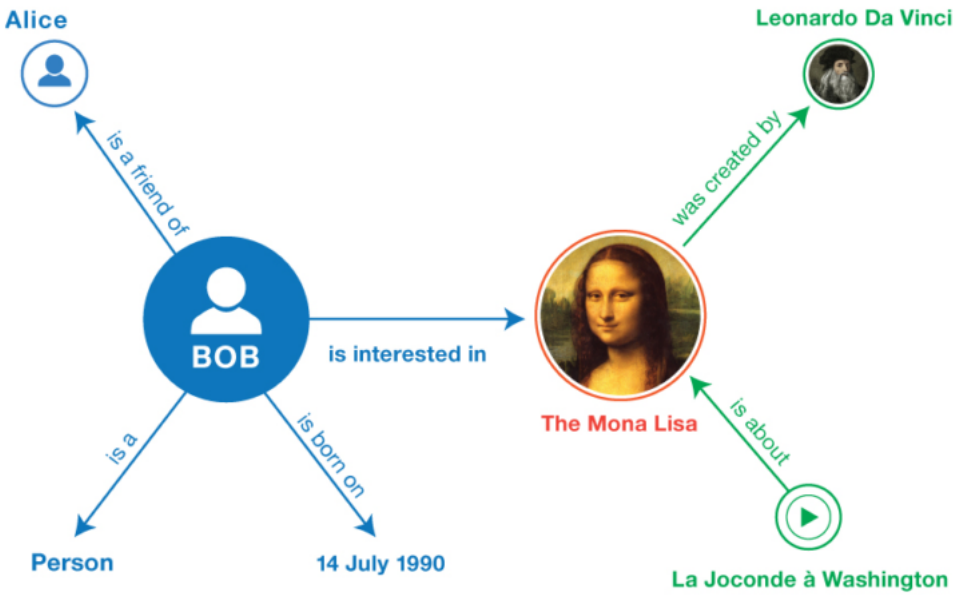
Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας

# Ενθέσεις Ερωτημάτων σε Γράφους Γνώσης

Διπλωματικη Εργασια

του

**ΚΑΚΑΜΠΑΚΟΥ Β. ΑΠΟΣΤΟΛΟΥ**




**Επιβλέπων:** Γιώργος Στάμου
Καθηγητής

Αθήνα, Νοέμβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας

# Ενθέσεις Ερωτημάτων σε Γράφους Γνώσης

Διπλωματικη Εργασια

του

**ΚΑΚΑΜΠΑΚΟΥ Β. ΑΠΟΣΤΟΛΟΥ**

**Επιβλέπων:** Γιώργος Στάμου
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31/10/2023.

| *(Υπογραφή)* | *(Υπογραφή)* | *(Υπογραφή)* |
|---|---|---|
| ..... | ..... | ..... |
| Γιώργος Στάμου | Αθανάσιος Βουλόδημος | Στέφανος Κόλλιας |
| Καθηγητής | Επίκουρος Καθηγητής | Καθηγητής |

Αθήνα, Νοέμβριος 2023

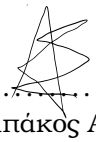
Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας



**ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ**

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

*(Υπογραφή)*



.....

Κακαμπάκος Απόστολος

31/10/2023

# Περίληψη

Οι κλάδοι της αναπαράστασης γνώσης και συλλογιστικής αποτελούν μερικά από τα σημαντικότερα πεδία της Τεχνητής Νοημοσύνης. Ο τρόπος με τον οποίο μπορεί κανείς να περιγράψει το κόσμο σε μία μηχανή ώστε αυτή να μπορεί να τον κατανοήσει, αλλά και να δράσει σε αυτόν είναι ένα ανοιχτό πρόβλημα. Από την αρχή, οι γράφοι γνώσης αποτέλεσαν ένα σημαντικό εργαλείο για την αναπαράσταση διαφόρων αντικειμένων και των σχέσεων μεταξύ τους. Μάλιστα με τη χρήση διαφόρων συστημάτων λογικής όπως οι περιγραφικές λογικές, μπορεί μια μηχανή να κάνει συλλογισμούς σε αυτούς τους γράφους. Από σχεδιασμού τους όμως, οι γράφοι γνώσης είναι πρακτικά πάντα μη πλήρεις, το οποίο όμως σίγουρα θα οδηγήσει και σε εσφαλμένους συλλογισμούς. Επομένως, δημιουργήθηκαν δύο γενικά προβλήματα. Το πρώτο είναι η συμπλήρωση γράφων γνώσης, από απλή συλλογή πληροφοριών μέχρι χρήση τεχνικών μηχανικής μάθησης. Το δεύτερο πρόβλημα, στο οποίο και υπάγεται η διπλωματική αυτή εργασία, είναι η συλλογιστική σε μη πλήρεις γράφους. Όπως στα άλλα προβλήματα της τεχνητής νοημοσύνης, έτσι και σε αυτό οι πρώτες τεχνικές βασίζονταν σε έξυπνους ¨γραμμένους με το χέρι¨ κανόνες, όμως στη συνέχεια οδηγήθηκε η προσπάθεια στην υιοθέτηση προσεγγίσεων που βασίζονται σε πολλά δεδομένα. Στη συνέχεια η χρήση μοντέρνων τεχνικών, όπως οι ενθέσεις, χρησιμοποιήθηκαν και στα δύο προβλήματα. Η αρχή έγινε στο πρόβλημα της πρόβλεψης συνδέσμου όπου χρησιμοποιήθηκαν απλές τεχνικές ενσωμάτωσης όπως το Transe. Μετέπειτα, οι τεχνικές ένθεσης αποτέλεσαν βασικό κορμό ανάλυσης προβλημάτων σε αυτό το πεδίο. Στόχος της διπλωματικής εργασίας είναι να ακολουθήσει και να εφαρμόσει μία σειρά ιδεών από διάφορες σύγχρονες εργασίες, όπως για παράδειγμα το Query2box. Σε αυτή την εργασία θα χρησιμοποιηθούν τα νευρωνικά δίκτυα για γραφήματα (GNN), για την ένθεση οντοτήτων και ερωτημάτων με στόχο το συλλογισμό σε μη πλήρεις γράφους γνώσης. Αρχικά θα γίνει αναφορά στη συλλογή δεδομένων για τις εκπαιδεύσεις. Έπειτα, θα παρουσιαστούν πληθώρα αρχιτεκτονικών, εμπνευσμένων από ερευνητικές εργασίες και κατ' επέκταση θα γίνει ανάλυση της εκπαίδευσης τους. Τέλος, θα γίνει σύγκριση με τις επιδόσεις τεχνικών τελευταίας τεχνολογίας στις μετρικές της βιβλιογραφίας.

## Λέξεις Κλειδιά

γράφοι γνώσης, ενθέσεις, ερωτήματα, οντότητες, σχέσεις, κατακερματισμός ερωτημάτων, SPARQL, FB15k237, WN18rr,GNN, rgcn, rgat, hits@, mrr

# Abstract

The fields of knowledge representation and reasoning, are fundamentally important to the development of Artificial Intelligence. The way in which one can describe the world to a machine so it can be able to understand as well as reason in said world is an open problem. Almost from the beginning, knowledge graphs became an important tool to represent objects and their relationships. Not only that, but using various systems of logic, like description logics, machines could in principle reason with these graphs about the world. Unfortunatelly, by design, knowledge graphs are almost always incomplete, which will certainly lead to false reasonings. Therefore, we have two general problems. The first is called knowledge graph completion, from using facts mining to using machine learning techniques. The second problem, which this thesis is based on, is called reasoning on incomplete knowledge graphs. As all other problems in AI, this started with clever "handwritten" rules, but grew to use data driven approaches. Later techniques such as embeddings where used for both problems. It all started with the problem of link prediction, in which simple embedding techniques like Transe where used. Later embedding techniques became the norm in the field. Aim of this work is to follow and apply a series of ideas expressed in various modern works, such as Query2box. In this thesis neural networks for graphs (GNN) will be used to embed both queries and entities so as to reason in incomplete graphs. First the data collection techniques used will be mentioned. Later, various architectures inspired by recent works will be presented and their training will be analyzed. In the end, the models will be compared to state of the art methods using metrics described in the literature

## Keywords

knowledge graphs, embeddings, queries, entities, relations, query hashing, SPARQL, FB15k237, WN18rr,GNN, rgcn, rgat, hits@, mrr

*στους γονείς μου*

# Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Γιώργο Στάμου για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης. Επίσης ευχαριστώ ιδιαίτερα τον υποψήφιο Διδάκτορα Έντι Ντιρβάκο για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Αθήνα, Νοέμβριος 2023

*Κακαμπάκος Απόστολος*

# Περιεχόμενα

# Κατάλογος Σχημάτων

# Κατάλογος Εικόνων

# Κατάλογος Πινάκων

# Πρόλογος

Η διπλωματική εργασία αυτή έγινε στα πλαίσια του μεταπτυχιακού προγράμματος Επιστήμη Δεδομένων και Μηχανική Μάθηση (ΔΠΜΣ) του Εθνικού Μετσόβιου Πολυτεχνείου, για το ακαδημαϊκό έτος 2022-2023. Εκπονήθηκε στο εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης (AILS Lab).

Κεφάλαιο 1

# Εισαγωγή

Ο γράφος γνώσης [1] αποτελεί ένα από τα σημαντικότερα εργαλεία αναπαράστασης γνώσης, που μπορεί να επεξεργαστεί ένας υπολογιστής. Με τη χρήση τους μπορεί κανείς να παρουσιάσει τις πολλαπλές σχέσεις μεταξύ διαφόρων οντοτήτων, το οποίο επιτρέπει, μέσω της αναζήτησης τους, την ανακάλυψη επιπλέον γνώσης που δεν ήταν προφανής προηγουμένως. Για την αποτελεσματική χρήση τους αλλά και αποθήκευση τους, χρησιμοποιούνται εξειδικευμένες βάσεις για γράφους [2]. Μία βάση γράφων είναι σχεδιασμένη για να μπορέσει να χειριστεί δεδομένα με δομή γράφων. Προσφέρει τρόπους αναπαράστασης, αποθήκευσης αλλά και δημιουργίας συνθέτων ερωτημάτων σε μία από τις πολλές γλώσσες που μπορούν να αναπαραστήσουν ερωτήματα σε γράφους. Όλα αυτά τα πετυχαίνουν οι βάσεις αυτές με τη χρήση συγχρόνων βελτιστοποιημένων αλγορίθμων διάσχισης γράφων. Παραδείγματα γνωστών βάσεων γνώσης αποτελούν βάσεις όπως η GraphDB της Ontotext, στην οποία μπορεί κανείς να δημιουργήσει ερωτήματα με τη SPARQL [3] και η Neo4j, η οποία έχει δικιά της γλώσσα τη Cypher [4]. Η χρήση των γράφων γνώσης και των βάσεων τους έχει επιτρέψει πληθώρα εφαρμογών στη βιομηχανία, όπως τα συστήματα συστάσεων [5]. Βέβαια, λόγω του ότι οι γράφοι γνώσης ποτέ δεν μπορούν να είναι πλήρεις, τα ερωτήματα πάνω στις βάσεις γνώσης μπορεί να δώσουν ελλειπή αποτελέσματα.

Ο κλάδος της μηχανικής μάθησης έχει υπάρξει ιδιαίτερα προσοδοφόρος τα τελευταία 20 χρόνια. Στη βάση της, η μηχανική μάθηση αναζητεί τη δημιουργία αλγορίθμων και μοντέλων που στηρίζονται σε πολλά δεδομένα για να πετύχουν πληθώρα έργων [6]. Ίσως η πιο διαδεδομένη τεχνική αποτελεί η βαθιά μάθηση [7], η οποία στηρίζεται στα νευρωνικά δίκτυα [8] που μιμούνται τη λειτουργία των βιολογικών νευρώνων, στα ¨μεγάλα¨ δεδομένα [9] τα οποία με την ¨έκρηξη¨ του ίντερνετ και των τεχνολογιών του υπάρχουν σε αφθονία, καθώς και στη χρήση gpu(s) [10] για τις υπολογιστικά ακριβείς πράξεις. Ανάλογα το πρόβλημα, υπάρχει διαφοροποίηση της αρχιτεκτονικής του νευρωνικού δικτύου αλλά και του μεγέθους του. Τυπικά έχει προκύψει η λογική, όσο μεγαλύτερα τα νευρωνικά δίκτυα τόσο καλύτερα, φυσικά με τις κατάλληλες προϋποθέσεις [11]. Μάλιστα τα τελευταία χρόνια έχει γίνει προσπάθεια ενσωμάτωσης νευρωνικών δικτύων για δεδομένα γράφων (GNN) [12, 13]

Κατ επέκταση έχει γίνει η προσπάθεια συνδυασμού της μηχανικής μάθησης αρχικά, και αργότερα της βαθιάς μάθησης με τους γράφους γνώσης. Λόγω της ανάπτυξης πολλαπλών γράφων γνώσης το πλήθος των δεδομένων επιτρέπει τις τεχνικές της μηχανικής μάθησης για διάφορα προβλήματα στους γράφους γνώσης, όπως η πρόβλεψη σύνδεσης [14] μεταξύ οντοτήτων.

## 1.1 Αντικείμενο της διπλωματικής

Η παρούσα διπλωματική εργασία απασχολείται με την χρήση τεχνικών μηχανικής μάθησης για την επίλυση του προβλήματος της απάντησης ερωτημάτων σε γράφους γνώσης [15]. Συγκεκριμένα, θα χρησιμοποιηθούν ειδικά είδη νευρωνικών δικτύων για γράφους, που όμως λαμβάνουν υπόψιν τους τον τύπο της σχέσης μεταξύ δύο οντοτήτων [16, 17]. Επιλέχθηκαν αυτού του είδους τα νευρωνικά δίκτυα καθώς μπορούν να διακρίνουν όχι μόνο τη δομή ενός ερωτήματος, αλλά και το περιεχόμενό του, δηλαδή τις οντότητες αλλά και σχέσεις που έχει. Κάθε ερώτημα και οντότητα θα αποκτούν ένθεση (embedding) και θα ορίζεται ένα score όπου όσο μεγαλύτερο είναι τόσο πιο σίγουρο είναι το εκάστοτε μοντέλο πως μία οντότητα αποτελεί απάντηση σε ένα ερώτημα, κατά αντιστοιχία με τη βιβλιογραφία. Θα γίνει ανάλυση τις μεθόδου δημιουργίας δεδομένων ερωτημάτων-απαντήσεων, καθώς και η εκπαίδευση διαφόρων μοντέλων νευρωνικών δικτύων για γράφους. Θα αναπτυχθούν τεχνικές ένθεσης και θα συγκριθούν με μεθόδους τελευταίας τεχνολογίας σε γνωστές μετρικές της βιβλιογραφίας. Τέλος θα γίνει αναφορά στα συμπεράσματα και επόμενα βήματα.

## 1.2 Σχετικές εργασίες

Οι τεχνικές ένθεσης έγιναν πρώτα διαδεδομένες στο τομέα της Επεξεργασίας Φυσικής Γλώσσας με τη θεμελιώδη δουλειά των Mikolov κ.α [18, 19]. Εκεί όρισαν ως ένθεση μίας λέξης ένα άνυσμα σε ένα πολυδιανυσματικό χώρο. Με την τεχνική τους προέκυψε πως τα ανύσματα αυτά, αν αντιστοιχούν σε λέξεις με κοντινό νόημα, θα είναι και αυτά κοντινά! Οπότε θα αποτελούν και χρήσιμες αναπαραστάσεις για διάφορα προβλήματα στο τομέα αυτό. Στο τομέα των πολυσχεσιακών δεδομένων άρχισαν σχετικά νωρίς οι εφαρμογές με χρήση ενθέσεων, όπως η Παραγοντοποίηση Πινάκων [20]. Η βασική εργασία, η οποία σίγουρα επηρέασε και την διπλωματική αυτή εργασία, αποτέλεσε η δουλειά των Bordes κ.α [21], στην οποία κάθε οντότητα έλαβε ένθεση, αλλά και κάθε σχέση. Έγινε η απαίτηση αν μία οντότητα συνδέεται με μία άλλη με μία κατευθυνόμενη σχέση, τότε το άθροισμα της ένθεσης της σχέσης με την ένθεση της οντότητας που βρίσκεται στην αρχή της κατεύθυνσης, οφείλει να είναι ίσο με την ένθεση της οντότητας στη κορυφή της σχέσης. Η τεχνική αυτή χρησιμοποιήθηκε στο πρόβλημα της πρόβλεψης συνδέσμου [14], πάνω στους γράφους γνώσης WN11 [22] και FB15k237 [23]. Η εργασία αυτή άνοιξε τις πύλες για πληθώρα εργασιών με αντίστοιχες τεχνικές, όπως η [24] και η [25], οι οποίες πέραν του προβλήματος της πρόβλεψης συνδέσμου απασχολήθηκαν με την αναπαράσταση των σχέσεων από τις τεχνικές τους σε σχέση με τα είδη των σχέσεων, που αναφέρθηκαν στην υποενότητα [2.1.1].

