίστροφα, αν από την συνάρτηση metricBin πάρουμε την τιμή "1", τότε η απάντηση κρίνεται ως ορθή.

Παρακάτω, δίνεται η συνάρτηση σε μορφή κώδικα:

```
def metricBin(LlmAns, GeoAns):

    """
    LlmAns: is a list of preprocessed words (only the first sentence).
    For example, ["yes", "the", "capital", "of", "greece", "is", "athens"]
    GeoAns: is a list containing "yes", "no" or another word

    Returns:
```

```
0: in case the LlmAns is false
1: if the LlmAns is correct
"""

#Other possible answers that could align with the reference answer
synonyms = {"yes": ["correct", "true", "indeed"],
                    "no": ["false", "wrong", 'not']}
wordsGeo = list()
wordsGeo.append(GeoAns[0])
try:
    wordsGeo.extend(synonyms[GeoAns[0]])
except:
    pass
    count = 0
    for word in wordsGeo:

        if word in LlmAns:
            count = 1
            break

return count
```

5.3.2 Αλγόριθμος αξιολόγησης Ποσοτικών ερωτήσεων

Από την άλλη, αν η ερώτηση είναι ποσοτικής μορφής, θα προ-επεξεργαστούμε μόνο την απάντηση του μοντέλου LLM. Στην ουσία, το επόμενο βήμα θα είναι να κρατήσουμε όλους τους αριθμούς της απάντησης του LLM εκφρασμένα σε αριθμητικές τιμές.

Αρχικά, θα κάνουμε ότι και πριν, αλλά θα μετατρέψουμε όλους τους χαρακτήρες σε μικρούς, θα κάνουμε tokenize τις λέξεις και θα εφαρμόσουμε pos_tagging.

Μια διαφορά για τις ερωτήσεις αυτής της περίπτωσης είναι ότι θα κρατήσουμε μόνο τα στοιχεία αυτά που έχουν την ετικέτα ”CD”, καθώς δηλώνουν αριθμούς. Μετέπειτα, θα αλλάξουμε τυχόν χαρακτήρα ’,’ σε ’.’ για να μπορούν να αποδοθούν ως floats. Για παράδειγμα, το ’2,34’ θα γίνει ’2.34’ και η συνάρτηση της python float() θα μπορεί να το μετατρέψει σε αριθμητικό δεδομένο.

Εδώ θα χρειαστεί να αναφέρουμε ξανά ότι ”CD” ετικέτα μπορεί να έχει και strings όπως το ”million”. Δηλαδή, λέξεις που αναφέρονται σε αριθμούς αλλά σε φυσική γλώσσα. Συνεπώς, κάθε στοιχείο που έχει ετικέτα ”CD” θα το μετατρέψουμε σε αριθμητικό στοιχείο με την συνάρτηση float() της Python.

Αν αυτό παρουσιάσει κάποιο σφάλμα, δηλαδή στην περίπτωση που έχουμε αριθμούς εκφρασμένους σε φυσική γλώσσα, τότε θα χρησιμοποιήσουμε το module w2n που αναφέρθηκε στην παρούσα ενότητα. Αν αυτό το στοιχείο αναφέρεται σε κανονικό αριθμό που αναφέρθηκε προηγουμένως, θα μετατρέψουμε τα δύο αυτά στοιχεία σε έναν αριθμό.

Για παράδειγμα, αν έχουμε ”10 million”, ο αλγόριθμος θα κρατήσει το ”10” που είναι αριθμός, θα μετατρέψει το ”million” σε ”1000000”, θα πολλαπλασιάσει τους δύο παραγόμενους αριθμούς και θα αποθηκεύσει στην τελική λίστα τον αριθμό ”10000000”.

Ο κώδικας της παραπάνω διαδικασίας περιγράφεται παρακάτω:

```
def preprocessQuantLlmAns(ans):
    """
    ans is string
    We tokenize and then apply pos tagging to all the words.
    Then we retain only the tokens that have a "CD" tag,
    we replace for these tokens
    ',' with '.' in order to be able to convert them to floats
    and set the flag to true
    to denote that we have encountered a number.
    If the next token is not a number or it is one but
    expressed in natural language (millions) we will multiply
    it with the exact previous encountered number.
    If the previous token was not a number we simply
    convert it to a number: million --> 1000000
    """
    ans = ans.lower()
    tokenText = word_tokenize(ans)
    tags = pos_tag(tokenText)
    lsTarget = []
    flag = False
    for ek in tags:
        if ek[1] == "CD":
            num = ek[0].replace(',', '.')
            try:
                lsTarget.append(float(num))
                flag = True
            except ValueError:
                # print(num)
                try:
                    num = w2n.word_to_num(num)
                    if flag:
                        lsTarget.append(lsTarget.pop() * num)
                        flag = False
                    else:
                        lsTarget.append(num)
                        flag = True
                except ValueError:
                        flag = False
        else:
            flag = False

    return lsTarget
```

Η συνάρτηση που θα υπολογίσει κατά πόσο σωστή είναι απάντηση του LLM ονομάζεται metricQuant. Στην ουσία για κάθε αριθμό (όπου υπάρχει) της απάντησης αναφοράς, τον συγκρίνουμε με αυτούς που έχουν βρεθεί στην απάντηση του LLM. Αν η απόλυτη ποσοστιαία διαφορά ενός ζευγαριού είναι μεταξύ 0% και 5%, τότε θεωρούμε ότι το μοντέλο βρήκε τον υπό εξέταση αριθμό.

Έπειτα συγκρίνουμε πόσους αριθμούς βρήκε το LLM από αυτούς που είχε η απάντηση

του benchmark και αναλόγως χαρακτηρίζουμε αυτήν του LLM ως σωστή, μερικώς σωστή ή λάθος.

Παρακάτω παρατίθεται ο κώδικας σε Python.

```
def metricQuant(llmAns, geoAns):
    """
    geoAns is a list of numbers
    """
    if len(llmAns) == 0 or len(geoAns) == 0:
        return 0
    AllMindis = list()
    for num in geoAns:
        dis = list()
        for num2 in llmAns:
            try:
                # print(num)
                # print(num2)
                if float(num) != 0:
                    eps = abs(float(num) - float(num2))/float(num)
                else:
                    eps = abs(float(num) + 1 - \
                    (float(num2) + 1))/(float(num) + 1)
                dis.append(eps)
            except ValueError:
                dis.append(100000000)
        AllMindis.append(min(dis))

    count = 0
    for val in AllMindis:
        if val < 0.05 :
            count +=1

    return count/len(geoAns)#, AllMindis
```

5.3.3 Αλγόριθμος αξιολόγησης για Descriptive ερωτήσεις

Για το συγκεκριμένο είδος ερωτήσεων ακολουθήσαμε δύο τρόπους ως προς την αξιολόγηση τους, καθότι οι απαντήσεις του γράφου δεν είναι μονολεκτικές ή με αριθμητικά δεδομένα αλλά εμπεριέχουν συνήθως πολλές λέξεις.

Ανεξάρτητα την μέθοδο που ακολουθούμε η προ επεξεργασία των γλωσσικών δεδομένων που εμπεριέχονται στην απάντηση του δείκτη αναφοράς ή στην αντίστοιχη του LLM είναι ίδια.

Αρχικά, για την περίπτωση της απάντησης του LLM, μετατρέπουμε όλα τα γράμματα σε μικρά με την εντολή ans = ans.lower() και αντικαθιστούμε όλα τα σημεία στίξης με τον χαρακτήρα whitespace. Μετέπειτα, όταν βρίσκουμε τους χαρακτήρες "'s" σε αυτή την σειρά τους αφαιρούμε, δηλαδή η πρόταση "George's store is the best" θα μετατραπεί σε "George store is the best". Τα παραπάνω γίνονται με την εντολή:

```
procText = re.sub(r'[^\w\s]', '␣', ans.replace("'s", ""))
```

Το επόμενο βήμα είναι να κάνουμε tokinize την απάντηση, tokenText = word_tokenize(procText), και με list comprehension να αφαιρέσουμε τα stop words με τον εξής τρόπο: flt_text = [word for word in tokenText if word not in stop_words]. Αφού πάρουμε την προαναφερόμενη λίστα με τις εναπομείνουσες λέξεις, θα εφαρμόσουμε pos tagging, tags = pos_tag(flt_text) και θα κρατήσουμε αυτές που έχουν τις ακόλουθες ετικέτες : ”UH”, ”CD”, ”RB”, ”JJ”, ”DT”, ”VBP” καθώς και όλα αυτά που ξεκινούν με ”N”, ήτοι τα nouns.

Οι ανωτέρω ετικέτες αναφέρονται στα εξής μέρη του λόγου:

- UH (Interjection): επιφώνηματα
- CD (Cardinal Number): νούμερα (συμπεριλαμβανομένου και αυτά που είναι εκφρασμένα σε φυσική γλώσσα)
- RB (Adverb): επιρρήματα
- JJ (Adjective): επίθετα
- DT (Determiner): άρθρα και αντωνυμίες
- VBP (Verb, Non-3rd Person Singular Present): ρήματα

Τέλος, κάνουμε lemmatization για κάθε λέξη που κρατήσαμε με την εντολή convAns = [lemmatizer.lemmatize(x[0], pos=”n”).lower() for x in lsTarget] και επιστρέφουμε μια λίστα με τις τροποποιημένες λέξεις.

Παρατίθεται η ανωτέρω διαδικασία σε κώδικα Python.

```
def preprocessDescLlmAns(ans):
    """
    ans here is a string
    We tokenize the words and apply POS tag for each word.
    Then we keep all the nouns and some other categories.
    Finally, we lemmatize all the words and create
    a list of the final transformed words.
    """
    ans = ans.lower()
    procText = re.sub(r'[^\w\s]', '␣', ans.replace("'s", ""))
    tokenText = word_tokenize(procText)
    flt_text = [word for word in tokenText\
        if word not in stop_words]
    #possible solution [word for word in
    #tokenText if word not in stop_words and len(word) > 1]
    tags = pos_tag(flt_text)
    evalSet = {"UH", "CD", "RB", "JJ", "DT", "VBP"}
    lsTarget = []

    for ek in tags:
        if ek[1][0] == "N" or ek[1] in evalSet:
            lsTarget.append((ek[0], ek[1]))
```

```
    #a more correct way would be to pos='v' for verbs
    convAns = [lemmatizer.lemmatize(x[0], pos="n").lower()\
    for x in lsTarget]

    return convAns
```

Για την περίπτωση της απάντησης του Geoquestions1089, ακολουθούμε την ίδια διαδικασία με την διαφορά ότι στην αρχή χρησιμοποιούμε ans = ' '.join(ans) για να μετατρέψουμε την λίστα με τις λέξεις που έχουμε πάρει ως απάντηση σε ένα string. Το ακόλουθο βήμα είναι να αφαιρέσουμε τα σημεία στίξης και τους χαρακτήρες "'s" και να εφαρμόσουμε tokenize.

Σε αυτήν την περίπτωση πριν κάνουμε lemmatization, αφαιρούμε τις διπλές λέξεις από την απάντηση αναφοράς με την εντόλη unique = list(dict.fromkeys(tokenText)) και αποθηκεύουμε το αποτέλεσμα σε μια νέα λίστα.

Τέλος, υλοποιούμε τα τελευταία βήματα, ήτοι κάνουμε pos tagging και lemmatization, χωρίς να αφαιρούμε κάποιες λέξεις βάση κάποιου συγκεκριμένου μέρους του λόγου.

```
    def preprocessDescGeoAns(ans):
    """
    ans here is a list of strings
    We remove punctuation and tokenize the words.
    We remove the duplicates and then
    we lemmatize the remaining words.
    Returns a list of words.
    """
    ans = ''.join(ans)
    # ans = ans.lower()
    procText = re.sub(r'[^\w\s]', '', ans.replace("'s", ""))
    tokenText = word_tokenize(procText)
    unique = list(dict.fromkeys(tokenText)) #Remove duplicates
    tags = pos_tag(unique)
    convAns = [lemmatizer.lemmatize(x[0], pos="n").lower() for x in tags]

    return convAns
```

5.3.3.1 Αξιολόγηση Descriptive ερωτήσεων βάση συχνότητας λέξεων

Στην παρούσα υποενότητα θα παρουσιάσουμε την πιο απλή περίπτωση από τις δύο μεθόδους αξιολόγησης ως προς την ορθότητα των απαντήσεων που έχουμε πάρει από τα LLMs.