Το επόμενο βήμα έκανε η εργασία των Hamilton κ.α [26], όπου αντί να προσπαθήσει να λύσει το πρόβλημα της πρόβλεψης σύνδεσης, το γενίκευσε σε πρόβλημα της απάντησης ερωτημάτων σε μη πλήρεις γράφους (γνώσης). Συγκεκριμένα, προέκυψε η ιδέα της ένθεσης του ερωτήματος, εφόσον αναπαρασταθεί ως γράφος, καθώς και ένθεσης των οντοτήτων, ώστε το εσωτερικό γινόμενο της ένθεσης ενός ερωτήματος με την ένθεση μίας οντότητας απάντησης να είναι υψηλό, και αντίστοιχα αν δεν είναι απάντηση το εσωτερικό γινόμενο να είναι χαμηλό. Η ιδέα αυτή αποτέλεσε βασική έμπνευσή για αυτή την εργασία.

Μετέπειτα, η τεχνική Query2box [27] εισήγαγε τη χρήση πιο εξωτικών ενθέσεων και όχι μόνο ανυσμάτων. Συγκεκριμένα, η εργασία εισήγαγε την έννοια των ενθέσεων κύβου για τα ερωτήματα, όπου αν οι ενθέσεις ανύσματα των οντοτήτων, βρίσκονται μέσα στον κύβο, θεωρούνται απαντήσεις των ερωτημάτων. Με αυτή τη τεχνική υπάρχει η δυνατότητα τα ερωτήματα να καλύπτουν περισσότερο έδαφος, ώστε να μπορεί να γίνει καλύτερη αναπαράσταση σε ερωτήματα με πολλαπλές απαντήσεις. Παράλληλα έγινε η χρήση τεχνικών για άλλα προβλήματα όπως η εκμάθηση λογικών κανόνων [28, 29, 30], η χρήση τους σε υπεργράφους [31], αλλά και έχει γίνει αμφισβήτηση για το πόσο επεξηγήσιμες είναι όλες οι τεχνικές αυτές [32]. Ίσως, όμως η πιο θεμελιώδης εργασία και συνάμα κοντινή στη διπλωματική αυτή, αποτέλεσε η εργασία των Daniel Daza και Michael Cochez [33], όπου χρησιμοποίησαν σχεσιακούς γράφους γνώσης (RGNN) για την ένθεση των ερωτημάτων. Βέβαια τα δεδομένα που χρησιμοποίησαν, αποτελούσαν δεδομένα γράφων με τύπους, όπως AIFB, MUTAG [34] και όχι χωρίς, όπως το FB15k237. Η μέθοδος τους μάλιστα στηρίζεται στους τύπους αυτούς για τη πρόβλεψη. Σε αυτή την εργασία αντίθετα θα γίνει χρήση γράφων γνώσης χωρίς τύπους (που να είναι εκπεφρασμένοι).

## 1.3 Οργάνωση του τόμου

### Θεωρητικό Μέρος

Το **Θεωρητικό μέρος** περιέχει τα κεφάλαια **Θεωρητικό υπόβαθρο** και **Περιγραφή Θέματος**. Στο **Θεωρητικό υπόβαθρο** γίνεται αναφορά σε θεωρητικά στοιχεία που χρειάζεται κανείς για να καταλάβει την εργασία. Συγκεκριμένα περιγράφονται περιληπτικά οι γράφοι γνώσης και τα ερωτήματα σε αυτούς, καθώς και οι τα νευρωνικά δίκτυα σε γράφους γενικά αλλά και σε γράφους με πολλαπλές σχέσεις, όπως δηλαδή οι γράφοι γνώσης.

### Πρακτικό Μέρος

Το **Πρακτικό μέρος** περιέχει τα κεφάλαια **Ανάλυση και Σχεδίαση** και **Υλοποίηση**. Στο πρώτο κεφάλαιο αναφέρεται η σχεδίαση των αλγορίθμων για τη παραγωγή δεδομένων ερωτήματα-απαντήσεις, για την εκπαίδευση, καθώς και για τις μετρικές τις βιβλιογραφίας. Επίσης γίνεται ανάλυση των διάφορων μοντέλων ένθεσης της εργασίας. Στο κεφάλαιο **Υλοποίηση** παρουσιάζονται τα πειράματα και αποτελέσματα για τα διάφορα μοντέλα, καθώς και διάφορες παρατηρήσεις πάνω σε αυτά.

### Επίλογος

Στον Επίλογο αναλύονται τα συμπεράσματα της διπλωματικής εργασίας και αναφέρονται μελλοντικές επεκτάσεις που μπορεί να οδηγήσουν την έρευνα στο μέλλον.

# Μέρος Ι

# Θεωρητικό Μέρος

Κεφάλαιο 2

# Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό θα παρουσιαστούν περιληπτικά οι γράφοι γνώσης, τα ερωτήματα σε αυτούς με τη γλώσσα SPARQL, αλλά και θα γίνει μία αναφορά στα GNNs, αλλά και στα RGNNs.

## 2.1 Γράφοι Γνώσης

### 2.1.1 Εισαγωγή

Οι γράφοι γνώσης αποτελούν μία δομημένη αναπαράσταση πληροφορίας που περιγράφει τις σχέσεις μεταξύ οντοτήτων σε ένα πεδίο. Μάλιστα αναπαριστάνουν με τρόπο που είναι αναγνώσιμος από μηχανές, άρα και επεξεργάσιμος.

Σε ένα γράφο γνώσης, οι οντότητες αναπαριστώνται ως κόμβοι, και οι σχέσεις μεταξύ των οντοτήτων (κόμβων) αναπαριστάνουν ως ακμές που ενώνουν αυτούς τους κόμβους. Γενικά υπάρχουν πολλά ήδη γράφων γνώσης [1], σε αυτή τη διπλωματική θα χρησιμοποιηθούν γράφοι που έχουν μόνο τα παραπάνω στοιχεία.

Η αναπαράσταση ενός γράφου μπορεί να γίνει εύκολα με τη χρήση τριπλετών. Μια τριπλέτα αποτελείται από 3 κομμάτια: υποκείμενο, κατηγόρημα, και αντικείμενο.

- Υποκείμενο: Η οντότητα που περιγράφεται από το κατηγόρημα
- Κατηγόρημα: Η σχέση που περιγράφει τη τριπλέτα
- Αντικείμενο: Η οντότητα που είναι η τιμή της περιγραφής

Για παράδειγμα:

- (Υ: μπουκάλι νερό) (Κ: κόστος) (Α: 0.50)

Πολύ απλά ένα μπουκάλι νερό κοστίζει 50 λεπτά (σε Ευρώ), είναι η αναπαριστάμενη πληροφορία.

Εδώ λοιπόν δίνεται η ευκαιρία να διευκρινιστεί πως ένας γράφος γνώσης είναι ένας κατευθυνόμενος γράφος. Δεν αρκεί απλώς να υπάρχει μία σχέση μεταξύ οντοτήτων, αλλά αυτή οφείλει να έχει κατεύθυνση. Επίσης δύο οντότητες μπορεί να έχουν μεταξύ τους πολλαπλές σχέσεις, διαφορετικής ταυτότητας ή διεύθυνσης κάθε μία. Μάλιστα, κάθε σχέση ορίζει και τι είδους υποκείμενα και αντικείμενα μπορεί να δεχτεί γενικά. Σε αυτή την εργασία βέβαια

δεν θα χρησιμοποιηθούν γράφοι γνώσης που έχουν και εκπεφρασμένους τύπους για κάθε οντότητα, οπότε να μπορεί να γίνει αυτός ο ορισμός των σχέσεων. Συνεπώς, για τη συνέχεια του κειμένου θα γίνετε αναφορά μόνο σε γράφους χωρίς τύπους που είναι κατευθυνόμενοι!

Ένα άλλο παράδειγμα:

- (Υ: Αντώνης) (Κ: αγόρασε) (Α: μπουκάλι νερό)

Το οποίο σημαίνει πως ο Αντώνης έχει αγοράσει ένα μπουκάλι νερό. Το ανάποδο προφανώς δεν έχει κάποιο νόημα. Όμως αυτό δεν ισχύει πάντα.

- (Υ: Αντώνης) (Κ: είναιΔίπλα) (Α: μπουκάλι νερό)
- (Υ: μπουκάλι νερό) (Κ: είναιΔίπλα) (Α: Αντώνης)

Εδώ, οι δύο τριπλέτες έχουν ακριβώς το ίδιο νόημα, οπότε και σε μία βάση γνώσης δεν χρειάζεται κανείς να βάλει και τις δύο. Όμως πρέπει με κάποιο τρόπο να περάσει τη πληροφορία πως το κατηγόρημα ή σχέση ¨είναιΔίπλα¨ έχει αυτή την ιδιότητα.

Συνεπώς οι σχέσεις μπορεί να έχουν διάφορες ιδιότητες. Με τη χρήση αυτών των ιδιοτήτων μπορεί κανείς να μειώσει τον αριθμό των τριπλετών που έχει αποθηκευμένες, αλλά και να κάνει πολλούς συλλογισμούς.

Μερικά από τα πιο γνωστά ήδη σχέσεων είναι:

1. Συμμετρικές σχέσεις: Οι συμμετρικές σχέσεις είναι αυτές που δεν παίζει ρόλο η διεύθυνση. Αν Α σχέση Β, τότε Β σχέση Α.
2. Ασύμμετρες σχέσεις: Οι ασύμμετρες σχέσεις απλούστατα δεν μπορούν να αντιστραφούν. Αν Α σχέση Β, τότε δεν ισχύει ότι Β σχέση Α. Παράδειγμα αποτελεί η σχέση ¨κόστος¨.
3. Μεταβατικές σχέσεις: Οι σχέσεις αυτές έχουν την εξής ιδιότητα. Αν Α σχέση Β και Β σχέση Γ τότε Α σχέση Γ. Μία σχέση που έχει αυτή την ιδιότητα είναι η σχέση ¨περιέχει¨. Για παράδειγμα, (Ελλάδα, περιέχει, Αθήνα), (Αθήνα, περιέχει, ΕΜΠ) άρα (Ελλάδα, περιέχει, ΕΜΠ).
4. Αντίστροφες σχέσεις: Οι σχέσεις αυτές έχουν την ανάποδη διεύθυνση από την αντίστροφή τους. Για παράδειγμα η σχέση ¨περιέχεται¨ είναι αντίστροφη της σχέσης ¨περιέχει¨.

### 2.1.2 Υπόθεση Ανοικτού Κόσμου

Η Υπόθεση Ανοικτού Κόσμου [35] είναι μία θεμελιωτική υπόθεση στο κλάδο της αναπαράστασης και συλλογισμού γνώσης. Διαδραματίζει κρίσιμο ρόλο στο πως κανείς αναπαριστά τις πληροφορίες που θέλει σε ένα γράφο γνώσης αλλά και τι μπορεί να συμπεράνει.

Υπόθεση ανοικτού κόσμου: Η υπόθεση αυτή ορίζει πως ο,τι δεν ορίζεται ως αληθές δεν σημαίνει απαραίτητα πως είναι ψευδές. Άρα η έλλειψη μίας πληροφορίας (από ένα γράφο γνώσης) δεν υπονοεί άρνηση.

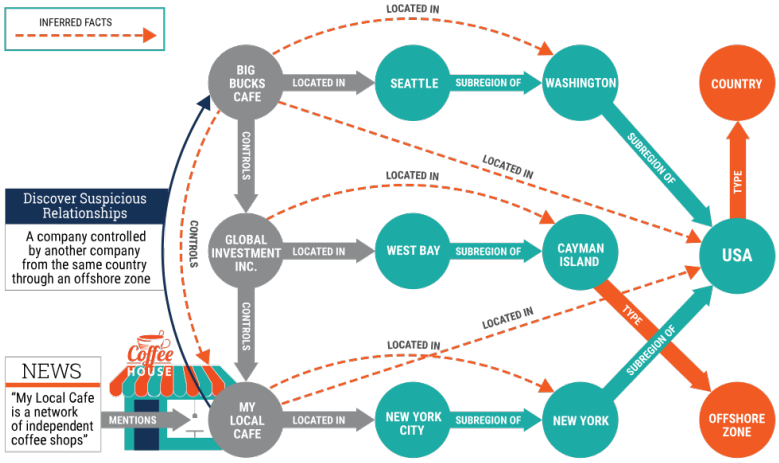
Αντίστοιχα η υπόθεση κλειστού κόσμου δηλώνει πως αν κάτι δεν περιέχεται τότε είναι αναγκαστικά ψευδές.

Στους γράφους γνώσης γενικά ισχύει η Υπόθεση Ανοικτού Κόσμου. Αυτό γεννά διάφορες συνέπειες στην ανάλυση τους. Η βασική συνέπεια είναι πως οι γράφοι γνώσης ποτέ δεν είναι πλήρεις!

Αυτό φυσικά είναι πρακτικά προφανές, καθώς δεν είναι ποτέ δυνατόν να καταγράψει κανείς κάθε πιθανή τριπλέτα που μπορεί να υπάρχει σε κάποιο τομέα (π.χ. Ιατρική). Επίσης θα υπήρχαν ζητήματα αποθηκευτικού χώρου.

Όμως αυτό έχει ως συνέπεια το να μη μπορεί κανείς να κάνει συλλογισμούς με πλήρη ασφάλεια. Πολύ απλά το ότι λείπουν κάποιες τριπλέτες μπορεί να οδηγήσει τα ερωτήματα σε λάθος απαντήσεις. Το ιδανικό θα ήταν να μπορούσε κανείς να κάνει ¨σωστούς¨ συλλογισμούς σε μη πλήρεις γράφους.

Εδώ υπεισέρχεται η μηχανική μάθηση για να βοηθήσει. Με τεχνικές όπως η πρόβλεψη συνδέσμου [14], υπάρχει η δυνατότητα να προστεθούν επιπλέον σύνδεσμοι, ώστε να αλλάξει η δομή του γράφου, άρα και των πιθανών απαντήσεων σε ένα ερώτημα.




Εικόνα 2.1: *Πηγή: https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/*

Όπως μπορεί να δει κανείς στην εικόνα 2.1, υπάρχει ένας μικρός γράφος γνώσης, στον οποίο μπορούν να προβλεφθούν επιπλέον σύνδεσμοι. Για παράδειγμα η σχέση ¨CONTROLS¨ είναι μεταβατική, αφού αν A CONTROLS B, B CONTROLS C τότε A CONTROLS C.

### 2.1.3 Ερωτήματα γράφων

Έχει γίνει πολύ αναφορά σε ερωτήματα στους γράφους γνώσης, όμως δεν έχει δοθεί μία εξήγηση. Τα ερωτήματα αυτά βασίζονται στην έννοια της αντιστοίχησης μοτίβων (pattern matching). Συγκεκριμένα εκφράζει κάποιος μία δομή που επιθυμεί να βρει στο γράφο, και μετά μπορεί να χρησιμοποιήσει τις οντότητες και σχέσεις που έχει επιλέξει για να κάνει διάφορες πράξεις, παρόμοιες πολλές φορές με πράξεις της SQL. Αυτή είναι η λογική γλωσσών για ερωτήματα σε γράφους, όπως η SPARQL [3]. Η παρουσίαση του συντακτικού, αλλά και της γενικότερης λειτουργίας, της SPARQL φεύγει εκτός του σκοπού της διπλωματικής αυτής εργασίας. Όμως, θα δοθούν λίγα παραδείγματα για την κατανόηση κάποιων στοιχείων σχετικά με τη διπλωματική.

> **ΣΗΜΕΙΩΣΗ:** Τα παρακάτω SPARQL ερωτήματα δεν είναι απολύτως συντακτικά σωστά, όμως για κατανόηση της ουσίας γίνεται αφαίρεση κάποιων λεπτομερειών, όπως τα IRI...

#### 1ο Παράδειγμα

```
SELECT ?t WHERE {
  bottleWater costs ?t .
}
```

Το συγκεκριμένο ερώτημα βρίσκει τη πιθανή τιμή ενός μπουκαλιού νερού. Αυτό κάνει μία αντιστοίχηση στον γράφο που μπορεί κανείς να έχει και αναζητά μοτίβα που να ικανοποιούν τις τριπλέτες που περιέχονται στο WHERE.

#### 2ο Παράδειγμα

```
SELECT ?b WHERE {
  Antonis bought ?b .
  Liza near ?b .
}
```

Το παραπάνω ερώτημα αναζητεί κάτι που έχει αγοράσει ο Αντώνης και βρίσκεται δίπλα στη Λίζα. Αν το σκεφτεί κανείς, το ερώτημα αυτό αντιστοιχεί σε ένα μικρό γράφο που έχει 2 συνδέσεις, κάθε μία να ξεκινά από τα δύο άτομα και να καταλήγουν σε ένα κόμβο που συμβολίζει την άγνωστη μεταβλητή ?b.

Αν για παράδειγμα είχαμε την τριπλέτα:

- (Υ: Λίζα) (Κ: είναιΔίπλα) (Α: μπουκάλι νερό)

Τότε, με βάση όλες τις άλλες τριπλέτες που έχουμε αναφέρει προηγουμένως, η απάντηση θα ήταν, μπουκάλι νερό

**3ο Παράδειγμα**

```
SELECT ?t WHERE {
 Antonis bought ?b .
 Liza near ?b .
 ?b costs ?t .
}
```

Το ερώτημα αυτό, αποτελεί ένα γράφο που έχει 2 όμως άγνωστες μεταβλητές ?t, ?b, και επιλέγει να παρουσιάσει μόνο την ?t. Πρακτικά αυτό σημαίνει πως θα βρει όλους τους υπογράφους που έχουν την εν λόγω δομή και θα φέρει πίσω τα αποτελέσματα. Οι μεταβλητές επιτρέπουν την οποιαδήποτε οντότητα να κάνει αντιστοίχηση. Στο παραπάνω ερώτημα η απάντηση θα ήταν 50 λεπτά (το κόστος του μπουκαλιού νερού)

Η παρατήρηση πως μπορεί κανείς να δει τα ερωτήματα σε γράφους ως γράφους καθαυτούς είναι μία από τις βασικές ιδέες που θα χρησιμοποιηθούν σε αυτό το έργο. Η SPARQL, όπως και οι άλλες γλώσσες ερωτημάτων σε γράφους μπορούν να παράξουν ερωτήματα με πολλές δομές. Η διπλωματική αυτή θα ασχοληθεί με ερωτήματα που αντιστοιχούν σε ένα υποσύνολο της πρώτης-τάξεως λογικής, που χρησιμοποιούν μόνο τελεστές σύζευξης και υπαρξιακής ποσοτικοποίησης και ονομάζονται συζευκτικά [36].

για παράδειγμα, μπορεί κανείς να εκφράσει το παράδειγμα 3 ως:

$$t.\exists b : Bought(Antonis, b) \wedge Near(Liza, b) \wedge Costs(b, t) \tag{2.1}$$

Η παραπάνω έκφραση είναι ισοδύναμη με το γράφο που αναφέρθηκε και στο ψευτοκώδικα που παρατέθηκε.

### 2.1.4 Δομή εξεταζομένων ερωτημάτων

Συνοψίζοντας, η διπλωματική εργασία θα απασχοληθεί με ερωτήματα συζευκτικά τα οποία θα έχουν δομή κατευθυνόμενου άκυκλου γράφου (DAG). Οι κόμβοι που έχουν μόνο εξερχόμενες σχέσεις θα ονομάζονται άγκυρες κόμβοι (anchor nodes). Όλοι οι άγκυρες κόμβοι θα είναι γνωστές οντότητες και όχι μεταβλητές. Τα ερωτήματα αυτά θα έχουν μόνο ένα κόμβο που έχει μόνο εισερχόμενες σχέσεις, ο οποίος θα αντιστοιχεί σε μεταβλητή οντότητα και θα αποτελεί την μεταβλητή που προσπαθεί το ερώτημα να απαντήσει (σε αντιστοιχία με το παράδειγμα 3) και θα ονομάζεται καταχρηστικά ρίζα κόμβος (root node). Όλοι οι άλλοι κόμβοι (πέραν των άγκυρα κόμβων και του ρίζα κόμβου) θα είναι μεταβλητές κόμβοι, αλλά δεν απασχολείται το ερώτημα να τους βρει. Τέλος όλες οι σχέσεις είναι γνωστές (ενώ για παράδειγμα γίνεται να έχουμε μεταβλητές σχέσεις ως γενικά ερωτήματα).

## 2.2 Νευρωνικά δίκτυα σε γράφους

Τα νευρωνικά δίκτυα [8] έχουν αποτελέσει την αιχμή του δόρατος στις εξελίξεις του τομέα της Τεχνικής Νοημοσύνης. Το κατ εξοχήν χαρακτηριστικό τους είναι η προσαρμοστικότητα που έχουν να επεξεργαστούν οποιαδήποτε μορφή δεδομένων τους δοθεί, αρκεί φυσικά να προσαρμοστεί η αρχιτεκτονική τους. Στο κλάδο της Υπολογιστικής Όρασης βρήκαν ιδιαίτερη χρήση τα Συνελικτικά νευρωνικά δίκτυα [37], ενώ στις χρονοσειρές βρήκαν θέση τα Αναδρομικά νευρωνικά δίκτυα [38]. Βέβαια, τα τελευταία χρόνια μετά την θεμελιώδη δουλεία των Vaswani κ.α. [39] ο μηχανισμός προσοχής φαίνεται να χρησιμοποιείται σε πολλαπλούς κλάδους πέραν της Επεξεργασίας Φυσικής Γλώσσας, όπως η όραση υπολογιστών [40].

Στη βάση τους, όλες οι αρχιτεκτονικές βασίζονται στην τεχνική του backpropagation [38], όπου η λογική του είναι ο υπολογισμός των παραγώγων των διαφόρων βαρών του δικτύου με βάση το κανόνα αλυσίδας. Στο πως θα αλλάζουν τα βάρη με χρήση των παραγωγών τους ορίζει ένας αλγόριθμος βελτιστοποίησης, όπως ο αλγόριθμος στοχαστικής κατάβασης κλίσης [41], αλλά και πιο καινούργιων όπως ο Adam [42].

Για τους γράφους θα μιλήσουμε στις επόμενες υποενότητες.

### 2.2.1 GNN

Η ιδέα της χρήσης εξειδικευμένων νευρωνικών δικτύων για την ανάλυση δεδομένων μορφής γράφων, έχει γίνει αντιληπτή εδώ και δεκαετίες [12, 13], όμως πολύ πρόσφατα οι προσπάθειες αυτές έπιασαν καρπό. Δύο βασικές έρευνες πάνω στο θέμα που πρέπει να αναφερθούν είναι για τα Συνελικτικά Νευρωνικά Δίκτυα Γράφων (GCN) [43] και τα Νευρωνικά Γράφων με μηχανισμό προσοχής (GAT) [44]. Με τη βοήθεια αυτών και άλλων ερευνών τα νευρωνικά δίκτυα γράφων απέκτησαν αρκετό ερευνητικό ενδιαφέρον και συνεχίζουν ακόμη και σήμερα.

Ένας μη κατευθυνόμενος γράφος $G = (V, E)$ ορίζεται ως ένα σύνολο κόμβων $V$ και ακμών $E$. Κάθε ακμή ορίζεται ως $e_{ij} = (v_i, v_j)$ με $v_i, v_j \in V$. Άρα και περιγράφει τη σύνδεση μεταξύ κόμβων. Στους μη κατευθυνόμενους γράφους ισχύει μάλιστα πως $e_{ij} = e_{ji}$, οπότε η κατεύθυνση δεν έχει σημασία.

Μήτρα Γειτνίασης $A$ ονομάζεται μία μήτρα $(|V|, |V|)$, όπου $|V|$ είναι ο αριθμός των κόμβων σε ένα γράφο, η οποία έχει 1, όποτε δύο κόμβοι συνδέονται με τουλάχιστον μία ακμή και 0 αν δεν συνδέονται μεταξύ τους με μία ακμή, άρα:

$$A_{ij} = \begin{cases} 1, & \text{αν } e_{ij} \in E, \\ 0, & \text{αλλιώς.} \end{cases} \tag{2.2}$$

Ένα ζήτημα με τον ορισμό αυτόν είναι πως έχει εκφραστεί έμμεσα μία διάταξη στους κόμβους, ώστε να θεωρεί ένα κόμβο ως ¨1¨ και έναν άλλο ως ¨2¨ κτλ. Αυτή η διάταξη είναι πλασματική και οφείλει να μην επηρεάζει κάτι σε σχέση με το γράφο! Ο γράφος, ως προς αυτή την αναπαράσταση έχει συμμετρία μετάθεσης. Δηλαδή δεν αλλάζει ο γράφος αν απλώς αλλάξει τη διάταξη που προαναφέρθηκε.

Επίσης μπορεί να οριστεί το σύνολο των γειτονικών κόμβων ενός κόμβου $v$ ως $N(v)$.

Για τη χρήση νευρωνικών δικτύων σε γράφους μπορεί να οριστεί πως κάθε κόμβος έχει ένα άνυσμα χαρακτηριστικών $X_i \in \mathbb{R}^d$. Το ζήτημα είναι πως πάλι υπάρχει έμμεση χρήση διάταξης του κόμβου (με την εισαγωγή δείκτη). Για να συμπεριφέρεται σωστά σε ένα γράφο, ώστε να παράγει καλές αναπαραστάσεις χαρακτηριστικών, οφείλει ένα νευρωνικό να σέβεται τη συμμετρία μετάθεσης. Αυτό ακριβώς πετυχαίνουν τα νευρωνικά δίκτυα γράφων.

Μάλιστα, όλα τα νευρωνικά δίκτυα γράφων έχουν την εξής γενική δομή:

- Διαδοχικά layer που μετασχηματίζουν τα διανύσματα χαρακτηριστικών των κόμβων, σεβόμενα πάντα τη συμμετρία μεταθέσεων
- Aggregation στα χαρακτηριστικά των κόμβων (πάλι σεβόμενο την εν λόγω συμμετρία)

Μία γενική έκφραση για ένα από τα layer είναι η ακόλουθη:

$$h_v^{k+1} = NonLin(Comb(AGG(\{m_u^{k+1} | u \in N(v)\}), m_v^{k+1})) \tag{2.3}$$

όπου

$$m_u^{k+1} = MSG^{k+1}(h_u^k) \tag{2.4}$$
$$m_v^{k+1} = MSG(self)^{k+1}(h_v^k) \tag{2.5}$$

Καταρχήν, οι εξισώσεις [2.4, 2.5] αναφέρονται στην έννοια του μηνύματος του εκάστοτε κόμβου. Η ιδέα είναι πως ο κάθε κόμβος αποκτά πληροφορία από τους γειτονικούς κόμβους με ίδιο τρόπο (δημοκρατικά) μέσο μηνυμάτων [2.4], και χρησιμοποιεί την τοπική πληροφορία από τον εαυτό του [2.5].

Η συνάρτηση *AGG* κάνει aggregation στα γειτονικά μηνύματα, έπειτα η *Comb* συνδυάζει την ομαδοποίηση αυτή με την τοπική πληροφορία [2.5], και μετά χρησιμοποιείται συνήθως ένα μη γραμμικό φίλτρο, όπως *ReLU, Sigmoid*, οπότε και προκύπτει ο υπολογισμός της σχέσης [2.3]. Φυσικά κάθε συνάρτηση σέβεται τη συμμετρία μεταθέσεων, άρα και όλο το layer. Ο κύριος λόγος που συμβαίνει αυτό είναι πως χρησιμοποιούνται μόνο κόμβοι που είναι γείτονες, καθώς η πληροφορία αυτή είναι αναλλοίωτη κάτω από τις μεταθέσεις!

Η φιλοσοφία όλων των νευρωνικών τα τελευταία χρόνια είναι πως όσο περισσότερα layer έχει κανείς τόσο καλύτερα αποτελέσματα, εφόσον φυσικά έχει και πολλά δεδομένα. Όμως φαίνεται πως στα νευρωνικά δίκτυα γράφων έχουν προκύψει προβλήματα στην άμεση και δίχως σκέψη αύξηση του βάθους. Αν κανείς στοιβάξει layer της μορφής [2.3], τότε θα προκύψει το φαινόμενο του oversmoothing [45]. Η γενική ιδέα είναι πως η διαδικασία της μηνυματοδοσίας δίνει πληροφορία σε ένα κόμβο από όλους τους γειτονικούς κόμβους, αν όμως γίνει μία καινούργια θα αποκτήσει και πληροφορία των γειτόνων που έχουν οι γείτονές του. Άρα σε ένα βάθος διαδοχικών layer μορφής [2.3], θα αποκτήσει γνώση για όλο το γράφο, και αν προχωρήσει κανείς σε επιπλέον βάθος όλοι οι κόμβοι θα αποκτήσουν πρακτικά παρόμοια αναπαράσταση με όλους τους άλλους. Έτσι θα γίνει πολύ ομαλή η κατανομή αναπαραστάσεων, εξού και το ¨oversmoothing ¨.

Στη συνέχεια θα γίνει αναφορά στα layer των δύο πιο διαδεδομένων αρχιτεκτονικών στο τομέα των GNN. Για ευκολία ορίζεται $\tilde{N}(v) = N(v) \cup \{v\}$, οπότε το σύνολο αυτό περιέχει το $v$ με τους γείτονες του.

**gcn**

Για ένα Συνελικτικό Δίκτυο Γράφων (gcn) [43], ο μετασχηματισμός ορίζεται ως:

$$h_v^{k+1} = \sigma(\frac{1}{|\tilde{N}_v|} \sum_{u \in \tilde{N}(v)} W_{k+1} h_u^k) \tag{2.6}$$

Η βασική ιδέα εδώ είναι η ίση μεταχείριση όλων των γειτονικών κόμβων του $v$, μαζί με τον κόμβο $v$ ισοδύναμα με τη λογική της συνέλιξης.

**gat**

Για ένα Νευρωνικό Γράφων με μηχανισμό προσοχής (GAT) [44], ο μετασχηματισμός ορίζεται ως:

$$h_v^{k+1} = \sigma(\sum_{u \in \tilde{N}(v)} a_{vu}^{k+1} W_{k+1} h_u^k) \tag{2.7}$$

όπου,

$$a_{vu}^{k+1} = \frac{\exp(e_{vu}^{k+1})}{\sum_{l \in \tilde{N}_v} \exp(e_{vl}^{k+1})} \tag{2.8}$$

οι συντελεστές προσοχής! Μάλιστα ισχύει ότι $\sum_{u \in \tilde{N}_v} a_{vu} = 1$
Τέλος,

$$e_{vu}^{k+1} = LeakyRelU((a^{k+1})^T [W_{k+1} h_v^k || W_{k+1} h_u^k]) \tag{2.9}$$

Προκύπτει πως ο μετασχηματισμός του συνελικτικού [2.6] είναι απλώς η ειδική περίπτωση, όπου $a_{vu} = 1/\tilde{N}_v$. Άρα και δίνεται ισομερής προσοχή σε κάθε κόμβο. Στην γενικότερη περίπτωση της [2.7] γίνεται διάκριση της πληροφορίας που θα πάρει ο κόμβος από κάθε άλλο κόμβο ανάλογα με τις εσωτερικές αναπαραστάσεις που έχει. Συνεπώς, μπορεί σε πολλές περιπτώσεις να αγνοεί ή να προτιμά πληροφορία από συγκεκριμένους κόμβους.

### 2.2.2 RGNN

Σε αντίθεση με τους απλούς γράφους που οι ακμές δεν φέρουν κάποια σημαντική πληροφορία (πέρα φυσικά από την ύπαρξή τους), υπάρχουν οι γράφοι με πολλαπλές σχέσεις. Σε αυτούς τους γράφους, υπάρχει και η πληροφορία του είδους της σύνδεσης και στη περίπτωση των γράφων γνώσης που αναφέρθηκαν στη προηγούμενη ενότητα υπάρχει και η κατεύθυνση ως επιπλέον πληροφορία. Προκύπτει πως τα δίκτυα που αναφέραμε στη προηγούμενη υποενότητα δεν έχουν την εκφραστική ικανότητα να προσφέρουν ικανές και πλούσιες αναπαραστάσεις για τη χρήση σε προβλήματα μηχανική μάθησης. Αυτό μπορεί κανείς να το δει από το γενικό ορισμό ενός GNN layer [2.3] και των μηνυμάτων [2.4, 2.5], καθώς δεν υπάρχει κάπου η πληροφορία του είδους της σύνδεσης, παρά μόνο η ύπαρξή της. Άρα δύο γράφοι που διαφέρουν μόνο στο είδος μίας εκ των συνδέσεων τους θα είχαν την ίδια αναπαράσταση, άρα η πληροφορία έχει χαθεί.

Ακριβώς αυτό το πρόβλημα έρχονται να λύσουν τα σχεσιακά νευρωνικά γράφων (RGNN). Κατά αντιστοιχία με τα GNN, υπάρχουν διαδοχικά layer RGNN που μετασχηματίζουν τα διανύσματα χαρακτηριστικών των κόμβων, σεβόμενα πάλι τη συμμετρία μεταθέσεων. Ανάλογα μετά το πρόβλημα, υπάρχει κάποιος μετασχηματισμός που οφείλει να σέβεται και αυτός τη συμμετρία μεταθέσεων και μετατρέπει αντίστοιχα τη πληροφορία (πχ ολικό aggregation και μετά γραμμικά layer).