Η συνάρτηση metricDesc υλοποιεί τον παραπάνω τρόπο. Πιο συγκεκριμένα, η metricDesc λαμβάνει τις δύο απαντήσεις ακριβώς με τον τρόπο που προ-επεξεργάστηκαν αυτές, δηλαδή σαν λίστες από επιλεγόμενες λέξεις όπως τις πήραμε από το βήμα της προ-επεξεργασίας.

Στην συνέχεια, αρχικοποιούμε την μεταβλήτη count και της εκχωρούμε την τιμή "0". Για κάθε λέξη από την λίστα που αντιστοιχεί στην επεξεργασμένη απάντηση αναφοράς, ψάχνουμε να βρούμε αν υπάρχει στην αντίστοιχη λίστα της απάντησης του LLM.

Αν αυτή βρεθεί, τότε αυξάνουμε την μεταβλητή count κατά 1. Όταν ολοκληρωθεί η εν λόγω διαδικασία για όλες τις λέξεις της απάντησης του γνωσιακού γράφου, τότε η συνάρτηση

metricDesc θα επιστρέψει το πηλίκο του count σε σχέση με τον συνολικό αριθμό των λέξεων της απάντησης του γράφου.

Παραθέτουμε την συνάρτηση metricDesc σε Python:

```
def metricDesc(LlmAns, GeoAns):
"""
LlmAns: is a list of preprocessed words. For example,
we want to have this structure ["capital", "greece", "athens"].
We have retained only the nouns through POS tagging
and we have performed lemmatization.
GeoAns: is a list containing containing processed words
from the answer of the knowledge graph.
"""

count = 0
for word in GeoAns:
    if word in LlmAns:
        count += 1
return count/len(GeoAns)
```

5.3.3.2 Αξιολόγηση Descriptive ερωτήσεων με την μετρική cosine similarity

Για τα πλαίσια υλοποίησης αυτής της μεθόδου γράφτηκε ένα ξεχωριστό script, από το οποίο φορτώνουμε εκεί τις απαντήσεις αναφοράς καθώς και αυτές που μας δόθηκαν από το LLM.

Η προ-επεξεργασία των απαντήσεων γίνεται με τον ίδιο τρόπο, όπως ακριβώς υλοποιήθηκε στην απλή μέθοδο που εξετάσαμε την συχνότητα εμφάνισης των λέξεων των απαντήσεων αναφοράς σε σύγκριση με αυτών των μεγάλων γλωσσικών μοντέλων.

Γι΄αυτό τον λόγο φορτώνουμε και πάλι τα ίδια modules. Συγκεκριμένα, καλούμε τα εξής:

```
import nltk
from nltk import pos_tag
from nltk import word_tokenize
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
stop_words.remove("no")
stop_words.remove("not")
stop_words.remove('d')
```

Ταυτόχρονα, θα δουλέψουμε την βιβλιοθήκη sklearn [4] καλώντας τα ακόλουθα modules:

- from sklearn.feature_extraction.text import CountVectorizer

[4] https://scikit-learn.org/stable/

- from sklearn.metrics.pairwise import cosine_similarity

Η κλάση CountVectorizer μας βοηθάει στο να φτιάξουμε το κατάλληλο λεξιλόγιο από τις λέξεις που συναντήσαμε από απαντήσεις που δόθηκαν σε αυτό. Στην συγκεκριμένη ορολογία, οι λέξεις αυτές καλούνται documents και το σύνολο των λέξεων με το οποίο θα γίνει η παρακάτω ανάλυση καλείται bag of words. Το σύνολο όλων των documents ονομάζεται corpus [14].

Επίσης, η CountVectorizer διατηρεί το πλήθος των λέξεων που συνάντησε από τα documents αυτά κάθε αυτά (term-frequency). Για την πληρέστερη κατανόηση της συγκεκριμένης κλάσης, παρατίθεται το κάτωθι παράδειγμα.

Πρώτα φορτώνουμε την εν λόγω κλάση από το module sklearn.feature_extraction.text.

```
from sklearn.feature_extraction.text import CountVectorizer
```

Παρακάτω παραθέτουμε και το corpus (συλλογή των documents) που θα δουλέψουμε για το παρόν παράδειγμα.

```
corpus = [ ”Python is amazing and fun.”, ”Python is not just fun but also powerful.”,
    ”Learning Python is fun and Python rocks!”, ”Java is very important” ]
```

Στην συνέχεια, αρχικοποιούμε ένα αντικείμενο CountVectorizer:

```
vectorizer = CountVectorizer()
```

Αυτό το αντικείμενο εμπεριέχει τις μεθόδους με τις οποίες θα δημιουργήσουμε το λεγόμενο λεξιλόγιο ή bag of words καθώς και θα μετατρέψουμε κάθε document σε one-hot vector [14].

Το επόμενο βήμα είναι να καλέσουμε την μέθοδο fit_transform και να της δώσουμε ως όρισμα το corpus. Η συγκεκριμένη μέθοδος φτιάχνει το λεξιλόγιο και επιστρέφει το document-term frequency [14] ως πίνακα.

```
X = vectorizer.fit_transform(corpus)
```

Το X.toarray() παράγει τον πίνακα 5.1

0	1	1	0	1	0	1	0	0	0	0	0	1	0	0
1	0	0	1	1	0	1	0	1	0	1	1	1	0	0
0	0	1	0	1	0	1	0	0	1	0	0	2	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	1

Table 5.1: Document-Term matrix

Ο πίνακας 5.1 αντιστοιχεί σε κάθε γραμμή του ένα document καθώς και δίνει το term-frequency της κάθε λέξης που βρέθηκε στο πρώτο για αυτές που έχουν συμπεριληφθεί στο vocabulary.

Αντίστοιχα, η μέθοδος vectorizer.get_feature_names_out() δίνει το vocabulary ως λίστα. Άρα η εντολή print(”Vocabulary:”,vectorizer.get_feature_names_out()), θα μας δώσει το ακόλουθο μήνυμα:

Vocabulary: [’also’ ’amazing’ ’and’ ’but’ ’fun’ ’important’ ’is’ ’java’ ’just’ ’learning’ ’not’ ’powerful’ ’python’ ’rocks’ ’very’]

Βάση της σειράς που εμφανίζονται οι παραπάνω λέξεις, το αντικείμενο ”vectorizer” μετατρέπει το κάθε document στα παραπάνω διανύσματα. Για παράδειγμα, στον πίνακα 5.1 το τρίτο διάνυσμα αντιστοιχεί στο document ”Learning Python is fun and Python rocks!” του corpus.

Ειδικότερα, παρατηρούμε ότι τα δύο στοιχεία είναι μηδέν. Αυτό συμβαίνει επειδή η εν λόγω πρόταση δεν έχει τις λέξεις ”also” και ”amazing” που είναι αντίστοιχα η πρώτη και η δεύτερη σε σειρά λέξεις στο vocabulary. Αντίστοιχα, η λέξη ”Python”, εμφανίζεται δύο φορές στην πρόταση και είναι η τρίτη από το τέλος λέξη στο vocabulary. Γι’ αυτό και στο διάνυσμα το τρίτο στοιχείο από το τέλος έχει την τιμή δύο, δηλαδή είναι το term frequency. Αναφέρεται, δηλαδή, στην συχνότητα εμφάνισης της λέξης Python.

Αντίστοιχα, απ’ το module metrics.pairwise θα χρησιμοποιήσουμε την συνάρτηση cosine_similarity, όπου λαμβάνει δύο διανύσματα και υπολογίζει το μέτρο του cosine similarity [14] που αυτά έχουν μεταξύ τους.

Το συγκεκριμένο δίνει ένα μέτρο ομοιότητας μεταξύ δύο μη-μηδενικών διανυσμάτων που έχουν οριστεί σε έναν χώρο όπου μπορεί να γίνει η πράξη εσωτερικού γινομένου.

Το cosine similarity είναι η εφαπτομένη της γωνίας των δύο διανυσμάτων, δηλαδή το άθροισμα του γινομένου κάθε συντεταγμένης του ενός διανύσματος με την αντίστοιχη του άλλου διαιρούμενο με το γινόμενο των μηκών τους.

Η εξεταζόμενη μετρική, στην ουσία, εξαρτάται από την γωνία που σχηματίζουν τα δύο διανύσματα και όχι από το μέγεθός τους. Επίσης, η μετρική αυτή παίρνει τιμές από το διάστημα [-1, 1]. Στο δικό μας πλαίσιο, η cosine μετρική δεν μπορεί να πάρει αρνητικές τιμές, καθότι όλα τα στοιχεία θα είναι μη αρνητικοί αριθμοί.

Αν η τιμή είναι 1, τότε τα διανύσματα πρέπει να είναι ανάλογα. Στα πλαίσια της ανάλυσής μας αυτό θα σήμαινε ότι τα documents έχουν τις ίδιες λέξεις που εμπεριέχονται στο vocabulary. Αντίθετα, αν είναι 0, τότε τα διανύσματα είναι ορθογώνια και αυτό συνεπάγεται ότι δεν έχουν κανένα όμοιο στοιχείο.

Η ουσία του εξεταζόμενου αλγορίθμου είναι να μετατρέψουμε τις απαντήσεις του LLM και του γνωσιακού γράφου σε one-hot vectors, δηλαδή διανύσματα που έχουν είτε το 0 είτε το 1 σε κάθε σημείο τους. Αγνοούμε δηλαδή το term-frequency και βρίσκουμε το cosine similarity των προαναφερόμενων διανυσμάτων.

Η υλοποίηση του παραπάνω αλγορίθμου περιγράφεται ως εξής:

Φορτώνουμε τέσσερα αρχεία που εμπεριέχουν τις σωστές απαντήσεις, αυτές που δόθηκαν από τα LLMs, τις κατηγορίες που ανήκουν οι ερωτήσεις, καθώς και τον τύπο της κάθε ερώτησης.

Φτιάχνουμε το dictionary όπου θα αποθηκευτούν τα αποτελέσματα καθώς και τη συχνότητα της κάθε κατηγορίας ερωτήσεων. Μετά προσθέτουμε στο corpus τις επεξεργασμένες απαντήσεις του γράφου. Η σειρά εισαγωγής στην λίστα corpus, διατηρείται με τον τρόπο που προσθέτονται οι απαντήσεις. Επειδή όμως τα κλειδιά δεν αυξάνονται κατά 1 κάθε φορά που πάμε στην επόμενη Descriptive ερώτηση, φτιάχνουμε ένα dictionary που αντιστοιχεί τα κλειδιά αυτού του τύπου ερωτήσεων σε μια ακολουθία αριθμών που ξεκινάει από το μηδέν και αυξάνεται κατά ένα κάθε φορά που πάμε να προσθέσουμε την επόμενη απάντηση.

Άρα, η πρώτη ερώτηση που προστέθηκε στο corpus θα αντιστοιχεί στο 0 ανεξάρτητα ποιο είναι το κλειδί της στο αρχείο που έχουμε το benchmark Geoquestions1089. Αντίστοιχα, το κλειδί της δεύτερης απάντησης που προστέθηκε στο σύνολο θα απεικονίζεται στον αριθμό

1 κ.ο.κ.

Φυσικά και εδώ, αν η απάντηση αναφοράς είναι κενή, τότε δεν λαμβάνεται υπόψη. Έπειτα, αρχικοποιούμε ένα αντικείμενο CountVectorizer, δίνοντας το όρισμα binary=True, το οποίο δηλώνει ότι οι απαντήσεις θα μετατρέπονται σε διανύσματα που αποτελούνται μόνο από μηδέν και ένα.

Έπειτα, χρησιμοποιούμε την fit_transform, βάζοντας όλες τις απαντήσεις του benchmark σαν ορίσματα. Κατά αυτόν τον τρόπο, δημιουργούμε το λεξικό καθώς και τον πίνακα με όλα τα term-frequencies για κάθε απάντηση.

Έπειτα για κάθε κλειδί των descriptive ερωτήσεων, παίρνουμε τις απαντήσεις των LLM, τις επεξεργαζόμαστε όπως κάναμε στην προηγούμενη διαδικασία αξιολόγησης και χρησιμοποιούμε την μέθοδο transform για να τις μετατρέψουμε σε διανύσματα. Βρίσκουμε και το αντίστοιχο διάνυσμα της απάντησης αναφοράς που δημιουργήθηκε όταν βάλαμε το corpus στην μέθοδο fit_transform.

Για αυτή την διαδικασία χρησιμοποιούμε την αντιστοίχηση των κλειδιών με τον αριθμό της σειράς που εισήχθησαν οι ερωτήσεις στο corpus. Έτσι, έχουμε και τις δύο απαντήσεις στην μορφή one-hot vectors.