Ένα layer RGNN έχει την εξής μορφή:

$$h_v^{k+1} = NonLin(Comb(AGG_{r\in R}(AGG(\{m_{u;r}^{k+1} | u \in N_r(v)\})), m_v^{k+1})) \tag{2.10}$$

όπου

$$m_{u;r}^{k+1} = MSG_r^{k+1}(h_u^k) \tag{2.11}$$

$$m_v^{k+1} = MSG(self)^{k+1}(h_v^k) \tag{2.12}$$

Εδώ ορίζεται το σετ των κόμβων που συνδέονται με το κόμβο $v$, όπου γίνεται διάκριση ως προς το είδος της σχέσης, και τη κατεύθυνση, ως $N_r(v)$, όπου $r$ περιέχει σαν πληροφορία και το είδος της σχέσης καθώς και τη διεύθυνση (έρχεται, εξέρχεται).

Αρχικά γίνεται ένα aggregation ως προς τους κόμβους με σύνδεση ίδιου είδους και διεύθυνσης, και μετά γίνεται aggregation ως προς όλες τις aggregated σχέσεις. Έπειτα γίνεται συνδυασμός πληροφορίας μαζί με την αναπαράσταση του ίδιου κόμβου. Τέλος η αναπαράσταση περνά μία μη γραμμική συνάρτηση.

Το μήνυμα της εξίσωση [2.11] περιέχει τη πληροφορία για τη σχέση ρ, όπου ο εκάστοτε αλγόριθμος αποφασίζει πως θα την εκφράσει. Συνεπώς, με αυτό το τρόπο μπορούν να αναπαρασταθούν όλοι οι κατευθυνόμενοι και πολλαπλών σχέσεων γράφοι, άρα και οι γράφοι γνώσης. Θα γίνει χρήση των δικτύων αυτών σε αυτή την εργασία ως το βασικό κομμάτι της επεξεργασίας. Στη συνέχεια, θα παρουσιαστεί το βασικό συνελικτικό που χρησιμοποιήθηκε.

**rgcn**

Για ένα Σχεσιακό Συνελικτικό Δίκτυο Γράφων (rgcn) [16], ο μετασχηματισμός ορίζεται:

$$h_v^{k+1} = \sigma(\sum_{r \in R} \sum_{u \in \tilde{N}_r(v)} \frac{1}{c_{v,r}} W_r^{k+1} h_u^k) \tag{2.13}$$

όπου το $c_{u;r}$ είτε επιλέγεται ή είναι ικανό να μαθευτεί, αρκεί $\sum_{r \in R} \sum_{u \in \tilde{N}_r(v)} c_{v,r} = 1$

Ο μετασχηματισμός αυτός είναι αρκετά όμοιος με τον μετασχηματισμό [2.6]. Μάλιστα αν υπάρχει μόνο ένα είδους σχέσεων καταρρέει στον μετασχηματισμό αυτό.

# Μέρος II

# Πρακτικό Μέρος

Κεφάλαιο 3

# Ανάλυση και σχεδίαση

Στο κεφάλαιο αυτό παρουσιάζεται η ανάλυση και σχεδίαση της διπλωματικής εργασίας. Συγκεκριμένα, θα γίνει αναφορά στη παραγωγή των δεδομένων Ερωτημάτων-Απαντήσεων, στο πως εκπαιδεύονται και πως τεστάρονται. Τέλος θα γίνει ανάλυση όλων των αλγορίθμων που χρησιμοποιήθηκαν για τα διάφορα μοντέλα.

## 3.1 Παραγωγή Συνόλου Δεδομένων

Σε αυτή την ενότητα θα γίνει αναφορά στη δημιουργία συνόλου δεδομένων, ώστε να γίνει η εκπαίδευση και ο έλεγχος των μοντέλων στο πρόβλημα των ερωτήσεων-απαντήσεων. Για τη παραγωγή δεδομένων ερωτημάτων-απαντήσεων σε γράφους γνώσης, χρειάζεται πρώτα να έχει κανείς σε επεξεργάσιμη μορφή αυτούς τους γράφους. Έγινε χρήση 2 συνόλων δεδομένων, FB15k237 [23], WN18rr [22]. Όλα αυτά ήταν σε αρχεία τριπλετών [2.1.1], ήδη χωρισμένα σε train, test, και valid. Η λογική αυτού του χωρισμού είναι ώστε όλοι να έχουν μία κοινή αρχή στην ανάλυση. Η ιδιότητα που έχουν είναι πως κάθε πιθανή σχέση αλλά και οντότητα βρίσκονται σίγουρα σε τριπλέτες στο train. Μπορεί κανείς να δει την ένωση των 3 αρχείων ως τον ¨πραγματικό γράφο¨, οπότε και προκύπτουν τα 3 αρχεία από μία διαδικασία όπου επιλέγονται τυχαία τριπλέτες από τον ¨πραγματικό γράφο¨ και γεμίζουν τα 3 αρχεία. Συνεπώς, προσπαθεί η διαδικασία αυτή να μιμηθεί την έννοια της μη πληρότητας με τη χρήση για εκπαίδευση του train. Με αυτόν τον τρόπο, η επιτυχία σε πρόβλεψη απαντήσεων στο test αποτελεί μία απόδειξη πως το εκάστοτε μοντέλο μπορεί να αντιμετωπίσει τη μη πληρότητα του γράφου για να απαντήσει σωστά. Βέβαια, ακόμα και ο ¨πραγματικός γράφος¨ είναι μη πλήρης, το οποίο και εγείρει και ερωτήματα για τη διαδικασία καθώς το μοντέλο μπορεί να προβλέπει κάλλιστα και άλλες απαντήσεις, που είναι σωστές όμως επειδή δεν τις έχει το σύνολο δεδομένων (train, test, και valid), θεωρούνται λάθος. Επιπλέον, όπως θα παρουσιαστεί και στη συνέχεια [3.2.2], δεν τίθεται θέμα του απολύτως σωστού αποτελέσματος, αλλά οι προβλέψεις των απαντήσεων να τοποθετούν τις γνωστές σωστές σε επαρκώς υψηλή κατάταξη.

### 3.1.1 Ερωτήματα-Απαντήσεις

Σε αυτή την ενότητα θα γίνει αναφορά στη δημιουργία ερωτημάτων-απαντήσεων με βάση αρχεία τριπλετών. Κύρια πηγή έμπνευσης αποτέλεσε η στρατηγική παραγωγής ερωτημάτων-

απαντήσεων στην εργασία [26], όπου ξεκινώντας από τη κορυφή (root) που αποτελεί και την απάντηση του ερωτήματος, επιλέγει τυχαία ακμές (που όντως υπάρχουν οι ολικοί σύνδεσμοι στο πραγματικό γράφο) και αντίστοιχα καινούργιους κόμβους. Για τη παραγωγή, βέβαια, συγκεκριμένης δομής γράφων ακολουθήθηκαν διαφορετικά μονοπάτια. Στην εργασία [26] χρησιμοποιήθηκε η έννοια της τοπολογικής διάταξης ώστε να εφαρμοστεί μία δομή, ενώ στη διπλωματική εργασία υπήρχε η ελευθερία δημιουργίας οποιασδήποτε δομής γράφου, εφόσον είχε συγκεκριμένο αριθμό ακμών. Στη πρώτη περίπτωση κανείς δίνει μία λίστα από τοπολογικές διατάξεις για να παράξει συγκεκριμένα DAG ερωτήματα, ενώ στη δεύτερη παράγονται διάφορα ερωτήματα και υπάρχουν τεχνικές απόρριψης για την απόκτηση κάποιας συγκεκριμένης DAG δομής. Φυσικά, και στις δύο περιπτώσεις υπάρχουν στρατηγικές απόρριψης, για το να προκύψουν πρώτα γενικά DAG και έπειτα συγκεκριμένης δομής. Η στρατηγική της διπλωματικής είχε τη φιλοσοφία πως μόνο εάν γίνει εκπαίδευση με διάφορες δομές εκπαίδευσης συγχρόνως, μπορεί να αποκτήσουν τα μοντέλα καλύτερη γενίκευση. Βέβαια, έπρεπε να γίνουν προσαρμογές, ώστε να μπορεί να παράξει συγκεκριμένες δομές για να γίνει κάποια σύγκριση με τη βιβλιογραφία.

Τα παραπάνω αφορούν καθαρά τη παραγωγή δεδομένων ως δεδομένα εκπαίδευσης, και όχι επαλήθευσης και ελέγχου. Για τη παραγωγή αυτών των δεδομένων, θα οριστούν οι γράφοι $G_{train}$, $G_{val}$, $G_{test}$ όπου ο πρώτος περιέχει τις τριπλέτες μόνο του train αρχείου, ο δεύτερος του train και valid, και ο τρίτος τις τριπλέτες των train και valid και test αρχείων (άρα είναι ο ¨πλήρης¨ γράφος). Κατά αντιστοιχία ορίζονται τα σύνολα των απαντήσεων ενός ερωτήματος $q$ που προκύπτουν από τις τριπλέτες ενός αρχείου ως $[q]_{train}$, $[q]_{valid}$, $[q]_{test}$. Οι απαντήσεις ερωτήματος $q$ από τα δεδομένα εκπαίδευσης ανήκουν στο σύνολο $[q]_{train}$. Οι απαντήσεις ερωτήματος $q$ για επαλήθευση ανήκουν στο $[q]_{valid}/[q]_{train}$, δηλαδή οι απαντήσεις αυτές ανήκουν στο $[q]_{valid}$ χωρίς να ανήκουν στο $[q]_{train}$! Αντίστοιχα οι απαντήσεις ερωτήματος $q$ για έλεγχο ανήκουν στο $[q]_{test}/[q]_{valid}$. Η λογική αυτών των απαντήσεων είναι πως για να απαντήσει σωστά κάποιος αλγόριθμος οφείλει να χρησιμοποιήσει πρόβλεψη τριπλετών που δεν έχει δει κατά τη διάρκεια μίας εκπαίδευσης και επαλήθευσης, άρα όταν προβλέπει μία απάντηση ελέγχου, πραγματικά πρέπει να χρησιμοποιήσει πληροφορία που δεν έχει εκπαιδευτεί για να πετύχει στη πρόβλεψη. Το στοίχημα είναι πως ακριβώς οι εσωτερικές αναπαραστάσεις που παράγει το κάθε ένα μοντέλο μπορούν να εξηγήσουν αυτήν την ικανότητα απάντησης ερωτημάτων σε μη πλήρεις γράφους.

Ο αλγόριθμος παραγωγής ερωτημάτων-απαντήσεων χωρίζεται σε 4 βασικά μέρη, στην εξαγωγή πληροφορίας της δομής του γράφου, στη τυχαία δημιουργία δομών DAG, στις στρατηγικές απόρριψης και στην συλλογή όλων των απαντήσεων για κάποιο ερώτημα, δεδομένου του γράφου.

### Εξαγωγή Πληροφορίας

Η πρώτη φάση του αλγορίθμου είναι η εξαγωγή πληροφορίας για τις τριπλέτες. Συγκεκριμένα με μία προσπέλαση ενός αρχείου ο αλγόριθμος αποθηκεύει για κάθε οντότητα το σύνολο των σχέσεων που δείχνουν σε αυτή, και για κάθε διπλέτα (σχέσης, οντότητας) το σύνολο των οντοτήτων από τις οποίες ξεκινά η σχέση και δείχνει στην οντότητα της διπλέτας. Η πρώτη δομή ονομάζεται _2relationships και η άλλη _2heads. Επίσης μία άλλη δομή που

αποθηκεύεται ονομάζεται _2tails, η οποία για κάθε διπλέτα (οντότητα, σχέση) αποθηκεύει το σύνολο των οντοτήτων στις οποίες καταλήγει η οντότητα της διπλέτας με τη χρήση της σχέσεως της διπλέτας. Με χρήση αυτών των *key* : *value* δομών θα αναλυθούν τα υπόλοιπα βήματα.

**Δημιουργία DAG**

Σε αυτό το κομμάτι ο αλγόριθμος θα δημιουργήσει ένα ζεύγος DAG ερώτημα και απάντηση ή λόγω των στρατηγικών απόρριψης θα αποτύχει. Με δεδομένο έναν αριθμό ζητούμενων ακμών N που θα πρέπει να έχει το παραγόμενο ερώτημα, αρχικά επιλέγεται τυχαία μία οντότητα ¨α¨ η οποία είναι και η απάντηση του ερωτήματος που είναι να κατασκευαστεί. Στη συνέχεια με χρήση του _2relationships, θα επιλέξει τυχαία μία σχέση από όλες που συνδέονται στην απάντηση, και μετά με τη χρήση του _2heads θα επιλέξει μία οντότητα τυχαία η οποία αποτελεί τριπλέτα με τη σχέση και την πρώτη οντότητα. Για την παραγωγή της επόμενης ακμής, πρώτα θα γίνει τυχαία επιλογή μεταξύ των κόμβων που ήδη υπάρχουν (εδώ 2), και αφότου επιλεχθεί θα ξαναγίνουν όλα τα προηγούμενα βήματα. Αυτό θα επαναλαμβάνεται μέχρι ο αλγόριθμος να έχει παράξει N ακμές ή να έχει ακυρωθεί με μία από τις διάφορες απορρίψεις που θα γίνει αναφορά για όλες στη συνέχεια.

Επίσης, όποτε προκύπτει μία οντότητα ως κορυφή μίας σχέσης θα γίνεται αντιστοίχηση με μία κωδικοποίησή ¨_n¨, όπου n είναι ένας αύξων αριθμός. Συνεπώς, με αυτό το τρόπο όλοι οι κόμβοι που δεν είναι κόμβοι άγκυρες θα έχουν και κωδικοποίησή ¨_n¨. Παρατήρηση, για κάθε ερώτημα η απάντηση κόμβος θα αντιστοιχίζεται με την αναπαράσταση ¨_1¨. Η αναπαράσταση αυτή είναι σημαντική, ώστε να επιβληθεί η δομή του DAG. Τέλος, αν δεν γίνει κάποια απόρριψη ο αλγόριθμος γυρνάει διπλέτα με το ερώτημα σε κωδικοποιημένη μορφή, και την απάντηση του. Η μορφή αυτή προκύπτει με την αντικατάσταση κάθε κόμβου με την αναπαράσταση ¨_n¨ που του αντιστοιχεί. Η αναπαράσταση αυτή του ερωτήματος αποκρύβει όλες τις οντότητες πέρα από τους κόμβους άγκυρες, άρα μπορεί να προκύψει το ίδιο ερώτημα από παραπάνω από μία πηγές, το οποίο είναι και λογικό, αφού τα ίδια ερωτήματα μπορεί να έχουν πολλές απαντήσεις.

Συνοπτικά, εάν ο αλγόριθμος ξεπεράσει όλες τις απορρίψεις, θα παράξει μία απάντηση οντότητα (που εγγυάται την ύπαρξη τουλάχιστον μιας απάντησης) και ένα ερώτημα DAG με μορφή λίστας τριπλετών, όπου οι σχέσεις και οι κόμβοι άγκυρες έχουν τις ονομασίες τους, και όλοι οι άλλοι κόμβοι τις κωδικές ονομασίες ¨_n¨.

**Στρατηγικές Απόρριψης**

Ο σκοπός των στρατηγικών απόρριψης είναι η δημιουργία ερωτημάτων σε μορφή DAG.

- Απόρριψη συνδέσμων που η αρχή και το τέλος είναι η ίδια οντότητα
- Απόρριψη συνδέσμων που έχουν ήδη παραχθεί στο ερώτημα
- Απόρριψη ερωτήματος, όταν προκύπτει κόμβος ως μεταβλητή, ένας κόμβος που στον γράφο, από τον οποίο παράγονται τα ερωτήματα δεν είναι στο τέλος κάποιας σχέσης (αν ορίσουμε όλες τις αντίστροφες συνδέσεις δεν θα υπάρχει αυτή η ακύρωση)

- Απόρριψη συνδέσμου, αν για τη τριπλέτα $(h, r, t)$, ¨_n¨ και το τέλος ¨_m¨ και $n < m$. Η απόρριψη αυτή εγγυάται τη δομή DAG του ερωτήματος.
- Απόρριψη γράφου, αν δεν περιέχει τουλάχιστον ένα σύνδεσμο από τους γράφους valid ή test αν ο σκοπός είναι η παραγωγή ερωτημάτων valid ή test αντίστοιχα.
- Ακύρωση γράφου αν δεν ανήκει σε μία από τις εξεταζόμενες δομές. (Θα γίνει αναφορά στη συνέχεια)
- Ακύρωση διπλών εγγραφών ερώτημα-απάντηση. Δηλαδή, δεν πρέπει να εμφανίζεται στο σύνολο δεδομένων το ίδιο ερώτημα με την ίδια απάντηση πάνω από μία φορά. Αυτό, όμως δεν είναι τόσο προφανές με το ερώτημα γράφο, αφού δύο αναπαραστάσεις λιστών τριπλετών μπορεί να είναι ισοδύναμες! Η λύση δίνεται από τον αλγόριθμο HashQuery, που θα γίνει αναφορά στη συνέχεια.
- Απόρριψη γράφων που περιέχουν στη δομή τους μία σχέση που έχει ως κόμβο tail τον ίδιο κόμβο που έχει η αντίστροφή της ως head. [27]

**Συλλογή απαντήσεων**

Στο τελευταίο βήμα, αφότου έχει αποκτηθεί ένα ερώτημα, θα γίνει συλλογή όλων των απαντήσεων που απαντά το δεδομένο ερώτημα στο γράφο. Αυτό πρέπει να γίνει, ώστε κατά τη διαδικασία του φιλτραρίσματος, που θα αναφερθεί αργότερα, να είναι γνωστές όλες οι απαντήσεις (που τουλάχιστον γίνεται κανείς να ξέρει πως είναι σωστές). Η στρατηγική είναι η εξής. Θα γίνει προσπέλαση του γράφου ερωτήματος ξεκινώντας από τις συνδέσεις που βρίσκονται οι κόμβοι άγκυρας. Αφότου αφαιρεθούν, οπότε και θα προκύψουν νέοι κόμβοι άγκυρα, θα γίνει η ίδια διαδικασία. Αυτό θα γίνει μέχρι να αφαιρεθούν όλες οι συνδέσεις, και η πληροφορία που θα έχει συλλεχθεί, να αντιστοιχεί στις απαντήσεις που ικανοποιεί το εκάστοτε ερώτημα. Κάθε κόμβος άγκυρα, με μία γνωστή σχέση μπορεί μέσω της δομής _2tails, να δώσει ένα σύνολο απαντήσεων, τα οποία θα αποτελέσουν τις πιθανές τιμές που μπορεί να αποκτήσει ένας κόμβος μεταβλητή. Συνεπώς, αφότου έχουν συλλεχθεί όλες οι πιθανές τιμές που ένας κόμβος μεταβλητή μπορεί να λάβει, ο κόμβος αυτός θεωρείται κόμβος άγκυρα, άρα με τις σχέσεις που συνδέεται θα παράξει πάλι με τη χρήση της _2tails, όλες τις πιθανές οντότητες που ο εκάστοτε κόμβος που συνδέεται μπορεί να λάβει. Βέβαια, ένας κόμβος μεταβλητή μπορεί να έχει σύνδεση ως το τέλος της, με πολλαπλές σχέσεις. Συνεπώς, πρέπει να υπολογιστεί η τομή όλων των πιθανών απαντήσεων που μπορεί να λάβει η εκάστοτε μεταβλητή, από τις σχέσεις που υπολογίζει η κάθε μία τους, όπου όλες έχουν στο τέλος τη μεταβλητή. Αφού, για τη παραγωγή του ερωτήματος είχε βρεθεί μία απάντηση (αυτή που ξεκίνησε ο αλγόριθμος) είναι δεδομένο πως η διαδικασία θα βγάλει τουλάχιστον μία απάντηση για το γράφο. Όταν πρέπει να βρεθούν απαντήσεις που πρέπει να έχει χρησιμοποιηθεί κομμάτι της πληροφορίας από τους άλλους γράφους, όλες οι παραγόμενες απαντήσεις πρέπει να έχουν προκύψει από τουλάχιστον ένα σύνδεσμο που δεν βρίσκεται στο σετ εκπαίδευσης, και να μην είναι οι ίδιες οι απαντήσεις και των ίδιων ερωτημάτων στο σετ εκπαίδευσης!

Τέλος, τα παραγόμενα δεδομένα θα είναι σε μορφή ερώτημα-απαντήσεις, όπου με αυτό το τρόπο μπορεί κανείς να ξέρει τις πλήρεις απαντήσεις για κάθε ερώτημα, ανάλογα με το αν είναι στο σύνολο εκπαίδευσης, επαλήθευσης ή ελέγχου.

### 3.1.2 HashQuery

Ο αλγόριθμος HashQuery (κατακερματισμός ερωτήματος) έχει ως σκοπό να αντιστοιχίσει ένα ερώτημα γράφο με ένα μοναδικό hash, άρα δύο γράφοι που η αναπαράσταση τους ως λίστα τριπλετών μπορεί να φέρεται διαφορετικοί, αν είναι ο ίδιος γράφος να έχουν το ίδιο hash, και αν είναι διαφορετικοί τότε να έχουν διαφορετικά hash. Συνεπώς ο αλγόριθμος αυτός προσπαθεί να λύσει το Graph Isomorphism Problem [46]. Φυσικά το πρόβλημα θεωρείται πως δεν είναι πολυωνυμικά επιλύσιμο, όμως στη περίπτωση των πολύ περιορισμένων σε δομή γράφων ερωτημάτων μπορεί να δοθεί ένας αλγόριθμος που είναι γρήγορος και μπορεί να βοηθήσει. Σαν λογική ο αλγόριθμος αποτελεί μία παραλλαγή του αλγορίθμου hash στα Merkle Trees [47], όπου συμπεριλαμβάνεται η πληροφορία του είδους της σχέσης στο hash.

Ο αλγόριθμος χωρίζεται σε δύο βασικά μέρη.

- Συλλογή κλειδιών κόμβων και τιμών διπλετών κόμβου-ακμής που συνδέονται με κατεύθυνση σε κάθε κλειδί κόμβο, στο γράφο του ερωτήματος. Με αυτό το τρόπο γίνεται γνωστή η δομή του γράφου.

- Στη βασική συνάρτηση hash_variable η οποία χρησιμοποιεί την πληροφορία που αναφέρθηκε και ένα δεδομένο κόμβο, γιά να παράξει το hash του κόμβου αυτού.

Η συνάρτηση hash_variable, με την πληροφορία του γράφου που έχει ήδη επεξεργαστεί βρίσκει τη λίστα διπλετών (κόμβος-σχέση) που συνδέονται με τον κόμβο, και για κάθε μία διπλέτα, αν ο κόμβος είναι κόμβος μεταβλητή (άρα είναι σε αναπαράσταση ¨_n¨) τότε καλείται πάλι η συνάρτηση hash_variable αλλά με μεταβλητή το κόμβο αυτόν. Αλλιώς, χρησιμοποιεί την απλή κωδικοποίηση του κόμβου άγκυρα, που είναι ένας αύξων αριθμός μοναδικός για κάθε οντότητα (όπως και στις σχέσεις). Αφότου υπολογιστούν όλα τα hash στη λίστα διπλετών γίνεται μία διάταξη σε αυτές, πρώτα στους κόμβους και μετά ως προς τις σχέσεις. Τέλος, ο αλγόριθμος υπολογίζει το hash της ολικής διατεταγμένης λίστας διπλετών το οποίο και είναι το αποτέλεσμα της συνάρτησης.

(Σημείωση, για να λειτουργήσει ο αλγόριθμος αυτός, πρέπει η συνάρτηση hash που θα χρησιμοποιηθεί για τον υπολογισμό της διατεταγμένης λίστας να γυρίζει έναν αριθμό. Η hash της Python [48] γυρνάει έναν ακέραιο αριθμό, οπότε και λειτουργεί ο αλγόριθμος)

Οπότε ο αλγόριθμος κάνει τα εξής βήματα:

- Πρώτα υπολογίζει την αντιστοίχηση κόμβων με της διπλέτες κόμβου-ακμής που συνδέονται με κατεύθυνση σε αυτόν

- Τη συνάρτηση hash_variable με μεταβλητή τον ίδιο τον κόμβο απάντηση (που έχει κωδικοποίηση ¨_1¨). Άρα το hash του κόμβου απάντησης είναι το hash του ερωτήματος γράφου. Μέσα σε αυτή τη συνάρτηση, λόγω της αναδρομικότητας της, υπολογίζονται τα hash όλων των κόμβων.

Ο αλγόριθμος αυτός κάνει DFS προσπέλαση του γράφου, μαζί με ένα sort όταν όλοι οι κόμβοι έχουν επισκεφτεί ως προς τον εκάστοτε κόμβο που συνδέεται με αυτούς (με τη σωστή κατεύθυνση). Συνεπώς η αλγοριθμική πολυπλοκότητα του οφείλει να είναι κοντά στη

πολυπλοκότητα του DFS, καθώς το εκάστοτε sort γίνεται μόνο σε ένα κομμάτι πάντα από τους κόμβους!

Ο αλγόριθμος αυτός έχει τη δυνατότητα να φέρει ισοδύναμες αναπαραστάσεις του ίδιου γράφου, που είναι σε μορφή λίστας τριπλετών, να έχουν το ίδιο αποτέλεσμα! Η τοπολογική αυτή ιδιότητα δίνεται από το sort που γίνεται σε κάθε κόμβο ως προς τους συνδέσμους των οποίων αποτελεί το τέλος. Άρα λύνεται το πρόβλημα αν δύο γράφοι ερωτήματα είναι ισοδύναμα.

Ο ψευτοκώδικας για τους παραπάνω αλγορίθμους βρίσκεται στο παράρτημα Α΄

## 3.2 Πειράματα

Σε αυτήν την ενότητα θα γίνει αναφορά στο πως οργανώθηκαν οι εκπαιδεύσεις και οι έλεγχοι των μοντέλων. Επίσης θα αναφερθούν οι μετρικές της βιβλιογραφίας.

### 3.2.1 Εκπαίδευση

Η εκπαίδευση ενός μοντέλου είναι αρκετά διαδεδομένη διαδικασία. Χρειάζεται κανείς τα δεδομένα για την εκπαίδευση, καθώς και μικρότερο αριθμό δεδομένων για επαλήθευση valid. Επίσης πρέπει να οριστεί ένας αλγόριθμος βελτιστοποίησης, για τον χειρισμό των παραγώγων των βαρών, στο πως θα αλλάξουν τα βάρη στο backpropagation, στις εκπαιδεύσεις της διπλωματικής χρησιμοποιείται το AdamW [49].

Στόχος της εκπαίδευσης είναι διπλός. Η εκπαίδευση των μοντέλων, ώστε να μπορούν να ¨ενθέσουν¨ ένα ερώτημα γράφο, και επίσης να εκπαιδευτούν οι ενθέσεις όλων των οντοτήτων. Συνεπώς, όταν έχει φτάσει στο τέλος της μία εκπαίδευση, έχουν αλλάξει και τα δύο, ώστε να μπορεί οσοδήποτε να πετύχει το στόχο, που αποτελεί η απάντηση ερωτημάτων.

Κάθε μοντέλο ορίζει ένα δικό του ¨σκορ¨, το οποίο σε αυτή την εργασία έχει γίνει μέριμνα να έχει μέγιστη τιμή τη μονάδα και ελάχιστη την αρνητική μονάδα. Οπότε, αν το σκορ ενός ερωτήματος είναι κοντά στο 1, σημαίνει πως σύμφωνα με το μοντέλο, η απάντηση αυτή αντιστοιχεί στο ερώτημα, και όσο μακρύτερα από το 1, όχι. Στην εκπαίδευση, πέρα από τις διπλέτες ερώτηση-απάντηση που το εκάστοτε μοντέλο πρέπει να τους δώσει υψηλό σκορ, ορίζονται επί τόπου και κάποιες διπλέτες που ονομάζονται διεφθαρμένες (corrupted). Αυτές οι διπλέτες, στη θεωρία πρέπει να αντιστοιχούν σε διπλέτες που δεν ισχύουν, δηλαδή το ερώτημα δεν έχει την απάντηση αυτή. Όμως, το ουσιώδες πρόβλημα της μη πληρότητας των γράφων, έρχεται να αμφισβητήσει αυτή την ιδέα, αφού από τη φύση τους μπορεί να μην ξέρει κανείς πως είναι απαντήσεις, αλλά τελικά να είναι, συνεπώς να μην είναι διεφθαρμένες, κάποιες από τις διπλέτες. Όμως, γίνεται μία βασική υπόθεση για την αντιμετώπιση αυτού του προβλήματος. Εν γένη, ένα μέσο ερώτημα θα έχει πολύ λιγότερες απαντήσεις οντότητες, από ότι συνολικά υπάρχουν οι οντότητες, οπότε και είναι αρκετά μικρή η πιθανότητα, να χρησιμοποιηθούν ως διεφθαρμένες διπλέτες, κανονικές ή όπως υπάρχει και η ονομασία ¨χρυσές¨ (golden). Μάλιστα, λόγω αλγοριθμικής πολυπλοκότητας, αντί για τις αυστηρά διεφθαρμένες, επιλεγόντουσαν για κάθε ερώτημα τυχαία διάφορες οντότητες ως τυχαίες με το ίδιο επιχείρημα. Ως το σφάλμα (loss) για την εκπαίδευση, κάθε μοντέλο ορίζει το δικό του, όμως όλα ακολουθούν τη λογική το σφάλμα να ελαχιστοποιείται, όταν το σκορ των

χρυσών τριπλετών (ή αλλιώς χρυσό σκορ) να μεγιστοποιείται και αντίστοιχα το διεφθαρμένο σκορ να ελαχιστοποιείται! Στα διάφορα πειράματα που θα παρουσιαστούν στη συνέχεια, θα παρακολουθούνται το σφάλμα, το χρυσό σκορ, και το διεφθαρμένο. Τέλος, το διεφθαρμένο σκορ στη πραγματικότητα προκύπτει όχι από μία, αλλά πολλές διεφθαρμένες διπλέτες, οπότε σε μία εκπαίδευση ένα ερώτημα μαθαίνει μία οντότητα να της δώσει υψηλό σκορ, και σε διάφορες τυχαίες χαμηλό. Όμως η συνεισφορά γίνεται κατά μέσο όρο στις διεφθαρμένες, οπότε παρ ελπίδα αν προκύψει κάποια χρυσή (ή κρυφά χρυσή) διπλέτα στις διεφθαρμένες, το μοντέλο λόγω του συντελεστή του μέσου όρου δεν θα μάθει με την ίδια ένταση να την βάλει σε χαμηλό σκορ.

### 3.2.2 Testing και Μετρικές

Αφότου γίνει μία εκπαίδευση ένα μοντέλο οφείλει να ελεγχθεί σε δεδομένα που δεν έχει δει ούτε άμεσα στην εκπαίδευση ούτε και στα δεδομένα επαλίθευσης, τα οποία χρησιμοποιούνται ώστε να επιτευχθεί το finetuning. Συνεπώς, θα γίνει χρήση των test δεδομένων. Στο πρόβλημα των ερωτήσεων-απαντήσεων σε γράφους υπάρχουν συγκεκριμένες μετρικές στη βιβλιογραφία. Στη διπλωματική αυτή έγινε χρήση δύο. Και οι δύο μετρικές χρησιμοποιούν την έννοια της διάταξης των σκορ των οντοτήτων για κάθε απάντηση, και όσο πιο κοντά στη πρώτη θέση βρίσκονται οι σωστές απαντήσεις για κάθε ερώτημα στη πρώτη θέση, τόσο καλύτερα για τις μετρικές αυτές.

#### hits@N

Η μετρική hits@N ορίζεται (εδώ) ως το ποσοστό των διπλετών ερώτημα-απάντηση, όπου η κατάταξη της απάντησης με το σκορ που ορίζει το ερώτημα είναι μέχρι την N-οστή θέση. Η μετρική αυτή θα χρησιμοποιηθεί μόνο στο link prediction

$$Hits@N(q, v) = 1[rank(score(q, v)) < N] \tag{3.1}$$

#### meanRank

Η meanRank μετρική, πολύ απλά εξετάζει το μέσο όρο της διάταξης μίας σωστής απάντησης. Όσο μικρότερη, τόσο καλύτερα! Η μετρική αυτή θα χρησιμοποιηθεί μόνο στο link prediction.

$$mrr(q, v) = \frac{1}{rank(score(q, a))} \tag{3.2}$$

#### hitsGrouped@N

Η μετρική Hits@NGrouped(q) έχει εξάρτηση μόνο ως πρός το ερώτημα και αποτελεί το μέσο όρο του hits@N ως προς κάθε απάντηση του test σετ. Η μετρική αυτή θα χρησιμοποιηθεί μόνο στο query answering.

$$HitsGrouped@N(q) = \frac{1}{|[q]_{test}/[q]_{val}|} \sum_{v \in [q]_{test}/[q]_{val}} 1[rank(score(q,v)) < N] \tag{3.3}$$

**mrrGrouped**

Η μετρική mrrGrouped(q) έχει εξάρτηση μόνο ως πρός το ερώτημα και αποτελεί το μέσο όρο του mrr ως προς κάθε απάντηση του test σετ. Η μετρική αυτή θα χρησιμοποιηθεί μόνο στο query answering.

$$mrrGrouped(q) = \frac{1}{|[q]_{test}/[q]_{val}|} \sum_{v \in [q]_{test}/[q]_{val}} (\frac{1}{rank(score(q,a))}) \tag{3.4}$$

Η διαφορά των απλών από της grouped είναι πως στη πρώτη περίπτωση οι μετρικές θα προκύψουν ως μέσο όρο ως προς ερωτήματα και απαντήσεις του test σετ, ενώ στην άλλη ως ένα μέσο όρο των ερωτημάτων του test σετ, ενός μέσου όρου ως πρός κάθε απάντηση. Αυτή η διαφορά σε μετρικές γίνεται με βάση την εργασία [27].