Μπορούμε λοιπόν, να βρούμε το cosine similarity των δύο απαντήσεων. Όπως αναφέρθηκε και παραπάνω, όσο πιο κοντά στο 1 είναι, τόσο θα ομοιάζουν οι απαντήσεις μεταξύ τους. Αντίθετα, όσο πιο κοντά στο 0 είναι το εξεταζόμενο score τόσο διαφέρουν οι απαντήσεις μεταξύ τους.

Εν συνεχεία, βρίσκουμε την κατηγορία που ανήκει το κλειδί της εξεταζόμενης ερώτησης και αυξάνουμε κατά ένα τον δείκτη που αντιστοιχεί στην εν λόγω κατηγορία στο dictionary freq, freq[cat] += 1.

Αν το score είναι μεταξύ του 0.25 και 0.55, τότε η απάντηση θεωρείται ως μερικώς σωστή, ενώ αν είναι μεγαλύτερο του 0.55 τότε την θεωρούμε σωστή. Όταν η κάθε απάντηση χαρακτηρίζεται ως σωστή ή μερικώς σωστή, τότε αυξάνουμε τον κατάλληλο δείκτη στο dictionary resultsDescr. Η συγκεκριμένη δομή, κρατάει όλες τις σωστές ή μερικώς σωστές απαντήσεις που έχουν βρεθεί ανά κατηγορία.

Τέλος, διαιρούμε για όλες τις σωστές και μερικώς σωστές απαντήσεις το πλήθος των ερωτήσεων για την κάθε κατηγορία.

Παρατίθεται ο πλήρης κώδικας παρακάτω:

```
resultsDescr = {}
for key in catKeys.keys():
    resultsDescr[key] = {'partially␣correct': 0, 'correct': 0}

freq = {}
for key in catKeys.keys():
    freq[key] = 0
#Create the corpus
corpus = list()
count = 0
keysMapp = {}
for key in typeKeys['Descr']:

    geoAns = cleanGeoQ[key]['Answer']
```

```
    geoAns = preprocessDescGeoAns(geoAns)
    if geoAns != '':
        corpus.append(geoAns)
        keysMapp[key] = count
        count += 1


#create the bag of words
vectorizer = CountVectorizer(binary=True)
X = vectorizer.fit_transform(corpus)

def find_category(key, categories):

    number = int(key)
    for category, values in categories.items():
        if number in values:
            return category

# Load the previously saved CountVectorizer instance
# with open('sparse_matrix.pkl', 'wb') as file:
#     pickle.dump(X, file)

correct = 0
partially = 0
for key in keysMapp.keys():

    #expect the answer to be in the first sentence.
    #or just put the whole answer of the LLM
    lans = llmAns[key]['Answer'].split('.')[0]
    lans = preprocessDescLlmAns(lans)
    enc_x = vectorizer.transform([lans])
    enc_y = X[keysMapp[key]].toarray()
    score = cosine_similarity(enc_x, enc_y)

    cat = find_category(key, catKeys)
    freq[cat] += 1
    if score >= 0.25 and score < 0.55:
        resultsDescr[cat]['partially␣correct'] += 1
    elif score >= 0.55:
        resultsDescr[cat]['correct'] += 1


for cat in resultsDescr.keys():

    resultsDescr[cat]['partially␣correct'] =
        resultsDescr[cat]['partially␣correct']/freq[cat] * 100
    resultsDescr[cat]['correct'] =
        resultsDescr[cat]['correct']/freq[cat] * 100
```

5.3.4 Δημιουργία γράφων για cardinal ερωτήσεις.

Η μεθοδολογία που ακολουθήσαμε για το επόμενο κομμάτι της πτυχιακής ήταν να αποτυπώσουμε με κάποιον φορμαλισμό στον ευκλείδειο χώρο γεωγραφικές σχέσεις για να αξιολογηθούν βάση αυτών η ικανότητα των LLMs να καταλαβαίνουν τις εν λόγω σχέσεις και να εξάγουν χρήσιμη πληροφορία από αυτά.

Ο μαθηματικός φορμαλισμός με τον οποίο θα αποτυπώσουμε τους γράφους μαθηματικά βασίζεται στην δουλειά του άρθρου ”Composing cardinal direction relations” [26].

Για τα πλαίσια της εργασίας μας, κρίνεται απαραίτητο να αναλυθεί το πλαίσιο του εν λόγω φορμαλισμού και μετά η μεθοδολογία και οι γράφοι που σχεδιάστηκαν στα πλαίσια της παρούσας αξιολόγησης.

5.3.4.1 Ορισμός minimum bounding box

Πρωτίστως, θα χρειαστεί να ορίσουμε την έννοια του minimum bounding box. Για να μπορέσουμε να ορίσουμε επαρκώς αυτόν τον όρο θα πρέπει πρώτα να αποτυπωθεί η έννοια του infimum και supremum μιας περιοχής.

Στα πλαίσια της ανάλυσής μας η περιοχή θα είναι ένας κόμβος, μια δηλαδή στρογγυλή επιφάνεια στο καρτεσιανό χώρο των δύο διαστάσεων. Επιπρόσθετα, στα πλαίσια της ανάλυσης μας λαμβάνεται υπόψη ότι όλοι οι κόμβοι ενός γράφου έχουν την ίδια επιφάνεια.

Δεδομένου ενός κόμβου a, ορίζουμε το $sup(a)_y$, ως το σημείο όπου η προβολή της ευθείας τέμνει τον κατακόρυφο άξονα ξεκινώντας από το υψηλότερο σημείου του γράφου.

Αντίστοιχα, το $inf(a)_y$ είναι το σημείο που η ευθεία εφάπτεται στον κάθετο άξονα ξεκινώντας από το χαμηλότερο σημείο του κόμβου a.

Επιπρόσθετα, το $sup(a)_x$ είναι το σημείο όπου η ευθεία τέμνει τον οριζόντιο άξονα όταν προβάλλεται από το δεξιότερο σημείο του κόμβου.

Τέλος, το $inf(a)_x$ είναι το σημείο όπου η ευθεία τέμνει τον οριζόντιο άξονα όταν προβάλλεται από το αριστερότερο σημείο του κόμβου.

Για την περαιτέρω κατανόηση των ανωτέρω εννοιών παρατίθεται το διάγραμμα 5.5.

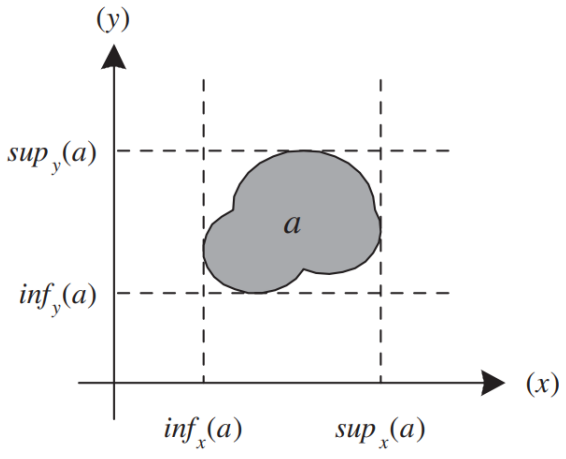

Figure 5.5: Μια περιοχή α και το bounding box

Μετά τον ορισμό των προαναφερόμενων τεσσάρων σημείων, θα εισάγουμε την έννοια του minimum bounding box.

Το minimum bounding box ενός κόμβου a, που συμβολίζεται ως mbb(a), είναι η ορθογώνια περιοχή που διαμορφώνεται από τις ευθείες γραμμές $x_1 = inf(a)_x$, $x_2 = sup(a)_x$, $y_1 = inf(a)_y$ και $y_2 = sup(a)_y$.

Είναι επακόλουθο, ότι οι προβολές στον κάθετο και οριζόντιο άξονα του κόμβου και του minimum bounding box έχουν τα ίδια τελικά σημεία.

5.3.4.2 Ορισμός cardinal σχέσεων

Στην συνέχεια θα ορίσουμε τις βασικές cardinal κατευθυντήριες σχέσεις του μοντέλου ως εξής:

Ορίζουμε το a S b, δηλαδή ότι ο κόμβος a είναι νότια του b αν και μόνο αν:

$$\text{infx}(b) \leq \text{infx}(a), \quad \text{supx}(a) \leq \text{supx}(b), \quad \text{infy}(b) \leq \text{infy}(a) \quad \text{and} \quad \text{supy}(a) \leq \text{supy}(b).$$

a SW b, δηλαδή ότι ο κόμβος a είναι νοτιοδυτικά του b αν και μόνο αν:

$$\text{supx}(a) \leq \text{infx}(b) \quad \text{and} \quad \text{supy}(a) \leq \text{infy}(b).$$

a W b, αν ο κόμβος a είναι δυτικά του b αν και μόνο αν:

$$\text{supx}(a) \leq \text{infx}(b), \quad \text{infy}(b) \leq \text{infy}(a) \quad \text{and} \quad \text{supy}(a) \leq \text{supy}(b).$$

a NW b, αν ο κόμβος a είναι βορειοδυτικά του b αν και μόνο αν:

$$\text{supx}(a) \leq \text{infx}(b) \quad \text{and} \quad \text{supy}(b) \leq \text{infy}(a).$$

a N b, αν ο κόμβος a είναι βόρεια του b αν και μόνο αν:

$$\text{supy}(b) \leq \text{infy}(a), \quad \text{infx}(b) \leq \text{infx}(a), \quad \text{and} \quad \text{supx}(a) \leq \text{supx}(b).$$

a NE b, αν ο κόμβος a είναι βορειοανατολικά του b αν και μόνο αν:

$$\text{supx}(b) \leq \text{infx}(a) \quad \text{and} \quad \text{supy}(b) \leq \text{infy}(a).$$

a E b, αν ο κόμβος a είναι ανατολικά του b αν και μόνο αν:

$$\text{supx}(b) \leq \text{infx}(a), \quad \text{infy}(b) \leq \text{infy}(a), \quad \text{and} \quad \text{supy}(a) \leq \text{supy}(b).$$

a SE b, αν ο κόμβος a είναι νοτιοανατολικά του b αν και μόνο αν:

$$\text{supx}(b) \leq \text{infx}(a) \quad \text{and} \quad \text{supy}(a) \leq \text{infy}(b).$$

Οι παραπάνω τύποι επαρκούν στον να αποτυπώσουμε τις σχέσεις των κόμβων ενός γράφου και να κάνουμε τις κατάλληλες ερωτήσεις στα LLMs για να διαπιστώσουμε αν μπορούν να κατανοήσουν τα ανωτέρω.

5.3.5 Δημιουργία prompts για αξιολόγηση των LLMs

Αρχικά, δοκιμάσαμε κάποιες ερωτήσεις στα μεγάλα μοντέλα, δηλαδή το ChatGPT-4o και το Llama 3.1 των 70 δις παραμέτρων.

Το υπόδειγμα του πρώτου prompt που δόθηκε παρατίθεται παρακάτω:

We have the following terms for a given node a of a graph defined in a two dimensional space. The supy(a) is the point in the vertical axis where the projection of the line from the highest point of the node is casted onto. The infy(a) is the point in the vertical axis where the projection of the line from the lowest point of the node is mapped. The supx(a) is the point in the horizontal axis where the projection of the line from the rightest point of the node is casted onto. The infx(a) is the point in the horizontal axis where the projection of the line from the most left point of the node is casted onto. The minimum bounding box of a region a, denoted by mbb(a), is the rectangular region formed by the straight lines x_1 = infx(a), x_2= supx(a), y_1 = infy(a) and y_2 = supy(a). Now we can formally define the atomic cardinal direction relations S, SW, W, NW, N, NE, E and SE of the model as follows: a S b iff supy(a) ≤ infy(b), infx(b) ≤ infx(a) and supx(a) ≤ supx(b). a SW b iff supx(a) ≤ infx(b) and supy(a) ≤ infy(b). a W b iff supx(a) ≤ infx(b), infy(b) ≤ infy(a) and supy(a) ≤ supy(b). a NW b iff supx(a) ≤ infx(b) and supy(b) ≤ infy(a). a N b iff supy(b) ≤ infy(a), infx(b) ≤ infx(a) and supx(a) ≤ supx(b). a NE b iff supx(b) ≤ infx(a) and supy(b) ≤ infy(a). a E b iff supx(b) ≤ infx(a), infy(b) ≤ infy(a) and supy(a) ≤ supy(b). a SE b iff supx(b) ≤ infx(a) and supy(a) ≤ infy(b). And from that point and onwards we describe the relations of some nodes using the aforementioned cardinals. Ask a question.