**Φιλτράρισμα**

Οι παραπάνω μετρικές, σε περίπτωση που κάποιο ερώτημα έχει πολλαπλές απαντήσεις, μπορεί να μειώσουν την απόδοση των μετρικών φαινομενικά, καθώς για κάθε διπλέτα ερώτημα-απάντηση μπορεί οι άλλες απαντήσεις να βρεθούν πιο μπροστά στη κατάταξη των σκορ, από ότι η εξεταζόμενη απάντηση της διπλέτας. Συνεπώς, ορίζεται η διαδικασία του φιλτραρίσματος, όπου για κάθε κατάταξη μίας απάντησης, αφαιρούνται όλες οι άλλες γνωστές απαντήσεις από τη κατάταξη, οπότε και δε θα επηρεάσουν τη μετρική. Η τεχνική αυτή έχει προκύψει από την εργασία του Transe [21].

## 3.3 Αλγόριθμοι

Σε αυτήν την εργασία θα εξεταστούν πολλές αρχιτεκτονικές μοντέλων. Σε αυτή την ενότητα θα γίνει αναφορά σε διάφορες τεχνικές οι οποίες και συνθέτουν τα μοντέλα αυτά.

Προτού όμως γίνει αναφορά στις τεχνικές αυτές, θα παρουσιαστεί ο τρόπος που επεξεργάζονται τα μοντέλα τη κωδικοποιημένη μορφή των ερωτημάτων γράφων. Κατ' αρχήν κάθε οντότητα γνωστή (οντότητες άγκυρες και απάντηση) αποκτά μία συγκεκριμένη ένθεση, η οποία αρχικοποιείται τυχαία. Όπως προαναφέρθηκε, ένθεση αυτή, θα αλλάζει κατά τη διάρκεια της εκπαίδευσης. Όμως, για όλους τους άλλους κόμβους στα ερωτήματα, τα οποία έχουν την αναπαράσταση ¨_n¨, θα αποκτήσουν ως ένθεση, όλα την ίδια η οποία θα είναι το μηδενικό άνυσμα στο χώρο των ενθέσεων. Σε αντίθεση με την εργασία [33], όπου οι άγνωστες μεταβλητές αναπαριστώνται από συγκεκριμένες ενθέσεις ανάλογα με το τύπο οντότητας, εδώ αφού δεν είναι δεδομένο από το είδος των γράφων διαλέχτηκε να αναπαριστώνται από το μηδενικό άνυσμα, το οποίο και θα αλλάξει μετά τις συνελίξεις.

Η γενική μορφή των μοντέλων είναι σε πρότυπο των RGNN [2.2.2], όπου αρχικά οι γράφοι θα επεξεργάζονται από RGNN layers, μετά θα γίνεται ένα βήμα aggregation, τέλος θα υπολογίζεται με κάποια τεχνική η ένθεση του ερωτήματος.

Τα RGNN layers θα είναι rgcn ή rgat. Το πιο απλό είδος μοντέλου, πολύ απλά είναι μία ακολουθία από RGNN layers, μετά μία άθροιση ως προς όλες τις ενθέσεις των κόμβων, μετά μία ακολουθία γραμμικών φίλτρων, όπου το τελευταίο και έχει τις κατάλληλες διαστάσεις για να θεωρηθεί ένθεση του ερωτήματος! Συνεπώς το σκόρ είναι το κανονικοποιημένο εσωτερικό γινόμενο της παραγόμενης ένθεσης του ερωτήματος, με την δεδομένη ένθεση μίας απάντησης.

Με το σετ γνωστών απαντήσεων για ένα ερώτημα $q$ να ορίζεται ως $A(q)$, και $N_c$ ο αριθμός των διεφθαρμένων απαντήσεων, το σφάλμα υπολογίζεται ως:

$$Loss(q, a; m) = \max(m - score(q, a) + \sum_{i=1|a_i \notin A(q)}^{N_c} softmax(\frac{score(q, \tilde{a}_i)}{T_{emb}}) * score(q, \tilde{a}_i), 0) \tag{3.5}$$

όπου $m$ είναι ένα περιθώριο θετικό. Η λογική του σφάλματος είναι η διαφορά του χρυσού σκορ, με του σταθμισμένου μέσου διεφθαρμένου να είναι ανώτερη του περιθωρίου $m$. Επίσης, η χρήση της softmax γίνεται με τη εισαγωγή μίας παραμέτρου θερμοκρασίας $T_{emb}$. Σκοπός της softmax είναι ώστε ο σταθμισμένος μέσος όρος να δίνει προτίμηση στις διεφθαρμένες απαντήσεις με μεγάλο σκορ, καθώς αυτές με το μικρό, δεν έχει λόγο να τις μειώσει.

### 3.3.1 Πολλαπλές Ενθέσεις

Η ιδέα εδώ είναι πως αντί για μια ένθεση, το μοντέλο να βγάζει πολλαπλές, εφόσον κάθε μία θα έχει ένα σκορ (ίδιο για όλες) τότε το σκορ του μοντέλου με μία απάντηση, θα είναι το μέγιστο των σκορ των ενθέσεων με την απάντηση.

Συνεπώς,

$$MultiScore(q, a) = \max_{Emb_i(q)} (Score_i(q, a)) \tag{3.6}$$

### 3.3.2 Root Ένθεση

Η ιδέα εδώ είναι πως, οι συνελίξεις, μαζί με τη δομή του γράφου ερωτήματος, έχουν ως αποτέλεσμα η ένθεση του κόμβου απάντησης να έχει πληροφορία από όλους τους κόμβους. Συνεπώς αντί το aggregation να γίνεται μέσω μίας άθροισης, να γίνεται η επιλογή της τελικής ένθεσης (μετά τα RGNN layers) κόμβου απάντησης, αφού αυτό σέβεται το τοπολογικό χαρακτήρα των RGNN.

### 3.3.3 Χρήση Θερμοκρασίας

Η μέθοδος αυτή στοχέυει στην αλλαγή του σφάλματος. Συγκεκριμένα, στο σφάλμα γίνεται χρήση μίας παραμέτρου T, που ονομάζεται θερμοκρασία. Το σφάλμα σε αντιστοιχία με το σφάλμα του [3.5], ορίζεται ως:

$$Loss(q, a; T) = -T \log[\sigma((score(q, a) - \sum_{i=1|a_i \notin A(q)}^{N_c} softmax(\frac{score(q, \tilde{a}_i)}{T_{emb}}) * score(q, \tilde{a}_i) - m)/T)] \tag{3.7}$$

Το σφάλμα της εξίσωσης [3.7], χρησιμοποιεί τη θερμοκρασία T, και τη σιγμοειδή για την εξομάλυνση του σφάλματος και της εκπαίδευσης. Μάλιστα προκύπτει πως το σφάλμα για το όριο της θερμοκρασίας $T \to 0^+$, γίνεται ανάλογο του [3.5].

## 3.4 Ενθέσεις Κόμβων Ερωτήματος

Στην εργασία [33], οι άγνωστοι είχαν εκφρασμένο τύπο, οπότε και υπήρχε η έννοια ένθεσης τύπου. Στη παρούσα εργασία όλοι οι άγνωστοι (ρίζα, μεταβλητή) θα αποκτήσουν την ίδια αρχική ένθεση, το μηδενικό άνυσμα. Οι κόμβοι άγκυρα αποκτούν τις ενθέσεις των οντοτήτων τους, ενώ στις σχέσεις θα τους ανατεθούν τα βάρη των RGNN. Ο σκοπός των RGNN είναι να δώσουν σε όλους τους κόμβους νέες αναπαραστάσεις στις ενθέσεις τους, ώστε να αποκτήσει το ερώτημα μία αντιπροσωπευτική ένθεση.

Ένα παράδειγμα υπάρχει στο παράτημα Ε΄

Κεφάλαιο 4

# Υλοποίηση

Στο κεφάλαιο αυτό περιγράφεται η υλοποίηση της εργασίας ως προς τα πειράματα. Αρχικά θα γίνει μία μικρή αναφορά στα προγραμματιστικά εργαλεία και στις δομές των γράφων που εξετάστηκαν. Μετά θα γίνει η ανάλυση των πειραμάτων και αποτελεσμάτων της εργασίας. Λεπτομέρειες για το κώδικα στο παράρτημα Δ΄

## 4.1 Προγραμματιστικά εργαλεία

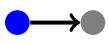
Για την εκπόνηση της εργασίας χρησιμοποιήθηκε η Python [48], μαζί με τη βιβλιοθήκη PyTorch [50] που έχει σχεδιαστεί για την εκπαίδευση νευρωνικών δικτύων. Μάλιστα, για τη χρήση αλγορίθμων των νευρωνικών δικτύων γράφων έγινε χρήση της σχετικά νέας βιβλιοθήκης PyTorchGeometric [51]. Τέλος, για την καταγραφή των πειραμάτων έγινε χρήση της βιβλιοθήκης [52].

Τα πειράματα εκτελέστηκαν με τη χρήση των Google Colab και Kaggle.

## 4.2 Δομές Γράφων

Τα ερωτήματα που επεξεργάστηκε η εργασία αυτή είναι τα βασικά που είναι αντικείμενο μελέτης στη βιβλιογραφία.

### 4.2.1 1p



Εικόνα 4.1: *1p*

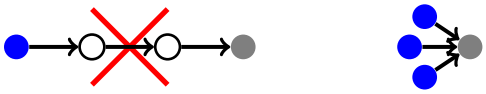
Η εύρεση απαντήσεων σε δομές 1p αποτελεί πλήρως ισοδύναμο πρόβλημα με το link prediction [14]. Όλες οι απαντήσεις προκύπτουν μόνο με τη χρήση των τριπλετών, χωρίς την ανάγκη λογικών πράξεων. Όλες οι υπόλοιπες δομές έχουν μέσα τους λογικές πράξεις και επιπλέον αγνώστους

### 4.2.2 2p, 2i

Εικόνα 4.2: *2p (αριστερά) και 2i (δεξιά)*

Για τα παραπάνω ερωτήματα πρέπει να χρησιμοποιήσει κανείς τη πληροφορία των τριπλετών με διαφορετικό τρόπο. Στη περίπτωση του 2i, οι δύο σύνδεσμοι, μπορεί ο καθένας να έχει πολλαπλές απαντήσεις, αλλά μόνο η τομή των απαντήσεων τους αποτελεί την απάντηση του ερωτήματος. Στη περίπτωση του 2p, ο πρώτος σύνδεσμος δίνει όλες τις απαντήσεις, και ο δεύτερος τις χρησιμοποιεί ως κορυφή για να βρεί την ένωση όλων των απαντήσεων. Γενικά όσες έχουν τομή, περιμένουμε να είναι λιγότερες, από όσες έχουν p. Μάλιστα, για αυτό και συμβολίστηκε στο 2p, ο ενδιάμεσος κόμβος με άλλο σύμβολο, αφού αποτελεί μεν άγνωστο, που δέχεται κάθε οντότητα που ταιριάζει, αλλά δεν καταγράφεται ως απάντηση, αλλά έμμεσα επηρεάζει η ύπαρξη του τον υπολογισμό.

### 4.2.3 3p, 3i



Εικόνα 4.3: *3p (αριστερά) και 3i (δεξιά)*

Η διαφορά των συμπεριφορών θα γίνει πιο έντονη εδώ, αφού οι απαντήσεις περιορίζονται περισσότερο στο 3i και αυξάνονται λόγω ύπαρξης 2 αγνώστων στο 3p.

Στα πλαίσια της διπλωματικής, έγινε η επιλογή να μην αναλυθούν οι δομές 3*p*, καθώς ανεβάζουν πολύ τις υπολογιστικές ανάγκες των πειραμάτων.

### 4.2.4 pi, ip

Εικόνα 4.4: *pi (αριστερά) και ip (δεξιά)*

Οι δομές pi, ip αποτελούν συνδυασμούς των παραπάνω περιπτώσεων και δεν είναι ξεκάθαρη η διαφορά στο πλήθος τους. Οι δομές αυτές δεν θα χρησιμοποιηθούν καθόλου στην εκπαίδευση των μοντέλων, ώστε να εξεταστεί το πόσο είναι ικανά τα μοντέλα να γενικεύσουν σε άλλες δομές που δεν τις έχουν δει κατά την εκπαίδευση.

## 4.3 Πειράματα

Τα πειράματα μπορούν να χωριστούν σε 2 γενικές κατηγορίες. Τα πρώτα σχετίζονται με το απλό link prediction, άρα με τις δομές 1p, και η άλλη κατηγορία είναι η χρήση όλων των δομών ως το γενικότερο πρόβλημα. Στοιχεία για το FB15k_237 υπάρχουν στο παράρτημα Γ'

### 4.3.1 Link Prediction

Στο πρόβλημα του Link Prediction θα γίνει σύγκριση με τις επιδόσεις που αναφέρθηκαν στο Query2Box [27], στις οποίες πέρα από τα δικά τους μοντέλα αναφέρουν και την επίδοση του Transe [21], το οποίο αποτελεί καλό baseline για το πρόβλημα.

Η στρατηγική επιλογής αυτών των αρχιτεκτονικών έγινε με διάφορους πειραματισμούς για ένα από τα μοντέλα (το απλό rgcn) και μετά εξετάστηκαν όλες οι μεταβολές από αυτό σε σχέση με τους αλγόριθμους που αναφέρθηκαν στο προηγούμενο κεφάλαιο. Το βασικό μοντέλο επιλέχθηκε με κριτήριο τη μεγιστοποίηση της *hits@* μετρικής στο σετ επαλήθευσης.

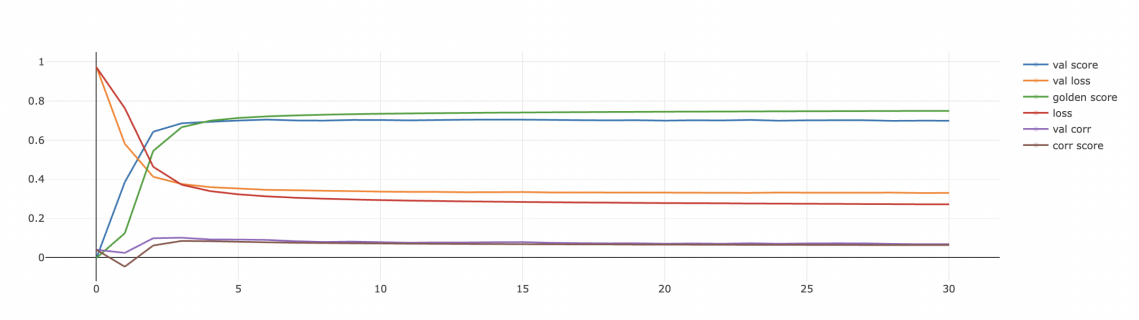
Όλα τα παρακάτω μοντέλα έχουν ως διάσταση ένθεσης 100, και χρησιμοποιούν 50 διεφθαρμένες απαντήσεις, για κάθε μία σωστή, με θερμοκρασία $T_{emb} = 0.25$. Επίσης έχουν ένα layer συνέλιξης διάστασης 300, καθώς και ένα γραμμικό layer για να επαναφέρει τη διάσταση στης ένθεσης. Τα μοντέλα εκπαιδεύτηκαν για μόλις 30 εποχές (στα περισσότερα έγινε χρήση early stopping) με ρυθμό μάθησης 0.0001, batch size 1024, και το *margin* = 1.0. Η εκπαίδευση έγινε (εδώ) μόνο με δομές 1$p$.