Ένα διαφορετικό prompt που δόθηκε στα LLMs για να εξετάσουμε αν βελτιώνει τις απαντήσεις που λαμβάνουμε από αυτά είναι το παρακάτω:

In a 2D graph, each node a has specific points: supy(a): The highest vertical point projected on the y-axis. infy(a): The lowest vertical point projected on the y-axis. supx(a): The farthest right point projected on the x-axis. infx(a): The farthest left point projected on the x-axis. The minimum bounding box (mbb) of node a is the rectangle formed by the lines:

x_1=inf(a)x, x_2= sup(a)x, y_1=inf(a)y, y_2= sup(a)y,

The directional relations (S, SW, W, NW, N, NE, E, SE) between two nodes a and b are defined as follows:

a S b iff supy(a) ≤ infy(b), infx(b) ≤ infx(a) and supx(a) ≤ supx(b).

a SW b iff supx(a) ≤ infx(b) and supy(a) ≤ infy(b).

a W b iff supx(a) ≤ infx(b), infy(b) ≤ infy(a) and supy(a) ≤ supy(b).

a NW b iff supx(a) ≤ infx(b) and supy(b) ≤ infy(a).

a N b iff supy(b) ≤ infy(a), infx(b) ≤ infx(a) and supx(a) ≤ supx(b).

a NE b iff supx(b) ≤ infx(a) and supy(b) ≤ infy(a).

a E b iff supx(b) ≤ infx(a), infy(b) ≤ infy(a) and supy(a) ≤ supy(b).

a SE b iff supx(b) ≤ infx(a) and supy(a) ≤ infy(b).

The cardinals directions can take only two values, ”0” for false and ”1” for true. Describe cardinal relations between some nodes. Ask the question.

5.3.6 Περιγραφή cardinals και ερωτήσεων για γράφους

Στην παρούσα ενότητα θα περιγράψουμε τις cardinal σχέσεις των κόμβων για κάθε γράφο που θα δώσουμε στα LLMs καθώς και θα αναφέρουμε τις ερωτήσεις πάνω στις οποίες θα αξιολογήσουμε τα μοντέλα.

Θα ξεκινήσουμε την ανάλυση μας με τον πρώτο γράφο με τους 3 κόμβους a, b και c, όπως αναφέρθηκε στο κεφάλαιο 4 στο διάγραμμα 4.7 Στον εν λόγω γράφο το a είναι βόρεια του b και και του c. Επίσης, ο b είναι βόρεια του c.

Στον εν λόγω γράφο, έχουμε τις εξής σχέσεις:

- a N b = 1 και a N c = 1
- b N c = 1 και b S a = 1
- c S a = 1 και c S b = 1

Εξυπακούεται ότι οι όλες οι άλλες σχέσεις ισούνται με το 0. Δηλαδή, b N a = 0, c N b = 0 και c N a = 0. Το ίδιο ισχύει και γα τα υπόλοιπα cardinals έτσι όπως ορίστηκαν ανωτέρω.

Για παράδειγμα έχουμε το ακόλουθο prompt για την περιγραφή του πρώτου γράφου και την πρώτη ερώτηση:

We define the following terms for a given node a of a graph in the Euclidean space of two dimensions. The supy(a) is the point in the vertical axis where the projection of the line from the highest point of the node is casted onto. The infy(a) is the point in the vertical axis where the projection of the line from the lowest point of the node is mapped. The supx(a) is the point in the horizontal axis where the projection of the line from the most point of the node is casted onto. The infx(a) is the point in the horizontal axis where the projection of the line from the most left point of the node is casted onto. The minimum bounding box of a node a, denoted by mbb(a), is the rectangular region formed by the straight lines x1 = infx(a), x2 = supx(a), y1 = infy(a) and y2 = supy(a). Now we can formally define the atomic cardinal direction relations S, SW, W, NW, N, NE, E and SE of the model as follows: a S b iff supy(a) ≤ infy(b), infx(b) ≤ infx(a) and supx(a) ≤ supx(b). a SW b iff supx(a) ≤ infx(b) and supy(a) ≤ infy(b). a W b iff supx(a) ≤ infx(b), infy(b) ≤ infy(a) and supy(a) ≤ supy(b). a NW b iff supx(a) ≤ infx(b) and supy(b) ≤ infy(a). a N b iff supy(b) ≤ infy(a), infx(b) ≤ infx(a) and supx(a) ≤ supx(b). a NE b iff supx(b) ≤ infx(a) and supy(b) ≤ infy(a). a E b iff supx(b) ≤ infx(a), infy(b) ≤ infy(a) and supy(a) ≤ supy(b). a SE b iff supx(b) ≤ infx(a) and supy(a) ≤ infy(b). The cardinals directions can take only two values, ”0” for false and ”1” for true. We have the following relations for a graph with three nodes a, b, c: a N b = 1, a N c = 1 and b N c = 1. What is the value of b N a?

Οι ερωτήσεις που θα ρωτήσουμε για τον απλό γράφο είναι οι εξής:

- What is the value of b N a?
- What is the value of c N b?
- What is the value of c N a?

- What is the value of b S a?
- What is the value of b S c?
- What is the value of c S b?
- What is the value of c S a?
- What is the value of a S b?
- What is the value of a S c?

Για τον δεύτερο γράφο θα δώσουμε τις εξής cardinal σχέσεις:

Μετέπειτα, θα δώσουμε και τον δεύτερο γράφο, που περιγράφηκε στην ενότητα 4 ως προς αξιολόγηση στο LLM. Η εν λόγω σχέσεις θα διαμορφωθούν ως εξής:

- a W b = 1 και a W c = 1
- b W c = 1 και b E a = 1
- c E a = 1 και c E b = 1

Η ακόλουθη σχέση a W b = 1 διαβάζεται ως εξής: Το a είναι ανατολικά του b είναι αληθής.

Επίσης, οι ερωτήσεις που θα ρωτήσουμε για τον δεύτερο γράφο παρουσιάζονται παρακάτω:

- What is the value of b W a?
- What is the value of c W b?
- What is the value of c W a?
- What is the value of b E a?
- What is the value of b E c?
- What is the value of c E b?
- What is the value of c E a?
- What is the value of a E b?
- What is the value of a E c?

Τέλος, θα παρουσιάσουμε και τις ερωτήσεις και την διατύπωση του τελευταίου γράφου. Οι σχέσεις που θα δοθούν στο prompt είναι οι εξής:

- b SW e = 1 και b NE a = 1
- b SE d = 1 και b NW c = 1

Τα ερωτήματα που θα τεθούν είναι οι αντίστροφες σχέσεις αυτών που δόθηκαν στην περιγραφή καθώς και κάποιες άλλες από τις τρεις εναπομένουσες θέσεις για κάθε ζεύγος από τα τέσσερα που δόθηκαν. Η τελευταία ερώτηση απλά αντιστρέφει μια σχέση που δόθηκε στην περιγραφή. Ουσιαστικά έχουμε:

- What is the value of b NW e?
- What is the value of b SW a?
- What is the value of b SW d?
- What is the value of b SE c?
- What is the value of e NE b?
- What is the value of a SW b?
- What is the value of d NW b?
- What is the value of c SE b?
- What is the value of e SW b?

5.3.7 Τι εξετάζουν οι ερωτήσεις

Τα LLMs στην ουσία θα εξεταστούν ως προς την ικανότητα τους να καταλαβαίνουν τις σχέσεις διευθύνσεις από το μοντέλο των cardinals, δηλαδή ως προς το spatial reasoning.

Για τον σκοπό αυτό έχουμε δημιουργήσει τρεις γράφους. Ο πρώτος περιγράφει για τους υποκείμενους κόμβους τις σχέσεις βόρεια και νότια. Ο δεύτερος τις σχέσεις ανατολικά και δυτικά και ο τρίτος τα cardinals των κατευθύνσεων: βορειοδυτικά, βορειοανατολικά, νοτιοδυτικά και νοτιοανατολικά.

Οι ερωτήσεις μπορούν να κατηγοριοποιηθούν σε τρεις ομάδες. Η πρώτη αφορά ερωτήσεις που αλλάζουν θέση τους όρους ενός cardinal που γνωρίζουμε ότι έχει τιμή 1. Για παράδειγμα, ”Δεδομένου ότι a R b = 1, ποια είναι η τιμή του b R a;”. Αυτή η περίπτωση θεωρείται τετριμμένη.

Ο δεύτερος τύπος ερωτήσεων αφορά τον υπολογισμό μιας σχέσης ενός cardinal δεδομένου ότι είναι αληθής η συμμετρική του. Παραδείγματος χάριν, ”Δεδομένο ότι a R b = 1, ποια είναι η τιμή του a R’ b;”. Αν το R αφορά το cardinal που απεικονίζεται ο βορράς, τότε έχουμε a N b = 1 και η συμμετρική του θα είναι b S a. Αντίστοιχα, η συμμετρική της σχέσης ”το b βρίσκεται νοτιοδυτικά του e”, είναι το ”e βρίσκεται βόρειο-ανατολικά του b”. Με τον συμβολισμό των cardinals θα έχουμε:

a N b = 1 <=> b S a = 1 b SW e = 1 <=> e NE b = 1

Στην ουσία, κάθε ένα από τα οκτώ cardinals που είναι αληθές μεταξύ δύο περιοχών συνεπάγεται και μια συμμετρική σχέση (των ίδιων περιοχών) που είναι επίσης αληθής.

Τέλος, με την τρίτη κατηγορία ερωτήσεων εξετάζουμε την ικανότητα του μοντέλου να καταλάβει το εξής:

Για μια σχέση κόμβων a και b, όπου a Rj b = 1, τότε a Ri b = 0 για i, j = 1,2,3,4,5,6,7,8, $i \neq j$ και R το σύνολο των cardinal σχέσεων.

Δηλαδή, για δύο κόμβους όπου a N b = 1, τότε όλες οι υπόλοιπες σχέσεις τους (a R b) είναι μηδέν. Για παράδειγμα, ρωτάμε το ”Ποια είναι η τιμή του b NW e, δεδομένου ότι b SW e = 1.”

6. ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΠΕΙΡΑΜΑΤΩΝ

6.1 Σύνοψη Κεφαλαίου 6

Σε αυτήν την ενότητα θα παρουσιάσουμε τα αποτελέσματα των πειραμάτων που βασίζονται στην ανάλυση του κεφαλαίου 5. Ειδικότερα, για τον πρώτο στόχο της εργασίας θα αποτυπώσουμε τα αποτελέσματα αναφορικά με την αξιολόγηση των LLMs πάνω στο benchmark GeoQuestions1089 που εξετάστηκαν και θα σχολιάσουμε τα αποτελέσματα.

Ταυτόχρονα, θα εξετάσουμε κατά πόσο τα LLMs είναι ικανά να κατανοήσουν τον φορμαλισμό των cardinal σχέσεων των κόμβων ενός γράφου από το μοντέλο που εισάγαμε και κατά πόσο ήταν αποτελεσματικά στο να εξάγουν σωστά και τις άλλες σχέσεις των κόμβων σύμφωνα με τις ερωτήσεις που τους τέθηκαν.

6.2 Αποτελέσματα LLMs στις ερωτήσεις GeoQuestions1089

Σύμφωνα με όσα περιγράψαμε στην ενότητα 5, θα παρουσιάσουμε τα αποτελέσματα που δόθηκαν από την απλή μέθοδο και την μετρική cosine για τις αντίστοιχες ερωτήσεις του δείκτη αναφοράς.

Συγκεκριμένα, για την απλή μέθοδο που υλοποιήθηκε για όλες τις ερωτήσεις επιχειρήσαμε να αποτυπώσουμε τα αποτελέσματα βάση των κατηγοριών των ερωτήσεων, έτσι όπως διαφοροποιούνται στο αρχείο GeoQuestions1089 καθώς και για την διάκριση των ερωτήσεων σε τρεις τύπους που έγινε στο πλαίσιο της ανάλυσής μας.

6.2.1 Αξιολόγηση με τον απλό τρόπο

Τα αποτελέσματα θα παρουσιαστούν σε μορφή ποσοστών, ήτοι σε πόσες ερωτήσεις το υπό εξέταση LLM απάντησε σωστά με βάση το πλήθος των ερωτήσεων που αντιστοιχεί σε κάθε ομαδοποίηση.

Για αυτό τον λόγο μετρήσαμε τον συνολικό αριθμό των ερωτήσεων που αντιστοιχεί σε κάθε ομάδα και τους απεικονίζουμε στον πίνακα 6.1:

Category	Binary	Descriptive	Quantitative
A	2	9	78
G	2	7	165
B	131	2	6
C	16	151	4
H	12	124	4
D	1	22	1
E	15	113	0
F	2	18	3
I	1	13	7

Table 6.1: Number of Questions allocated on each group

Με την βοήθεια του πίνακα 6.1 μπορούμε να διακρίνουμε την ομαδοποίηση των ερωτήσεων, βάση της οποίας θα γίνει η αξιολόγηση, καθώς και πόσες ερωτήσεις αντιστοιχούν στην

κάθε μία από αυτές τις ομάδες.

Ο αριθμός των ομάδων είναι 27. Στην ουσία για κάθε κατηγορία από τις 9 που υπήρχαν στο benchmark που εργαζόμαστε, τις χωρίζουμε περαιτέρω στους τρεις τύπους ερωτήσεων που αποφασίσαμε να γίνει η αξιολόγηση αυτών.

Για παράδειγμα, η κατηγορία A αναλύεται σε δύο ερωτήσεις τύπου Binary, εννιά τύπου Descriptive και εβδομηντα οκτώ ποσοτικών.

Category	Type	Partially Correct	Correct
A	Binary	0%	100%
A	Descr	11.11%	55.56%
A	Quant	0%	5.13%
G	Binary	0%	50%
G	Descr	28.57%	0%
G	Quant	0%	6.67%
B	Binary	0%	59.54%
B	Descr	0%	100%
B	Quant	0%	16.67%
C	Binary	0%	62.5%
C	Descr	26.49%	15.23%
C	Quant	0%	0%
H	Binary	0%	75%
H	Descr	25%	11.29%
H	Quant	0%	0%
D	Binary	0%	100%
D	Descr	4.55%	0%
D	Quant	0%	0%
E	Binary	0%	60%
E	Descr	15.93%	7.08%
E	Quant	0%	0%
F	Binary	0%	0%
F	Descr	16.67%	0%
F	Quant	0%	0%
I	Binary	0%	100%
I	Descr	30.77%	38.46%
I	Quant	0%	0%

Table 6.2: Αξιολόγηση του Llama 3.1 με τις 8 δις παραμέτρους (απλός τρόπος). Το correct σημαίνει ότι οι περισσότερες λέξεις ή όλες βρίσκονται στην απάντηση του LLM. Το partially correct σημαίνει ότι κάποιες μειοψηφικές λέξεις βρίσκονται στην εξεταζόμενη απάντηση

Από τα αποτελέσματα του πίνακα 6.2 μπορούμε αρχικά να διακρίνουμε ότι οι ερωτήσεις Binary θα είναι σωστές ή λάθος. Ταυτόχρονα, μπορούμε να δούμε ότι το μοντέλο έχει ικανοποιητική απόδοση σε αυτού του τύπου τις ερωτήσεις.

Αυτό είναι εύλογο, δεδομένου ότι αυτές οι ερωτήσεις έχουν απάντηση με ένα ”ναι” ή ”όχι”. Συνεπώς, το LLM μπορεί να απαντήσει με ικανοποιητική ακρίβεια καθότι δεν απαιτείται εξαιρετική δυσκολία σε αυτού του τύπου τις ερωτήσεις και μπορούν να αξιολογηθούν αποδοτικά με σχετικά απλό τρόπο.

Αντίθετα, η χειρότερη επίδοση παρατηρείται στις ποσοτικές ερωτήσεις. Από την εξερεύνηση των απαντήσεων για αυτού του τύπου τις ερωτήσεις τα αποτελέσματα είναι αναμενόμενα,

καθώς οι απαντήσεις που εμπεριέχονται στο GeoQuestions1089 δεν προσδιορίζουν πολλές φορές το μέτρο της ποσότητας που επιστρέφει και συχνά αυτό είναι ορισμένο έτσι ώστε να μην είναι εύκολο να απαντηθούν από τα εξεταζόμενα μοντέλα.

Για παράδειγμα, σε κάποιες ερωτήσεις του τύπου ”Ποια είναι η έκταση μιας περιοχής F;” ο γράφος επιστρέφει δύο αριθμούς για τους οποίους δεν προσδιορίζεται η ιδιότητά τους, ενώ το LLM συνήθως απαντά σε τετραγωνικά χιλιόμετρα. Αυτό κάνει την αξιολόγηση της απάντησης του μοντέλου δύσκολη.

Για τις περιγραφικές ερωτήσεις, η επίδοση είναι καλύτερη από αυτήν των ποσοτικών. Αυτό επίσης δεν μας προκαλεί κάποια εντύπωση, καθώς στις περισσότερες ερωτήσεις αυτού του τύπου η σωστή απάντηση αναφοράς εμπεριέχει πολλές λέξεις, κάποιες από τις οποίες είναι αποτέλεσμα εξειδικευμένης γνώσης που τα μικρά μοντέλα δεν την έχουν μάθει, λόγω των λίγων παραμέτρων τους.

Ουσιαστικά, η επίδοση της κατηγορίας Discriptive εξηγείται από το γεγονός ότι το Llama 3.1 βρίσκει ορισμένες λέξεις που εμπεριέχονται στις σωστές απαντήσεις αλλά χάνει τις περισσότερες.

Στους πίνακες 6.3 και 6.4 παρουσιάζονται τα αντίστοιχα αποτελέσματα για το μοντέλο mistral και gemma2.

Category	Type	Partially Correct	Correct
A	Binary	0%	100%
A	Descr	11.11%	55.56%
A	Quant	0%	2.56%
G	Binary	0%	100%
G	Descr	14.29%	14.29%
G	Quant	0%	5.45%
B	Binary	0%	63.36%
B	Descr	0%	100%
B	Quant	0%	0%
C	Binary	0%	50%
C	Descr	31.13%	16.56%
C	Quant	0%	0%
H	Binary	0%	83.33%
H	Descr	32.26%	12.10%
H	Quant	0%	0%
D	Binary	0%	100%
D	Descr	9.09%	0%
D	Quant	0%	0%
E	Binary	0%	33.33%
E	Descr	20.35%	8.85%
E	Quant	0%	0%
F	Binary	0%	100%
F	Descr	27.78%	0%
F	Quant	0%	0%
I	Binary	0%	100%
I	Descr	30.77	38.46%
I	Quant	0%	0%

Table 6.3: Αξιολόγηση του mistral των 7 δις παραμέτρων (απλός τρόπος)

Category	Type	Partially Correct	Correct
A	Binary	0%	100%
A	Descr	22.22%	44.44%
A	Quant	0%	2.56%
G	Binary	0%	50%
G	Descr	28.57%	0%
G	Quant	0%	2.42%
B	Binary	0%	64.12%
B	Descr	0%	100%
B	Quant	0%	0%
C	Binary	0%	43.75%
C	Descr	24.50%	12.58%
C	Quant	0%	0%
H	Binary	0%	66.67%
H	Descr	30.65%	7.26%
H	Quant	0%	0%
D	Binary	0%	0%
D	Descr	4.55%	0%
D	Quant	0%	0%
E	Binary	0%	60%
E	Descr	11.50%	7.08%
E	Quant	0%	0%
F	Binary	0%	0%
F	Descr	11.11%	0%
F	Quant	0%	0%
I	Binary	0%	100%
I	Descr	46.15%	23.08%
I	Quant	0%	0%

Table 6.4: Αξιολόγηση του gemma2 των 9 δις παραμέτρων (απλός τρόπος

Παρατηρούμε ότι η επίδοση των προαναφερόμενων μοντέλων κυμαίνονται σε παρόμοια πλαίσια με αυτή του Llama 3.1.

Αυτό σημαίνει ότι τα μοντέλα είναι καλύτερα στο να απαντούν σωστά σε Binary ερωτήσεις, μετά σε ένα σημαντικό αλλά μικρό ποσοστό των Descriptive ερωτήσεων και τέλος αποτυγχάνουν να απαντήσουν στην κατηγορία των ποσοτικών.

Τέλος, παρατηρούμε ότι τα μοντέλα έχουν παρόμοια επίδοση, με το Llama3.1 και το mistral να μπορούν να απαντήσουν σωστότερα σε περισσότερες απαντήσεις τύπου Binary σε σχέση με το gemma2.

Ταυτόχρονα το gemma2 παρόλο που έχει τις περισσότερες παραμέτρους στην αρχιτεκτονική του, έχει χειρότερη επίδοση από το Llama3.1 και από το mistral. Τα παραπάνω επιβεβαιώνουν την θέση ότι το μέγεθος των μοντέλων δεν είναι πάντα αρκετό για την αύξηση της απόδοσης τους.

6.2.2 Αξιολόγηση descriptive ερωτήσεων με την cosine μετρική

Σε αυτήν την υπό-ενότητα θα παρουσιάσουμε την επίδοση των μοντέλων για τις descriptive ερωτήσεις χρησιμοποιώντας την μετρική cosine, έτσι όπως περιγράφηκε στην προηγούμενη

ενότητα.

Πρώτα θα παρουσιάσουμε τον αριθμό των υποκείμενων ερωτήσεων που αντιστοιχούν σε κάθε κατηγορία, μιας και θα υπολογίσουμε τα αποτελέσματα σε μορφή ποσοστού σωστών απαντήσεων ανά κατηγορία. Αυτά παρατίθενται στον πίνακα 6.5.

Category	Count
A	9
G	7
B	2
C	151
H	124
D	22
E	113
F	18
I	13

Table 6.5: Αριθμός περιγραφικών ερωτήσεων ανά κατηγορία

Στο πίνακα 6.6 παρουσιάζουμε τα αποτελέσματα για το μοντέλο Llama 3.1.

Category	Partially Correct (%)	Correct (%)
A	22.22%	22.22%
G	28.57%	0%
B	0%	50.00%
C	20.53%	3.97%
H	22.58%	7.26%
D	4.55%	0%
E	13.27%	0.88%
F	16.67%	0%
I	23.08%	0%

Table 6.6: Αποτελέσματα για το Llama 3.1 με την cosine μετρική

Παρουσιάζουμε αντίστοιχα και τα αποτελέσματα για το mistral και το gemma2 στους πίνακες 6.7 και 6.8:

Category	Partially Correct (%)	Correct (%)
A	66.67	0.0
G	28.57	0.0
B	0.0	50.0
C	21.19	2.65
H	18.55	10.48
D	0.0	0.0
E	10.62	0.88
F	16.67	0.0
I	23.08	7.69

Table 6.7: Αποτελέσματα για το Mistral με την cosine μετρική

Category	Partially Correct (%)	Correct (%)
A	22.22	22.22
G	28.57	0.00
B	0.00	50.00
C	19.21	7.28
H	23.39	9.68
D	0.00	0.00
E	7.96	2.65
F	16.67	0.00
I	46.15	15.38

Table 6.8: Αποτελέσματα για το Gemma2 με την cosine μετρική

Από τα παραπάνω αποτελέσματα είναι εύκολο να διαπιστώσουμε ότι η cosine μετρική όχι μόνο δεν δίνει καλύτερα αποτελέσματα από την προηγούμενη μέθοδο αλλά η επίδοση του μοντέλου φθίνει.

Αυτό ισχύει και για τα τρία μοντέλα που εξετάσαμε παραπάνω. Ο λόγος που συμβαίνει αυτό έχει να κάνει με την υλοποίηση του cosine similarity και συγκεκριμένα με την δημιουργία λεξιλογίου από το corpus.

Όπως έχουμε αναλύσει και στην προηγούμενη ενότητα 5, για την δημιουργία του λεξιλογίου χρησιμοποιούμε όλες τις λέξεις που συναντάμε στο corpus.

Όμως, επειδή τα μοντέλα LLM περιττολογούν αυτό έχει ως αποτέλεσμα το one-hot vector της αντίστοιχης απάντησης να εμπεριέχει σε περισσότερες θέσεις τον αριθμό ”1”. Και αυτό επειδή κάποιες λέξεις που βρέθηκαν στην απάντηση του LLM, βρίσκονται στο λεξιλόγιο αλλά δεν εμπεριέχονται στην απάντηση του GeoQuestions1089.

Για παράδειγμα, στην ερώτηση ”Which are the biggest towns in Crete?”, το LLM μπορεί να απαντήσει ”Crete which is an island of Greece, has Chania and Heraklion.”. Αν η απάντηση του γράφου είναι [”Chania”, ”Heraklion”], τότε το LLM έχει απαντήσει σωστά.