Πίνακας 4.1: 1$p$ *αποτελέσματα FB15k_237*

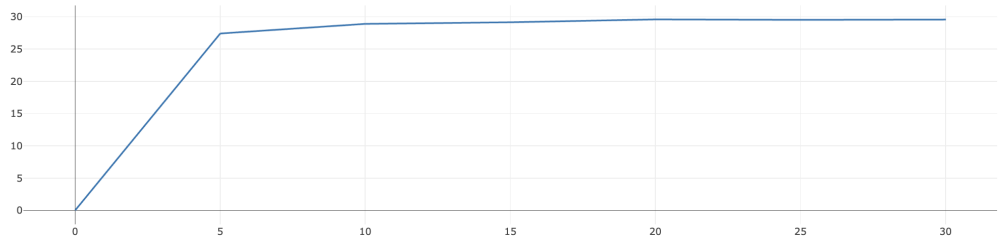
| Μέθοδος | FB15k_237 | |
|---|---|---|
| | hits@3 | MRR |
| rgcn | 33.0 | 30.4 |
| rgat | 32.2 | 29.7 |
| rgcn(root) | 32.2 | 29.6 |
| rgcn(T=0.1) | 33.1 | 30.6 |
| rgcn(M=3) | 32.8 | 30.1 |
| Transe | 31.8 | 28.9 |
| Q2B(1p) | 32.3 | 29.2 |
| Q2B | 33.1 | 29.5 |

Όπως μπορεί να δει κανείς στο πίνακα [4.2] όλα τα μοντέλα κατάφεραν να ξεπεράσουν το baseline Transe και στις 2 μετρικές. Βέβαια, όπως αναφέρθηκε και στο [27], δεν πρέπει να περιμένει κανείς μεγάλες διαφορές επίδοσης, καθώς οι δομές 1$p$ δεν περιέχουν κάποια λογική πράξη. Επίσης, κάποια μοντέλα φαίνεται να τα πήγαν καλύτερα από άλλα σε σχέση με τα Q2B (το Q2B(1p) εκπαιδεύτηκε μόνο με τις 1$p$ δομές) στη μετρική Hits@3, στην MRR όλα ξεπέρασαν και το Q2B μοντέλο, που έχει εκπαιδευτεί σε όλες τις δομές.

Το σχήμα 4.1 περιέχει τα στοιχεία εκπαίδευσης του μοντέλου rgcn(T=0.1). Ήδη από την 5η εποχή, το μοντέλο είχε συγκλίνει σχεδόν σε όλες τις μετρικές. Το σκορ των σωστών απαντήσεων βρίσκεται κοντά στο 0.7-0.8 για τις περισσότερες απαντήσεις, ενώ ο σταθμισμένος μέσος όρος των αρνητικών βρίσκεται λίγο κάτω του 0.1. Η συμπεριφορά των μετρικών εδώ μας δίνει την εντύπωση πως το μοντέλο δεν παρουσιάζει φαινόμενα overfit/underfit.




(α') σκόρ και συνάρτηση σφάλματος



(β') Μετρική hits@3

Σχήμα 4.1: *Εκπαίδευση μοντέλου rgcn(T=0.1)*

Επίσης, χρησιμοποιώντας το αποτέλεσμα του Transe από [53], στο WN18rr σετ δεδομένων, έγινε σύγκριση με την καλύτερη αρχιτεκτονική από τα προηγούμενα, άρα δεν έγινε καν βελτιστοποίηση ως προς τις υπερπαραμέτρους για αυτό το σύνολο δεδομένων. Τα αποτελέσματα στο πίνακα [4.2]

Πίνακας 4.2: *1p αποτελέσματα WN18rr*

| Μέθοδος | WN18rr | |
|---|---|---|
| | hits@3 | MRR |
| rgcn(T=0.1) | 47.2 | 43.5 |
| Transe | 29.5 | 18.2 |

### 4.3.2 Question Answering

Σε αυτή την υποενότητα θα γίνει ανάλυση επιπλέον δομών ερωτημάτων, πέρα από το $1p$. Θα γίνει σύγκριση με τα αποτελέσματα που προσφέρονται στο [27]. Στα πλαίσια της διπλωματικής εργασίας, πάρθηκε η απόφαση να μην χρησιμοποιηθούν οι δομές $3p$, καθώς φαίνεται να ανεβάζουν ιδιαίτερα τις υπολογιστικές ανάγκες της εκπαίδευσης. Το μοναδικό σύνολο δεδομένων που θα αναλυθεί θα είναι το FB15k_237.

Αρχικά θα γίνει η αναφορά στις εκπαιδεύσεις μόνο με τελεστές τομής $(2i, 3i)$ καθώς και τα $1p$. Τέλος θα γίνει ανάλυση εκπαίδευσης που χρησιμοποιούν μία άγνωστη μεταβλητή, δηλαδή τα $2p$. Σε κάθε περίπτωση θα εξεταστούν και οι δομές $pi$ και $ip$, που τα μοντέλα δεν τα έχουν δει στην εκπαίδευση, ούτε στην επαλήθευση. Σε όλα τα δεδομένα επαλήθευσης, ελέγχου (val, test), θα γίνει συλλογή 5000 μοναδικών ερωτημάτων, με όλες τους τις

απαντήσεις.

**intersection only**

Για τα δεδομένα $1p, 2i, 3i$, έγιναν εκπαιδεύσεις πολλών μοντέλων μίας και δύο συνελίξεων. Αρχικά, έγινε αναζήτηση της εξάρτησης της επίδοσης των μοντέλων ανάλογα με τον αριθμό των ερωτημάτων μορφής $2i, 3i$ που έγινε δειγματοληψία.

Τα 3 μοντέλα είχαν την ίδια αρχιτεκτονική με το μοντέλο rgcn(T=0.1), αλλά με διπλάσια διάσταση ένθεσης (200), και εκπαιδεύτηκαν για 20 εποχές το καθένα.

| $\#2i, 3i$ | $1p$ | $2i$ | $3i$ | $pi$ | $ip$ |
|---|---|---|---|---|---|
| $50k$ | 43.7 | 23.4 | 31.4 | 7.05 | 1.0 |
| $100k$ | 43.8 | 26.2 | 36.9 | 8.0 | 1.7 |
| 149689 | 44.0 | 27.3 | 39.0 | 8.0 | 1.0 |

Πίνακας 4.3: *αποτελέσματα hitsGrouped@3 κλιμάκωσης δεδομένων*

| $\#2i, 3i$ | $1p$ | $2i$ | $3i$ | $pi$ | $ip$ |
|---|---|---|---|---|---|
| $50k$ | 40.2 | 21.5 | 28.6 | 7.1 | 0.8 |
| $100k$ | 40.4 | 24.2 | 33.3 | 8.0 | 1.2 |
| 149689 | 40.5 | 24.8 | 35.1 | 8.0 | 1.1 |

Πίνακας 4.4: *αποτελέσματα mrrGrouped κλιμάκωσης δεδομένων*

Όπως μπορεί να δει κανείς, καθώς τα δεδομένα κλιμακώνονται οι επιδόσεις και στις 2 μετρικές αυξάνονται για όλες τις εξεταζόμενες δομές. Μάλιστα στη δομή $1p$, φτάνει ικανοποιητικά επίπεδα, όμως στα άλλα όχι, σε σύγκριση πάντα με τις επιδόσεις των μοντέλων στην εργασία [27]. Επίσης, οι δύο δομές $pi, ip$ δεν φαίνεται να γενικεύουν ικανοποιητικά, ειδικά μάλιστα η $ip$ φαίνεται να μην έχει σχεδόν καμία επιρροή. Φυσικά αυτό μπορεί να εξηγηθεί, καθώς το μοντέλο δεν έχει έρθει σε επαφή με δεδομένα που να χρησιμοποιούν μία μεταβλητή ως άγνωστο (πέρα από το κόμβο απάντηση).

Επιπλέον, ο αριθμός 149689 δεν είναι τυχαίος, καθώς αποτελεί το πλήθος των $1p$ ερωτημάτων που υπάρχουν συνολικά στο FB15k_237 (εφόσον συμπεριληφθούν και οι αντίστροφες σχέσεις).

Στα επόμενα πειράματα θα γίνει χρήση μοντέλων δύο συνελίξεων διάστασης 300. Μάλιστα, όλα τα άλλα χαρακτηριστικά είναι ίδια με το rgcn(T=0.1), με εξαίρεση τη χρήση 150 αρνητικών παραδειγμάτων για ένα θετικό. Τα μοντέλα θα ονομαστούν για ευκολία 2rgcn. Για τα παρακάτω πειράματα που θα αναφερθούν χρησιμοποιήθηκαν σε πλήθος 149689 $1p$, $2i$ και $3i$, μαζί με όλες τους τις απαντήσεις!

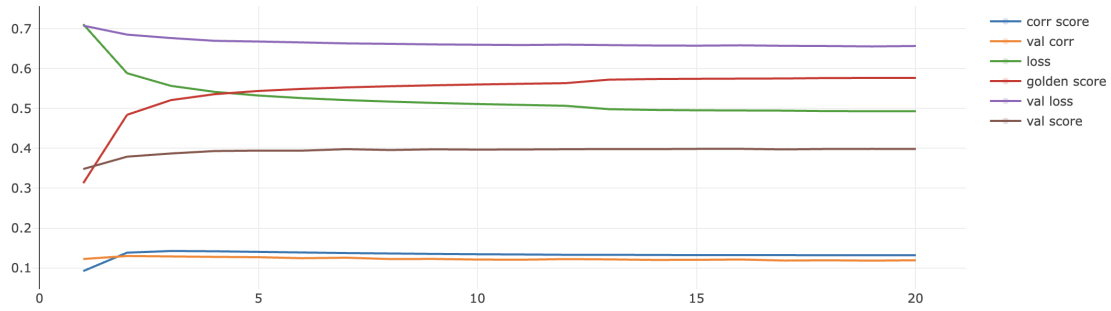
| hitsGrouped@3 | $1p$ | $2i$ | $3i$ | $pi$ | $ip$ |
|---|---|---|---|---|---|
| 2rgcn | 44.8 | 30.2 | 43.1 | 8.6 | 1.2 |
| 2rgcn($T_{emb}$=0.1) | 46.5 | 39.7 | 55.5 | 14.8 | 1.4 |
| Q2B | 46.7 | 32.4 | 45.3 | 20.5 | 10.8 |
| GQE | 40.2 | 29.2 | 40.6 | 17.0 | 8.3 |

Πίνακας 4.5: *αποτελέσματα hitsGrouped@3*

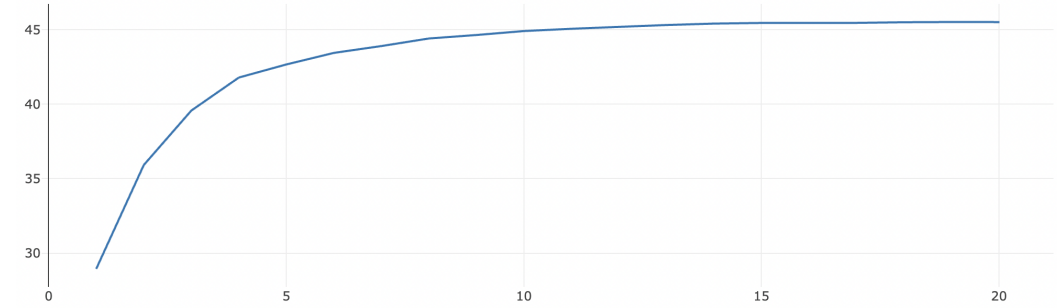
| mrrGrouped | 1$p$ | 2$i$ | 3$i$ | $pi$ | $ip$ |
|---|---|---|---|---|---|
| 2rgcn | 41.1 | 27.6 | 38.6 | 8.6 | 1.2 |
| 2rgcn($T_{emb}$=0.1) | 42.0 | 36.4 | 49.9 | 13.4 | 1.4 |
| Q2B | 40.0 | 27.5 | 37.8 | 18.0 | 10.5 |
| GQE | 34.6 | 25.0 | 35.5 | 15.6 | 8.6 |

Πίνακας 4.6: *αποτελέσματα mrrGrouped*

Εντυπωσιακά η αλλαγή της θερμοκρασίας $T_{emb}$ από 0.25 σε 0.1, φαίνεται να έχει βελτιώσει αρκετά τις επιδόσεις των 2rgcn. Συγκεκριμένα η επίδοση βελτιώθηκε δραματικά στα ερωτήματα τύπου $2i, 3i, pi$. Μάλιστα στη μετρική mrrGrouped, προέκυψε αντίστοιχο προβάδισμα σε όλες τις δομές εκπαίδευσης, όπως και στη προηγούμενη ενότητα. Βέβαια, στη γενίκευση στις δομές $ip, pi$ δεν ξεπέρασαν τα μοντέλα της [27].




(α') σκόρ και συνάρτηση σφάλματος




(β') Μετρική hitsGrouped@3

Σχήμα 4.2: *Εκπαίδευση μοντέλου 2rgcn($T_{emb}$=0.1)*

Ένα από τα μεγαλύτερα προβλήματα σε αυτή τη διπλωματική, αποτέλεσε η συμπεριφορά όλων των ποσοτήτων, με εξαίρεση το hitsGrouped@3, που μετρήθηκε στο σετ επαλήθευσης μόνο κατά τη διάρκεια της εκπαίδευσης (Σχήμα 4.2). Η συμπεριφορά των σκορ και σφάλματος από μόνα τους θα έδινε σε κάποιον την εντύπωση πως η εκπαίδευση του μοντέλου 2rgcn($T_{emb}$=0.1), δεν φαίνεται να συγκλίνει ικανοποιητικά. Μόνο με τη χρήση της υπολογιστικά βαριάς μετρικής του hitsGrouped@3, μπορεί κανείς να έχει κάπως αξιόπιστα αποτελέσματα. Επίσης δεν είναι υπολογιστικά εφικτός ο υπολογισμός του hitsGrouped@3 για το σύνολο της εκπαίδευσης, οπότε δεν μπορεί κανείς να έχει καλή εικόνα για overfit/underfit,

εκτός και αν συμβουλευτεί το σφάλμα, όπου όντως δηλώνει ξεκάθαρα για το 2rgcn($T_{emb}$=0.1) συμπεριφορά overfit. Επιπλέον, σε αντίθεση με το σκορ στις εκπαιδεύσεις 1$p$, στο 0.4 και 0.6 για τις σωστές απαντήσεις επαλήθευσης και εκπαίδευσης! Βέβαια, ο σταθμισμένος μέσος όρος αρνητικών απαντήσεων, πάλι βρίσκεται σε χαμηλό σκορ κοντά στο 0.1, πράγμα που επιτρέπει τη διάκριση μεταξύ σωστών και λάθος απαντήσεων, το οποίο ανταποκρίνεται στη συμπεριφορά της μετρικής hitsGrouped@3 (Σχήμα 4.2β)

**intersection plus 2p**

Οι δομές 2$p$ πολλαπλασιάζουν τις υπολογιστικές ανάγκες των εκπαιδεύσεων. Αυτό συμβαίνει διότι κάθε ερώτημα αυτής της δομής, περιέχει πολλαπλές απαντήσεις, άρα και οι διπλέτες ερώτημα-απάντηση αυξάνονται. Εδώ αναλύθηκαν αποκλειστικά και μόνο συνελίξεις βάθους 2, καθώς περιέχονται δομές βάθους 1 και 2, οπότε και είναι λογικό, σύμφωνα με τη λογική της μεταφοράς μηνυμάτων.

Τα μόνα πειράματα που έγιναν ήταν στη λογική κλιμάκωσης, καθώς η πλήρης μελέτη της εκπαίδευσης θα ήταν μία διαδικασία που κοστίζει πολύ υπολογιστικά, οπότε και χρειάζεται χρήση πολλών *gpus* και ξεφεύγει τα πλαίσια της διπλωματικής. Άρα τα δεδομένα 2$p$ θα είναι λίγα, με τα υπόλοιπα στο μέγιστο πλήθος τους (149689). Η αρχιτεκτονική του μοντέλου που θα αναλυθεί είναι το 2rgcn($T_{emb}$=0.1), όπου και θα εκπαιδευθεί με τα ίδια κριτήρια.

Σε αντίθεση με τα μοντέλα που θα εκπαιδευτούν, τα Q2B, GQE, είχαν στην εκπαίδευση τους το μέγιστο πλήθος 2$p$ (149689), συνεπώς δεν θα έπρεπε κανείς να περιμένει κάποια βελτίωση στις μετρικές, τουλάχιστον ως προς τη δομή 2$p$.

| hitsGrouped@3 | 1$p$ | 2$p$ | 2$i$ | 3$i$ | $pi$ | $ip$ |
|---|---|---|---|---|---|---|
| 2rgcn($T_{emb}$=0.1, 2$p$=10k) | 46.4 | 12.2 | 40.6 | 54.0 | 21.2 | 15.4 |
| Q2B | 46.7 | 24.0 | 32.4 | 45.3 | 20.5 | 10.8 |
| GQE | 40.2 | 21.3 | 29.2 | 40.6 | 17.0 | 8.3 |

Πίνακας 4.7: *αποτελέσματα hitsGrouped@3*

| mrrGrouped | 1$p$ | 2$p$ | 2$i$ | 3$i$ | $pi$ | $ip$ |
|---|---|---|---|---|---|---|
| 2rgcn($T_{emb}$=0.1, 2$p$=10k) | 42.2 | 11.8 | 36.4 | 49.0 | 19.7 | 14.3 |
| Q2B | 40.0 | 22.5 | 27.5 | 37.8 | 18.0 | 10.5 |
| GQE | 34.6 | 19.3 | 25.0 | 35.5 | 15.6 | 8.6 |

Πίνακας 4.8: *αποτελέσματα mrrGrouped*

Τα παραπάνω αποτελέσματα δείχνουν πως με την εισαγωγή πληροφορίας, για την επεξεργασία αγνώστου (2$p$) προέκυψε ικανοποιητική γενίκευση ως προς τις δύο δομές $ip, pi$. Βέβαια, οι δομή 2$p$ χρειάζεται βελτίωση, που θα μπορούσε να επιτευχθεί με την χρήση περισσοτέρων ερωτημάτων. Επίσης όλες οι άλλες δομές είχαν ικανοποιητική συμπεριφορά ως προς τα συγκρίσιμα μοντέλα.