Όμως, αν η λέξη ”Greece” και ”island” βρίσκονται στο λεξιλόγιο, τότε δεν θα πάρουμε cosine score ”1” αλλά 0.707168, λόγω ότι τα διανύσματα της απάντησης του LLM και της αναφοράς θα είναι [1,1,1,1] και [1,1,0,0].

Στο παράδειγμα παραπάνω οι δύο τελευταίες θέσεις αντιστοιχούν στις λέξεις ”island” και ”Greece”.

Τα παραπάνω έχουν ως αποτέλεσμα η τιμή του cosine να μειώνεται σημαντικά και συνεπώς δικαιολογεί γιατί αυτή η μεθοδολογία έχει χειρότερη επίδοση από την πιο άμεση που περιγράφηκε στην προηγούμενη υπο-ενότητα 6.2.1 .

6.3 Αξιολόγηση LLM βάση γράφων

6.3.1 Αξιολόγηση πάνω στον πρώτο γράφο

Πρωτίστως, θα παρουσιάσουμε τα αποτελέσματα του ChatGPT-4o για τις εννιά ερωτήσεις αναφορικά με τον πρώτο γράφο. Αυτά απεικονίζονται στον πίνακα 6.9.

Correct Answer	Answer of LLM	Result
$b\,N\,a = 0$	0	Correct
$c\,N\,b = 0$	0	Correct
$c\,N\,a = 0$	0	Correct
$b\,S\,a = 1$	0 or 1	Inconclusive-False
$b\,S\,c = 0$	0	Correct
$c\,S\,b = 1$	0 or 1	Inconclusive-False
$c\,S\,a = 1$	0 or 1	Inconclusive-False
$a\,S\,b = 0$	0	Correct
$a\,S\,c = 0$	0	Correct

Table 6.9: Results of ChatGPT-4o on the first graph (First and Second Prompts)

Αντίστοιχα, παρουσιάζουμε και τα αποτελέσματα για το Llama3.3 των 70B παραμέτρων στους πίνακες 6.10.

Correct Answer	Answer of LLM	Result
$b\,N\,a = 0$	0	Correct
$c\,N\,b = 0$	0	Correct
$c\,N\,a = 0$	0	Correct
$b\,S\,a = 1$	0 or 1	Inconclusive-False
$b\,S\,c = 0$	0	Correct
$c\,S\,b = 1$	0 or 1	Inconclusive-False
$c\,S\,a = 1$	0 or 1	Inconclusive-False
$a\,S\,b = 0$	0	Correct
$a\,S\,c = 0$	0	Correct

Table 6.10: Results of Llama3.3 70B on the first graph (First and Second Prompts)

Αξίζει να αναφερθεί ότι τα παραπάνω αποτελέσματα είναι τα ίδια και για τα δύο prompts που δώσαμε στα LLMs για την εξέταση τους.

Παράλληλα για τα δύο prompts για τον πρώτο γράφο, ρωτήσαμε και τα πιο μικρά μοντέλα. Παραθέτουμε τις απαντήσεις του Llama3.1 8B για τις 8 ερωτήσεις του πρώτου γράφου καθώς και για το Gemma2 των 9B.

Ξεκινάμε με την παρουσίαση του Gemma2 για τα δύο prompts. Αυτή απεικονίζεται στους πίνακες 6.11 και και 6.12.

Correct Answer	Answer of LLM	Result
$b\,N\,a = 0$	0	Correct
$c\,N\,b = 0$	0	Correct
$c\,N\,a = 0$	0	Correct
$b\,S\,a = 1$	0	Incorrect
$b\,S\,c = 0$	0	Correct
$c\,S\,b = 1$	0	Incorrect
$c\,S\,a = 1$	0	Incorrect
$a\,S\,b = 0$	0	Correct
$a\,S\,c = 0$	0	Correct

Table 6.11: Results of Gemma2 9B on the questions of the first graph (First Prompt)

Correct Answer	Answer of LLM	Result
$b\,N\,a = 0$	0	Correct
$c\,N\,b = 0$	1	Incorrect
$c\,N\,a = 0$	0	Correct
$b\,S\,a = 1$	0	Incorrect
$b\,S\,c = 0$	0	Correct
$c\,S\,b = 1$	0	Incorrect
$c\,S\,a = 1$	0	Incorrect
$a\,S\,b = 0$	0	Correct
$a\,S\,c = 0$	0	Correct

Table 6.12: Results of Gemma2 9B on the questions of the first graph (Second Prompt)

Παρομοίως στα διαγράμματα 6.14 και 6.15 παρουσιάζονται και τα αποτελέσματα του Llama3.1 για τα δύο prompts.

Correct Answer	Answer of LLM	Result
$b\,N\,a = 0$	0	Correct
$c\,N\,b = 0$	1	Incorrect
$c\,N\,a = 0$	1	Incorrect
$b\,S\,a = 1$	0	Incorrect
$b\,S\,c = 0$	1	Incorrect
$c\,S\,b = 1$	No answer	Undefined
$c\,S\,a = 1$	1	Correct
$a\,S\,b = 0$	1	Incorrect
$a\,S\,c = 0$	1	Incorrect

Table 6.13: Results of Llama3.1 8B on the questions of the first graph (First Prompt)

6.3.2 Αξιολόγηση πάνω στον δεύτερο γράφο

Σε αυτή την υποενότητα θα εξετάσουμε την επίδοση των LLMs στον δεύτερο γράφο και τις ερωτήσεις του.

Αρχικά, παρουσιάζουμε τα διαγράμματα 6.15 και 6.16 για το ChatGPT-4o στον πίνακα και για τα δύο prompts αντίστοιχα.

Correct Answer	Answer of LLM	Result
$b\,N\,a = 0$	0	Correct
$c\,N\,b = 0$	0	Correct
$c\,N\,a = 0$	0	Correct
$b\,S\,a = 1$	0	Incorrect
$b\,S\,c = 0$	0	Correct
$c\,S\,b = 1$	1	Correct
$c\,S\,a = 1$	0	Incorrect
$a\,S\,b = 0$	0	Correct
$a\,S\,c = 0$	0	Correct

Table 6.14: Results of Llama3.1 8B on the questions of the first graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	0	correct
$c\,W\,a = 0$	0	correct
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	0	correct
$c\,E\,b = 1$	0	incorrect
$c\,E\,a = 1$	0	incorrect
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.15: Results of chatGPT-4o on the questions of the second graph (First Prompt)

Στα διαγράμματα 6.17 και 6.18 παραθέτουμε τις απαντήσεις του Llama3.3 70B για τα δύο prompts ανάλογα.

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	0	correct
$c\,W\,a = 0$	0	correct
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	1	incorrect
$c\,E\,b = 1$	0	incorrect
$c\,E\,a = 1$	0	incorrect
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.16: Results of chatGPT-4o on the questions of the second graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	0	correct
$c\,W\,a = 0$	0	correct
$b\,E\,a = 1$	1	correct
$b\,E\,c = 0$	0	correct
$c\,E\,b = 1$	1	correct
$c\,E\,a = 1$	1	correct
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.17: Results of Llama3.3 70B on the questions of the second graph (First Prompt)

Τέλος, στους πίνακες 6.19 και 6.20 έχουμε τα αποτελέσματα για το gemma2. Τα αποτελέσματα για το Llama3.1 8B απεικονίζονται στα σχήματα 6.21 και 6.22.

6.3.3 Αξιολόγηση πάνω στον τρίτο γράφο

Σε αυτή την υποενότητα θα εξετάσουμε την επίδοση των LLMs στον τρίτο γράφο και τις ερωτήσεις του. Στα σχήματα 6.23 και 6.24 παρουσιάζεται η επίδοση του ChatGPT4-4o. Στους πίνακες 6.25 και 6.26 έχουμε τα αποτελέσματα του Llama3.3.

Σε αυτό το σημείο θα παρουσιάσουμε τα αποτελέσματα για τα μικρά LLMs ξεκινώντας από το gemma2 όπου αναφέρονται στα σχήματα 6.27 και 6.28. Στους πίνακες 6.29 και 6.30 έχουμε τα αντίστοιχα αποτελέσματα για το Llama3.1.

6.4 Αποτύπωση αποτελεσμάτων και σχολιασμός

6.4.1 Σχολιασμός αποτελεσμάτων πρώτου γράφου

Στον πρώτο γράφο τα αποτελέσματα των δύο μεγάλων LLMs ταυτίζονται και για τα δύο prompts που εξετάστηκαν.

Επίσης, για τα μεγάλα LLMs, μπορούμε να δούμε ότι αναγνωρίζουν σωστά ότι αν έχουμε ένα cardinal της μορφής a R b = 1, τότε αν αλλάξουμε τις θέσεις των όρων της εν λόγω σχέσης θα έχουμε b R a = 0.

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	0	correct
$c\,W\,a = 0$	0	correct
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	0	correct
$c\,E\,b = 1$	0	incorrect
$c\,E\,a = 1$	0	incorrect
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.18: Results of Llama3.3 70B on the questions of the second graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	0	correct
$c\,W\,a = 0$	0	correct
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	0	correct
$c\,E\,b = 1$	-	Undefined
$c\,E\,a = 1$	-	Undefined
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.19: Results of gemma2 9B on the questions of the second graph (First Prompt)

Επιπρόσθετα, μπορούμε να διαπιστώσουμε ότι το ChatGPT-4o και το Llama3.3 των 70B παραμέτρων είναι πολύ καλά στο να εξάγουν την εξής ιδιότητα των cardinals:

Για μια σχέση κόμβων a και b, όπου $aR_jb = 1$, τότε $aR_ib = 0$ για i, j = 1,2,3,4,5,6,7,8, $i \neq j$ όπου το R συμβολίζει το σύνολο των cardinal σχέσεων.

Αναφορικά με τον εντοπισμό των συμμετρικών σχέσεων μεταξύ βορρά και νότου, τα δύο μεγάλα μοντέλα δεν είναι ικανά στο να κατανοήσουν την εν λόγω σχέση. Αξίζει να σημειωθεί, ότι ρωτήσαμε την συγκεκριμένη κατηγορία ερωτήσεων πολλές φορές, και συνήθως η απάντηση ήταν ”0”, δηλαδή λάθος. Υπήρχαν και λίγες φορές όπου απάντησε σωστά το ChatGPT-4o και το Llama3.3.

Αξίζει να σημειωθεί ότι κατά την διάρκεια της εν λόγω εξέτασης το Llama3.3 φαίνεται να μπερδεύει κάποιες φορές το a N b = 1 με το εξής ”Είναι αληθής ότι το b είναι βόρεια του a”, ακόμα και όταν καταλήγει σε σωστό συμπέρασμα.

Αναφορικά με την επίδοση των μικρών μοντέλων, μπορούμε να δούμε ότι είναι ικανά να απαντήσουν σωστά και λάθος σε όλους τους τύπους των ερωτήσεων. Είναι πιο εύκολο να κάνουν λάθος στην τετριμμένη ερώτηση της πρώτης κατηγορίας απ’ ότι τα μεγάλα μοντέλα παρόλο που δεν συμβαίνει πολύ συχνά.

Επίσης, πρέπει να προσθέσουμε ότι τα μικρά μοντέλα είναι και εν γένει πολύ ασταθή. Δεδομένου μιας ερώτησης του τύπου δύο και τρία (συμμετρία και συμπληρωματικά cardinals) μπορούν την μια να δώσουν θετική απάντηση και την επόμενη αρνητική.

Ικανοποιητική είναι η επίδοση του gemma2 που στο πρώτο prompt που απαντάει σε πολλές ερωτήσεις σωστά. Φυσικά, στο δεύτερο prompt απαντάει και στις τρεις συμμετρικές ερωτήσεις λάθος, τονίζοντας την δυσκολία που έχει το LLM σε αυτού του τύπου την ερώτηση.

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	1	incorrect
$c\,W\,a = 0$	0	correct
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	0	correct
$c\,E\,b = 1$	0	incorrect
$c\,E\,a = 1$	0	incorrect
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.20: Results of gemma2 9B on the questions of the second graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	1	incorrect
$c\,W\,a = 0$	1 and 0	Incoclusive
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	1	incorrect
$c\,E\,b = 1$	1	correct
$c\,E\,a = 1$	1	correct
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.21: Results of Llama3.1 8B on the questions of the second graph (First Prompt)

Το μικρό Llama3.1 έχει δυσκολία στο να απαντήσει σωστά στις ερωτήσεις του πρώτου prompt αλλά απαντάει σωστά στο δεύτερο με εξαίρεση δύο συμμετρικών ερωτήσεων.

6.4.2 Σχολιασμός αποτελεσμάτων δεύτερου γράφου

Στον δεύτερο γράφο το μεγάλο Llama απέδωσε καλύτερα γιατί βρήκε όλες τις ερωτήσεις, ακόμα και τις συμμετρικές που δυσκόλεψε τα μεγάλα μοντέλα αναφορικά με το πρώτο prompt. Στο δεύτερο prompt, η επίδοση του Llama είναι πάλι καλύτερη καθότι βρήκε μια εκ των τριών ερωτήσεων που αναφέρονται στην ιδιότητα της συμμετρίας, ενώ το ChatGPT δεν βρήκε καμία.

Αντίθετα, το ChatGPT-4o είχε κακή απόδοση καθώς απάντησε σωστά μόνο στις ερωτήσεις της πρώτης και τρίτης κατηγορίας και για τα δύο prompts.

Το gemma2 επίσης δεν είχε καλή απόδοση και βρήκε μόνο κάποιες από τις εύκολες απαντήσεις. Μάλιστα σε κάποιες δεν μπορούσε να απαντήσει ολοκληρωμένα και ισχυριζόταν ότι έπρεπε να έχει και άλλα στοιχεία για να απαντήσει με σωστό ή λάθος.

Το μικρό Llama3.1, είχε επίσης μια κακή επίδοση καταφέρνοντας όμως να βρει δύο από τις συμμετρικές απαντήσεις στο πρώτο prompt, παρόλο που δεν απάντησε σωστά στις τετριμμένες και στις ερωτήσεις του τρίτου τύπου. Για το δεύτερο prompt, ήταν ακόμη χειρότερη η επίδοση καθότι απάντησε σωστά σε τρεις μόνο, οι οποίες ήταν εύκολες.

Correct Answer	Answer of LLM	Result
$b\,W\,a = 0$	0	correct
$c\,W\,b = 0$	1	incorrect
$c\,W\,a = 0$	1	incorrect
$b\,E\,a = 1$	0	incorrect
$b\,E\,c = 0$	1	incorrect
$c\,E\,b = 1$	0	incorrect
$c\,E\,a = 1$	0 and 1	Inconclusive
$a\,E\,b = 0$	0	correct
$a\,E\,c = 0$	0	correct

Table 6.22: Results of Llama3.1 8B on the questions of the second graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	0	correct
$b\,SW\,a = 0$	0	correct
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	0	correct
$e\,NE\,b = 1$	1	correct
$a\,SW\,b = 1$	1	correct
$d\,NW\,b = 1$	1	correct
$c\,SE\,b = 1$	1	correct
$e\,SW\,b = 0$	0	correct

Table 6.23: Results of chatGPT-4o on the questions of the third graph (First Prompt)

6.4.3 Σχολιασμός αποτελεσμάτων τρίτου γράφου

Από τους πίνακες βλέπουμε ότι το ChatGPT-4o κατάφερε να απαντήσει σε όλες τις ερωτήσεις σωστά και για τα δύο prompts. Αντίθετα, το μεγάλο Llama3.3 έκανε κάποια λάθη αλλά πήγε πολύ καλύτερα στις ερωτήσεις της συμμετρίας σε σχέση με τον πρώτο γράφο.

Επίσης, από τον τρίτο γράφο μπορούμε να διαπιστώσουμε ότι το ChatGPT-4o και το Llama3.3 των 70B παραμέτρων είναι πολύ καλά στο να εξάγουν την εξής ιδιότητα των cardinals:

Για μια σχέση κόμβων a και b, όπου $aR_jb = 1$, τότε $aR_ib = 0$ για i, j = 1,2,3,4,5,6,7,8, $i \neq j$ και R το σύνολο των cardinal σχέσεων.

Αυτό φαίνεται στις ερωτήσεις του τρίτου γράφου, όταν και τα δύο μοντέλα απαντούν σωστά στις πρώτες 4 που ανήκουν στην εν λόγω κατηγορία.

Αναφορικά με τα μικρά μοντέλα έχουμε κυρίως αρνητικά αποτελέσματα, καθώς το gemma2 σε 4 ερωτήσεις από τις 9 σχετικά με το πρώτο prompt δεν δίνει συγκεκριμένη απάντηση και απαντάει σωστά σε μόνο μια συμμετρική ερώτηση. Αντίθετα, το μικρό Llama3.1 έχει πιο καλή απόδοση παρόλο που δεν βρίσκει 3 συμμετρικές από τις 4 για κάθε prompt και χάνει την τελευταία τετριμμένη απάντηση στο δεύτερο prompt.

Παραθέτουμε τα συγκεντρωτικά αποτελέσματα στον πίνακα 6.31.

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	0	correct
$b\,SW\,a = 0$	0	correct
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	0	correct
$e\,NE\,b = 1$	1	correct
$a\,SW\,b = 1$	1	correct
$d\,NW\,b = 1$	1	correct
$c\,SE\,b = 1$	1	correct
$e\,SW\,b = 0$	0	correct

Table 6.24: Results of chatGPT-4o on the questions of the third graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	0	correct
$b\,SW\,a = 0$	0	correct
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	0	correct
$e\,NE\,b = 1$	1	correct
$a\,SW\,b = 1$	0	incorrect
$d\,NW\,b = 1$	0	incorrect
$c\,SE\,b = 1$	0	incorrect
$e\,SW\,b = 0$	0	correct

Table 6.25: Results of Llama3.3 70B on the questions of the third graph (First Prompt)

6.4.4 Συγκεντρωτικά αποτελέσματα για κάθε μοντέλο

Τα παραπάνω αποτελέσματα κατατάσσουν το ChatGPT-4o ως το καλύτερο μοντέλο με απόδοση 75,9% και δεύτερο με μικρή διαφορά το Llama3.3 των 70B παραμέτρων με 74,07%. Τρίτο κατατάσσεται το gemma2 9B με 53,703% και τέταρτο το μικρό Llama3.1 48,15%.

Όλα τα μοντέλα δυσκολεύτηκαν στο να απαντήσουν τις συμμετρικές ερωτήσεις σωστά με τα δύο μεγάλα μοντέλα να έχουν επιτυχία σ' αυτές του τρίτου γράφου.

Τα αποτελέσματα του πίνακα 6.31 δείχνουν ότι τα LLMs δεν μπορούν ακόμα να απαντήσουν ικανοποιητικά σε βασικές ερωτήσεις που σχετίζονται με την χωρική αντίληψη δύο αντικειμένων σε έναν χώρο.

Ταυτόχρονα, κοιτάζοντας τις απαντήσεις των μοντέλων στις ερωτήσεις που τους τέθηκαν είδαμε ότι το ChatGPT-4o είναι επιρρεπές σε ένα τετριμμένο σφάλμα.

Για να ισχύει b SW a = 1, θα πρέπει:

$\sup_x(b) \leq \inf_x(a)$ and $\sup_y(b) \leq \inf_y(a)$

όμως το LLM επηρεαζόμενο από την σχέση στο prompt κάνει λάθος την θέση των γραμμάτων και γράφει:

$\sup_x(b) \leq \inf_x(a)$ and $\sup_y(a) \leq \inf_y(b)$

Επίσης, το μεσαίο Llama3.3 πολλές φορές δεν ακολουθεί πιστά τις λογικές αλληλουχίες που του δίνονται. Παραδείγματος χάριν, έχουμε ότι ισχύει η σχέση b SW e = 1 => $\sup_x(e) \leq \inf_x(b)$. Παρόλα αυτά το LLM ισχυρίζεται ότι η σχέση e NE b είναι λάθος γιατί το $\sup_x(e) \leq$

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	0	correct
$b\,SW\,a = 0$	0	correct
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	0	correct
$e\,NE\,b = 1$	0	incorrect
$a\,SW\,b = 1$	1	correct
$d\,NW\,b = 1$	1	correct
$c\,SE\,b = 1$	0	incorrect
$e\,SW\,b = 0$	0	correct

Table 6.26: Results of Llama3.3 70B on the questions of the third graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	No answer	Inconclusive
$b\,SW\,a = 0$	No answer	Inconclusive
$b\,SW\,d = 0$	No answer	Inconclusive
$b\,SE\,c = 0$	No answer	Incoclusive
$e\,NE\,b = 1$	1	correct
$a\,SW\,b = 1$	0	incorrect
$d\,NW\,b = 1$	0	incorrect
$c\,SE\,b = 1$	1	correct
$e\,SW\,b = 0$	0	correct

Table 6.27: Results of gemma2 9B on the questions of the third graph (First Prompt)

$\inf_x(b)$ δεν ισχύει.

Το gemma2 ταυτόχρονα φαίνεται να μην μπορεί να κατανοήσει ότι για κάθε μια σχέση που δίνεται στην περιγραφή, οι συμπληρωματικές της θα είναι μηδέν. Δηλαδή, ότι για b N c = 1, όλα τα υπόλοιπα cardinals του b σε αναφορά του c θα είναι ίσα με το μηδέν.

Αντίθετα, σε πολλές περιπτώσεις το gemma2 καθώς και το Llama3.1 μπορεί να αντιληφθεί την ακόλουθη σχέση:

b SW e=1 <=> e SW b = 0.

Μια τελευταία αξιοσημείωτη παρατήρηση είναι ότι το ChatGPT-4o στον γράφο 3 αναγνωρίζει σωστά τα συμμετρικά cardinals και αναφέρεται επιγραμματικά σε αυτή την ιδιότητα. Για παράδειγμα, κάνει αναφορά ότι οι σχέσεις NW και SE είναι συμμετρικές, δείχνοντας κάποια σημάδια χωρικής ευφυΐας (spatial intelligence).

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	No answer	Incoclusive
$b\,SW\,a = 0$	0	correct
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	0	correct
$e\,NE\,b = 1$	0	incorrect
$a\,SW\,b = 1$	0	incorrect
$d\,NW\,b = 1$	0	incorrect
$c\,SE\,b = 1$	0	incorrect
$e\,SW\,b = 0$	0	correct

Table 6.28: Results of gemma2 9B on the questions of the third graph (Second Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	0	correct
$b\,SW\,a = 0$	1	incorrect
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	No answer	Incoclusive
$e\,NE\,b = 1$	1	correct
$a\,SW\,b = 1$	0	incorrect
$d\,NW\,b = 1$	0	incorrect
$c\,SE\,b = 1$	0	incorrect
$e\,SW\,b = 0$	0	correct

Table 6.29: Results of llama3.1 8B on the questions of the third graph (First Prompt)

Correct Answer	Answer of LLM	Result
$b\,W\,e = 0$	0	correct
$b\,SW\,a = 0$	0	correct
$b\,SW\,d = 0$	0	correct
$b\,SE\,c = 0$	0	correct
$e\,NE\,b = 1$	1	correct
$a\,SW\,b = 1$	0	incorrect
$d\,NW\,b = 1$	0	incorrect
$c\,SE\,b = 1$	0	incorrect
$e\,SW\,b = 0$	1	incorrect

Table 6.30: Results of llama3.1 8B on the questions of the third graph (Second Prompt)

LLMS	Scores
ChatGPT-4o	41/54
Llama3.3 70B	40/54
Gemma2 9B	29/54
Llama3.1 8B	26/54

Table 6.31: Συνολικά αποτελέσματα για τα εξεταζόμενα LLMs

7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ

Σε αυτή την ενότητα θα περιγράψουμε κάποια βασικά συμπεράσματα από την αξιολόγηση των LLMs σχετικά με την ικανότητά τους να απαντούν σωστά στο benchmark GeoQuestions1089 καθώς και ως προς ικανότητα που επιδεικνύουν σχετικά με την χωρική τους αντίληψη.

7.1 Συμπεράσματα από το GeoQuestions1089

Οι συγκεκριμένες ερωτήσεις εστίαζαν σε αρκετά εξειδικευμένη πληροφορία και εκ πρώτης όψεως υπήρχε η αίσθηση ότι τα μικρά LLMs δεν θα μπορέσουν να ανταποκριθούν σε ικανοποιητικό βαθμό.

Πράγματι, σε πολλές ερωτήσεις τα υποκείμενα μοντέλα είτε παρουσίαζαν ψευδαισθήσεις (hallucinations), είτε ξεκαθάριζαν ότι δεν γνωρίζουν την πληροφορία.

Το φαινόμενο αυτό υπήρχε κυρίως στις ποσοτικές και περιγραφικές ερωτήσεις. Αυτό είναι εύλογο καθότι, πολλές απαντήσεις από τον εξεταζόμενο δείκτη αναφοράς αφορούν γεωχωρικές πληροφορίες και η ένωση των γράφων που παρείχαν τις αντίστοιχες απαντήσεις εμπεριέχει πολλά στατιστικά στοιχεία αυτού του είδους που δόθηκαν από τις επίσημες αρχές της Ελλάδας, Αγγλίας και Ιρλανδίας.

Αυτό έχει ως αποτέλεσμα στις ποσοτικές ερωτήσεις τα LLMs να έχουν πολύ κακή απόδοση, μιας και οι μετρικές που δίνονται από τους γνωσιακούς γράφους δεν προσδιορίζονται από την ερώτηση.

Επίσης, αναφορικά με την κατηγορία των περιγραφικών ερωτήσεων πολλές από αυτές διατυπώνονται ως εξής ”Ποιες είναι οι πόλεις που βρίσκονται κοντά/εντός Χ χιλιομέτρων ...”. Αυτή η διατύπωση είχε ως αποτέλεσμα να μπερδεύει τα μοντέλα στην εύρεση της απάντησης.

Συνεπώς, τα μοντέλα βρίσκουν κάποια συγκεκριμένα κομμάτια της απάντησης αλλά όχι αρκετά για να θεωρηθούν επιτυχημένα. Η πιο καλή απόδοση και για τα τρία LLMs που εξετάστηκαν επιτεύχθει στις Binary ερωτήσεις.

Αυτό θεωρούμε ότι συμβαίνει καθότι οι συγκεκριμένες ερωτήσεις είναι πιο απλές στην απάντηση τους και σε πολλές από αυτές δεν απαιτείται κάποια πολύ εξειδικευμένη γνώση. Συνεπώς, είναι πιθανόν κάποιες απαντήσεις να βρίσκονται στο σύνολο των δεδομένων όπου αυτά τα μοντέλα προπονήθηκαν.

Επιπρόσθετα, το γεγονός ότι τα μοντέλα είναι μικρά, δηλαδή έχουν λίγες παραμέτρους, μας υποδεικνύει ότι δεν μπορούν να μάθουν πολύ πληροφορία από το σύνολο δεδομένων κατά την διάρκεια της προπόνησης.

Τέλος, τα μοντέλα που έχουν καλύτερη απόδοση είναι το Llama3.1 8B, μετά το mistral 7B και τελευταίο το gemma2 9B, παρόλο που το τελευταίο έχει παραπάνω παραμέτρους από τα άλλα δύο και θα περιμέναμε να αποδίδει καλύτερα.

7.2 Συμπεράσματα για την χωρική αντίληψη

Έπειτα από την ανάλυση των αποτελεσμάτων στις ερωτήσεις που τέθηκαν στα LLMs, μπορούμε να συμπεράνουμε πως τα μοντέλα ακόμα δεν έχουν την κατάλληλη ικανότητα

για επιτυχημένη χωρική συλλογιστική. Καλύτερη επίδοση κρίνεται αυτή του ChatGPT-4o, με δεύτερο να έρχεται το Llama3.3 70B, τρίτο να ακολουθεί το gemma2 9B και τελευταίο το llama3.1 8B.

Τα εξεταζόμενα μοντέλα κάνουν συχνά λάθη όταν οι ερωτήσεις δεν είναι τετριμμένες και είναι ασταθή στις πιο απαιτητικές ερωτήσεις. Συγκεκριμένα, αν τεθεί πολλές φορές μια ερώτηση, μπορεί η απάντηση να αλλάξει. Αυτό παρατηρείται στις συμπληρωματικές ερωτήσεις για το ChatGPT-4o και το μεσαίο Llama3.3. Στα μικρά μοντέλα, η αστάθεια μπορεί να είναι και στις άλλες κατηγορίες ερωτήσεων.

Παρόλα αυτά, τα LLMs δείχνουν ότι μπορούν να συλλάβουν κάποιες έννοιες, αφού δώσουμε τους ορισμούς των cardinals έτσι όπως περιγράφτηκαν στο κεφάλαιο 5. Συγκεκριμένα, μπορούν να καταλάβουν ότι πρόκειται για βορρά, νότο, δύση και ανατολή κάποιες φορές χωρίς αυτές οι λέξεις να έχουν δοθεί στο prompt.

Τέλος, τα μεγάλα μοντέλα δείχνουν και ικανότητα στο να συλλάβουν κάποιες πιο δύσκολες έννοιες όπως η συμπληρωματική ιδιότητα των cardinals, όπως περιγράφτηκε στο κεφάλαιο 5.

7.3 Μελλοντική δουλειά

Για την αξιολόγηση LLMs σε γεω-χωρικές ερωτήσεις θα ήταν καλό να εξεταστούν και άλλοι δείκτες αναφοράς που εμπεριέχουν παρόμοιες ερωτήσεις.

Επιπρόσθετα, θα μπορούσαν να χρησιμοποιηθούν πιο προχωρημένες τεχνικές επεξεργασίας γλωσσικών δεδομένων ,όπως dependency parse tree, για την επίτευξη καλύτερων αποτελεσμάτων.

Ενδιαφέρον θα ήταν να επίσης να χρησιμοποιηθούν και άλλα μοντέλα με περισσότερες παραμέτρους και να συγκριθεί η απόδοση τους σε ορισμένους δείκτες αυτού του τύπου.

Μια πολύ σημαντική προσέγγιση του εν λόγω προβλήματος θα ήταν να εξεταστούν μια σειρά από LLMs διαφορετικών αρχιτεκτονικών αφού γίνει σε αυτά fine-tuning στην ικανότητα αποτύπωσης και εκμάθησης γεω-χωρικής πληροφορίας. Θα μπορούσε να γίνει η σύγκριση ως προς το μέγεθος των LLMs πριν το fine-tuning και μετά.

Κατά αυτό τον τρόπο θα ήταν εφικτό να εξετάσουμε την επίδραση του αριθμού των παραμέτρων ενός μοντέλου στο να μαθαίνει γεω-χωρική πληροφορία καθώς και στο πόσο αποτελεσματική είναι η τεχνική του fine-tuning για την εκμάθηση αυτού του είδους πληροφορίας.

Αναφορικά με τον έλεγχο της ικανότητας των LLMs να μπορεί να κατανοήσει γεωμετρίες και σχέσεις αντικειμένων στον χώρο θα μπορούσε να επεκταθεί η παρούσα εργασία, τόσο στο να συμπεριληφθούν πιο ιδιαίτεροι γράφοι αλλά και να επεκταθούν οι ερωτήσεις.

Η αύξηση των ερωτήσεων τόσο ως προς το πλήθος τους όσο και προς τις κατηγορίες τους (τι ρωτούν) θα έδινε ένα πλήρες benchmark για την αξιολόγηση των μοντέλων σε αυτή την περιοχή.

Θα ήταν επίσης ωφέλιμο, να χρησιμοποιηθούν περισσότερα prompts για να εξεταστεί αν τα μοντέλα αποδίδουν καλύτερα όταν οι περιγραφές αποτυπώνονται σε φυσική γλώσσα με κάποιον συγκεκριμένο τρόπο.

Τέλος, θα ήταν πολύ χρήσιμο να γίνουν τα εν λόγω πειράματα πολλές φορές για τις ίδιες ερωτήσεις, μέσω διαφορετικών prompts και γράφων για να μετρήσουμε την αστάθεια των μοντέλων και κατά πόσο αυτά απαντούν σταθερά στο ίδιο ερώτημα και κάτω από τις ίδιες περιγραφές.

ABBREVIATIONS - ACRONYMS

RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
LLM	Large Language Model
GIS	Geographic Information System
KDE	Kernel Density Estimation
PDF	Probability Density Function
QSR	Qualitative Spatial Reasoning
RCC	Region Connection Calculus
AI	Artificial Intelligence
GenAI	Generative artificial intelligence
GPU	Graphical Processing Unit
OGC	Open Geospatial Consortium
AGI	Artificial General Intelligence
URI	Uniform Resource Identifier

BIBLIOGRAPHY

[1] Resource description framework (rdf).

[2] AI@Meta. Llama 3 model card. 2024.

[3] Tariq Alghamdi, Mukesh Prasad, and Ali Braytee. Zero-shot implicit fine-grained named entity categorization with llm-based deep pattern mining. In *Australasian Conference on Information Systems, ACIS 2024, Canberra, Australia, December 4-6, 2024*, 2024.

[4] Abhinav Anand, Shweta Verma, Krishna Narasimhan, and Mira Mezini. A critical study of what code-llms (do not) learn. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 15869–15889. Association for Computational Linguistics, 2024.

[5] Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1), February 2008.

[6] Robert Battle and Dave Kolas. Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web*, 3(4):355–370, 2012.

[7] Ruth M.J Byrne and P.N Johnson-Laird. Spatial reasoning. *Journal of Memory and Language*, 28(5):564–575, 1989.

[8] Pranjal Chitale, Jay Gala, and Raj Dabre. An empirical study of in-context learning in LLMs for machine translation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7384–7406, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[9] Anthony G. Cohn. An evaluation of chatgpt-4's qualitative spatial reasoning capabilities in RCC-8. *CoRR*, abs/2309.15577, 2023.

[10] Anthony G. Cohn and José Hernández-Orallo. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of llms. *CoRR*, abs/2304.11164, 2023.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[12] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. In Michael Martin, Martí Cuquet, and Erwin Folmer, editors, *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016*, volume 1695 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[13] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[14] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2024. Online manuscript released August 20, 2024.

[15] Sergios-Anestis Kefalidis, Dharmen Punjani, Eleni Tsalapati, Konstantinos Plas, Mariangela Pollali, Michail Mitsios, Myrto Tsokanaridou, Manolis Koubarakis, and Pierre Maret. Benchmarking geospatial question answering engines using the dataset geoquestions1089. In Terry R. Payne, Valentina Presutti, Guilin Qi, María Poveda-Villalón, Giorgos Stoilos, Laura Hollink, Zoi Kaoudi, Gong Cheng, and Juanzi Li, editors, *The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part II*, volume 14266 of *Lecture Notes in Computer Science*, pages 266–284. Springer, 2023.

[16] Davide Liga and Luca Pasetto. Testing spatial reasoning of large language models: the case of tic-tac-toe. In Alessandro Bruno, Arianna Pipitone, Riccardo Manzotti, Agnese Augello, Pier Luigi Mazzeo, Filippo Vella, and Antonio Chella, editors, *Proceedings of the 1st Workshop on Artificial Intelligence for Perception and Artificial Consciousness (AIxPAC 2023) co-located with the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023), Roma, Italy, November 8, 2023*, volume 3563 of *CEUR Workshop Proceedings*, pages 64–79. CEUR-WS.org, 2023.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[18] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[19] OpenAI. Chatgpt: Language model. `https://chat.openai.com`, 2025. Accessed: 2025-01-22.

[20] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076, 1962.

[21] Hrishitva Patel, Rohit Valecha, Nishant Vishwamitra, and H. Raghav Rao. Automating information categorization using llms in crisis mapping platforms: An examination of ”requests for help” during the 2010 haiti earthquake. In Douglas R. Vogel, Heiko Gewald, Assadaporn Sapsomboon, Christy M. K. Cheung, Sven Laumer, and Jason Thatcher, editors, *Proceedings of the 45th International Conference on Information Systems, ICIS 2024, Advances in Methods, Theories, and Philosophy, Bangkok, Thailand, December 15-18, 2024*. Association for Information Systems, 2024.

[22] William Peng and Sam Powers. Llms and spatial reasoning: Assessing roadblocks and providing pathways to improvement. *Journal of Student Research*, 13(2), May 2024.

[23] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3):16:1–16:45, 2009.

[24] Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S. Ryoo, and Tsung-Yu Lin. Learning to localize objects improves spatial reasoning in visual-llms. *CoRR*, abs/2404.07449, 2024.

[25] N. Shadbolt, T. Berners-Lee, and W. Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.

[26] Spiros Skiadopoulos and Manolis Koubarakis. Composing cardinal direction relations. *Artif. Intell.*, 152(2):143–171, 2004.

[27] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan,

Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[29] Tong Zhan, Chenxi Shi, Yadong Shi, Huixiang Li, and Yiyu Lin. Optimization techniques for sentiment analysis based on LLM (GPT-3). *CoRR*, abs/2405.09770, 2024.