# Μέρος III

# Επίλογος

Κεφάλαιο 5

# Επίλογος

## 5.1 Συμπεράσματα

Κατά την εκπόνηση της εργασίας προέκυψαν διάφορα συμπεράσματα για το γενικότερο πρόβλημα του question answering στους γράφους, καθώς και για την ίδια τη μέθοδο και τα RGNN που αποτέλεσαν το κορμό της μεθόδου, αλλά και για τις τεχνικές ένθεσης.

Καταρχήν, το πρόβλημα question answering στους γράφους έχει το ζήτημα ότι συγχρόνως προσπαθεί να αυξήσει το σκορ των σωστών απαντήσεων για κάποιο ερώτημα, ενώ ζητά να μειωθούν οι άλλες οντότητες, το οποίο έχει το ρίσκο να μειώσει το σκορ των οντοτήτων που δεν είναι γνωστό αν αποτελούν απαντήσεις, με βάση τις τριπλέτες εκπαίδευσης. Οπότε η ελπίδα είναι, η πίεση που δέχονται οι σωστές απαντήσεις στο σύνολο ελέγχου από τη στρατηγική μείωσης να ξεπεραστεί από τη πίεση που δέχεται έμμεσα από το γεγονός ότι οι σωστές οντότητες στο σύνολο εκπαιδεύσεως αποκτούν υψηλό σκορ. Το πρόβλημα αυτό έχει άμεση συνέπεια ποσότητες σαν το σκορ να είναι υψηλό κοντά στο 1 δεν σημαίνει απαραίτητα πως θα έχει καλή συμπεριφορά στη μετρική hits@. Σαν μετρικές, κατά τη διάρκεια της εκπαίδευσης, τα σκορ και τα χρυσά αλλά και τα διεφθαρμένα απλούστατα δεν μπορούν να μεταδώσουν τη πληροφορία πως η εκπαίδευση πηγαίνει αρκετά καλά. Συγκεκριμένα τα διεφθαρμένα, καθώς ανά εποχή και ερώτημα επιλέγονται τυχαία (αρκεί να μην ανήκουν στο σύνολο εκπαίδευσης) δεν δίνουν σε κάποιον τη πλήρη εικόνα για το ποία είναι η συμπεριφορά των όντως αρνητικών απαντήσεων και εύκολα μπορούν οντότητες που δεν ανήκουν στην απάντηση να έχουν παραμείνει με υψηλό σκορ, γιατί απλούστατα λόγο του πλήθους των οντοτήτων μπορεί ο αλγόριθμος να μην έχει προλάβει να μειώσει αυτές τις οντότητες. Επίσης το να υπολογίζει κανείς μετρικές σαν τη hits@ σε κάθε εποχή δεν αποτελεί ρεαλιστική λύση, καθώς είναι υπολογιστικά βαρύ. Συνεπώς πρέπει να αναζητηθούν άλλες μετρικές, ή τουλάχιστον μία διαφορετική προσέγγιση στο πρόβλημα γενικά.

Συγκεκριμένα τώρα για την εργασία, ως προς το link prediction τα rgcn μαζί με το σφάλμα που ορίστηκε, φαίνεται να έχουν τη δυνατότητα να ξεπεράσουν το baseline και να είναι ανταγωνιστικά. Αντίστοιχα στο πρόβλημα του query answering, φαίνεται πως τα μοντέλα μπορούν να έχουν ικανοποιητικές επιδόσεις στις δομές $2i, 3i$, αλλά και στις δομές που δεν εκπαιδεύτηκαν, δηλαδή τα $pi, ip$. Βέβαια για να πετύχει αυτή τη γενίκευση, έπρεπε να γίνει εκπαίδευση με δεδομένα $2p$, ώστε να αντιληφθούν τα μοντέλα πως θα γίνει η μεταφορά πληροφορίας σε ένα ερώτημα που περιέχει αγνώστους, πέρα από το κόμβο απάντησης. Μάλιστα, δεν χρειάστηκε μεγάλο σύνολο δεδομένων $2p$, για να επιτευχθεί η γενίκευση. Φαίνεται

πως τα RGNN μπορούν να χρησιμοποιηθούν επιτυχώς για την ένθεση λογικών ερωτημάτων σε γράφους γνώσης, και επιδέχονται επιπλέον μελέτης και ανάλυσης σε αυτό και σε άλλα προβλήματα

## 5.2 Μελλοντικές Επεκτάσεις

Η τεχνικές που αναπτύχθηκαν σε αυτή την εργασία μπορούν να επεκταθούν με πολλούς τρόπους για τη λύση του προβλήματος question answering.

Αρχικά, οφείλει να γίνει μία ανάλυση των υπολοίπων ερωτημάτων μορφής $2p$, αλλά και επέκταση στις δομές $3p$. Φυσικά, πρέπει να γίνει μία ενδελεχής έρευνα στο χώρο των παραμέτρων, τόσο της εκπαίδευσης αλλά και των αρχιτεκτονικών, με χρήση επαρκούς υπολογιστικής δύναμης. Στη συνέχεια, μπορεί να γίνει αναζήτηση σε πιο εξωτικές μεθόδους ένθεσης, όπως οι ενθέσεις κουτιού (box). Επιπλέον, η έρευνα πρέπει να γίνει και σε περισσότερα σύνολα δεδομένων για τους ελέγχους επίδοσης, αλλά ακόμη και να γίνει επέκταση σε ανεξερεύνητες (μέχρι στιγμής) δομές ερωτημάτων, τόσο ως προς την ίδια τη δομή, αλλά και με χρήση επιπλέον τελεστών όπως της άρνησης NOT, για τη δημιουργία καινούργιων ερωτημάτων. Στο ίδιο πνεύμα, μπορεί να γίνει εξειδικευμένη μελέτη στη συμπεριφορά ¨μεγάλων¨ δομών. Επιπλέον, σύμφωνα και με τα συμπεράσματα, θα ήταν ενδιαφέρουσα η αναζήτηση νέων μετρικών για την ανάλυση του προβλήματος. Τέλος, θα μπορούσε να γίνει αναζήτηση σε ερωτήματα που έχουν παραπάνω από ένα άγνωστο, ή ακόμα και άγνωστη σχέση ή και συνδυασμό, αφού όλα αυτά είναι επιτρεπτά ερωτήματα!

# Παραρτήματα

## Παράρτημα Α΄

# Ψευτοκώδικας HashQuery

Αλγοριθμος Α.1: *HashQuery algorithm*

```
function HashQuery(query)
    paths ← parse_graphs(query)
    return hash_variable(paths,"_1")
end function
```

Αλγοριθμος Α.2: *parse_graphs algorithm*

```
function parse_graph(query)
    Initialize dictionary paths
    for triple in query do
        h, r, t ← triple
        if t not in paths.keys() then
            path[t] = set()
        end if
        paths[t].add((h, r))
    end for
    return paths
end function
```

Αλγοριθμος Α.3: *hash_variable algorithm*

```
function hash_variable(paths, tail)
    Initialize connections list
    for double in paths[tail] do
        h, r ← double
        if not isinstance(h, int) then          ▷ If h is not an anchor node
            h ← hash_variable(paths, h)
        end if
        connections.append((h, r))
    end for
    connections.sortby(h, r)
    return hash(connections)        ▷ hash must be a hashing function returning integer
end function
```

# Παράρτημα Β΄

## Όριο $T \to 0^+$

Για ευκολία ορίζεται:

$$X = score(q, a) - \frac{1}{N_c} \sum_{i=1|a_i \notin A(q)}^{N_c} softmax(\frac{score(q, \tilde{a}_i)}{T_{emb}}) * score(q, \tilde{a}_i) - m \tag{Β΄.1}$$

Συνεπώς, το όριο της σχέσης [3.7], εφόσον $X < 0$, γίνεται:

$$\lim_{T \to 0^+} Loss(q, a; T) = \lim_{T \to 0^+} -T \log(\sigma(X/T)) \tag{Β΄.2}$$

$$= \lim_{T \to 0^+} T \log(1 + \exp(-X/T)) \tag{Β΄.3}$$

$$= \lim_{T \to 0^+} T \log(\exp(-X/T)) \tag{Β΄.4}$$

$$= \lim_{T \to 0^+} -TX/T = -Q \tag{Β΄.5}$$

για $X > 0$

$$\lim_{T \to 0^+} Loss(q, a; T) = \lim_{T \to 0^+} -T \log(\sigma(X/T)) \tag{Β΄.6}$$

$$= \lim_{T \to 0^+} T \log(1 + \exp(-X/T)) \tag{Β΄.7}$$

$$= 0 \tag{Β΄.8}$$

Άρα

$$\lim_{T \to 0^+} Loss(q, a; T) = \lim_{T \to 0^+} \max(-X, 0) \tag{Β΄.9}$$

$$= \max(m - score(q, a) + \frac{1}{N_c} \sum_{i=1|a_i \notin A(q)}^{N_c} softmax(\frac{score(q, \tilde{a}_i)}{T_{emb}}) * score(q, \tilde{a}_i), 0) \tag{Β΄.10}$$

Η οποία σχέση είναι ίδια με τη [3.5]

Παράρτημα Γ΄

# Στατιστικά Δεδομένων FB15k_237

| Δομες | # Ερωτημάτων | Μέσες QA διπλέτες |
|---|---|---|
| 1$p$ | 149689 | 3.64 |
| 2$p$ | 10000 | 259.98 |
| 2$i$ | 149689 | 5.87 |
| 3$i$ | 149689 | 4.38 |

Πίνακας Γ΄.1: *Στατιστικά train δεδομένων*

| Δομες | # Ερωτημάτων | Μέσες QA διπλέτες |
|---|---|---|
| 1$p$ | 20101 | 1.74 |
| 2$p$ | 5000 | 85.73 |
| 2$i$ | 5000 | 4.73 |
| 3$i$ | 5000 | 2.58 |

Πίνακας Γ΄.2: *Στατιστικά valid δεδομένων*

| Δομες | # Ερωτημάτων | Μέσες QA διπλέτες |
|---|---|---|
| 1$p$ | 149689 | 1.79 |
| 2$p$ | 5000 | 90.20 |
| 2$i$ | 5000 | 5.60 |
| 3$i$ | 5000 | 3.49 |
| $pi$ | 5000 | 29.09 |
| $ip$ | 5000 | 140.09 |

Πίνακας Γ΄.3: *Στατιστικά test δεδομένων*

Οι παραπάνω στατιστικές εκφράζουν τη διακύμανση των ερωτημάτων, ανάλογα με τη δομή του ερωτήματος. Είναι λογικό, οι δομές 2$p$, να έχουν περισσότερες απαντήσεις σε σχέση με τις άλλες δομές, αφού οποιαδήποτε ενδιάμεση οντότητα, ταιριάζει στο μοτίβο, που η δομή 2$p$ αναζητά, χρησιμοποιείται για την εύρεση των ερωτημάτων, οπότε και μπορεί να το δει κανείς σαν να αυξάνεται τετραγωνικά ως προς το πλήθος των ερωτημάτων δομής 2$p$. Βέβαια, αυτό κάνει την εκπαίδευση με όλες τις απαντήσεις των ερωτημάτων 2$p$, ακόμα πιο υπολογιστικά βαρύ!

Παράρτημα Δ΄

# Κώδικας Διπλωματικής

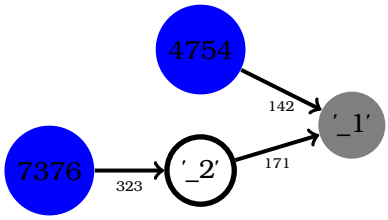
Ο κώδικας της διπλωματικής, από τη παραγωγή δεδομένων στην εκπαίδευση και έλεγχο των μοντέλων καθώς και οι αρχιτεκτονικές τους και όλοι οι λοιποί αλγόριθμοι, μπορούν να βρεθούν εδώ: https://github.com/tolios/query2vec

```
query2vec
├── Dockerfile
├── README.md
├── __init__.py
├── algorithms
│   ├── MDRTkernel
│   ├── __init__.py
│   └── base.py
├── config.py
├── create_filter.py
├── endpoint.sh
├── environment.yml
├── example
│   ├── model.json
│   └── train_config.json
├── form.py
├── gpu_colab.ipynb
├── gpu_kaggle/ipynb
├── graph.py
├── main.py
├── metrics.py
├── requirements.txt
├── run.py
├── run_tests.py
├── test.py
├── train.py
└── utils
```

Παράρτημα Ε΄

# Ενθέσεις ερωτήματος γράφου

Στη συνέχεια παρουσιάζεται ένα παράδειγμα ενός παραγόμενου ερωτήματος για το σύνολο δεδομένων FB15k_237, μορφής *pi*:




Εικόνα Ε΄.1: *Παράδειγμα αρχικών ενθέσεων*

Οι άγνωστοι κόμβοι _1, _2 θα λάβουν αρχικές ενθέσεις ένα μηδενικό άνυσμα ($[0, 0, ..., 0]$), με διάσταση που καθορίζεται από την αρχιτεκτονική του μοντέλου. Οι οντότητες 7376 και 4754 θα λάβουν ενθέσεις που θα καθοριστούν από την εκπαίδευση του μοντέλου!

Οι ταυτότητες των οντοτήτων:

- 4754 - "/m/0dryh9k" - Indians
- 7376 - "/m/0d8rs" - Groningen

Οι ταυτότητες των σχέσεων:

- 171 - "/people/person/spouse_s./people/marriage/type_of_union/-1"
- 142 - "/people/ethnicity/people"
- 323 - "/people/marriage_union_type/unions_of_this_type./people/marriage/location_of_ceremony/-1"

Προσοχή: Στις σχέσεις 171, 323 υπάρχει η προσθήκη ¨/-1¨, η οποία υποδηλώνει αντίστροφη σχέση και αποτελεί ονομασία της διπλωματικής, πάνω στην ήδη υπάρχουσα ονομασία (που είναι το υπόλοιπο κομμάτι).

# Βιβλιογραφία


[1] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen και Philip S. Yu. *A Survey on Knowledge Graphs: Representation, Acquisition, and Applications*. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[2] Ian Robinson, Jim Webber και Emil Eifrém. *Graph Databases*. O'Reilly Media, 2013.

[3] W3C. *SPARQL Query Language*. https://www.w3.org/TR/sparql11-query/.

[4] Inc. Neo4j. *Cypher Query Language*. https://neo4j.com/docs/cypher-manual/current/.

[5] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong και Qing He. *A Survey on Knowledge Graph-Based Recommender Systems*, 2020.

[6] Thomas Rincy N και Roopam Gupta. *A Survey on Machine Learning Approaches and Its Techniques:*. *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, σελίδες 1–6, 2020.

[7] Yann LeCun, Y. Bengio και Geoffrey Hinton. *Deep Learning*. *Nature*, 521:436–44, 2015.

[8] David E. Rumelhart, Geoffrey E. Hinton και Ronald J. Williams. *Learning representations by back-propagating errors*. *Nature*, 323(6088):533–536, 1986.

[9] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh και Angela Hung Byers. *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. *McKinsey Global Institute*, 2011.

[10] Rajat Raina, Anand Madhavan και Andrew Y. Ng. *Large-scale Deep Unsupervised Learning using Graphics Processors*. *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, σελίδες 873–880, 2009.

[11] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht και Oriol Vinyals. *Understanding deep learning requires rethinking generalization*. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[12] Marco Gori, Gabriele Monfardini και Franco Scarselli. *A new model for learning in graph domains*. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2:729–734 ολ. 2, 2005.

[13] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner και Gabriele Monfardini. *The Graph Neural Network Model*. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[14] David Liben-Nowell και Jon Kleinberg. *The Link Prediction Problem for Social Networks*. *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, σελίδα 556–559, New York, NY, USA, 2003. Association for Computing Machinery.

[15] Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann και Asja Fischer. *Introduction to neural network-based question answering over knowledge graphs*. *WIREs Data Mining and Knowledge Discovery*, 11(3):ε1389, 2021.

[16] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Riannevan den Berg, Ivan Titov και Max Welling. *Modeling Relational Data with Graph Convolutional Networks*, 2017.

[17] Dan Busbridge, Dane Sherburn, Pietro Cavallo και Nils Y. Hammerla. *Relational Graph Attention Networks*, 2019.

[18] Tomas Mikolov, Kai Chen, Greg Corrado και Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*, 2013.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado και Jeffrey Dean. *Distributed Representations of Words and Phrases and their Compositionality*, 2013.

[20] Rodolphe Jenatton, Nicolas Roux, Antoine Bordes και Guillaume R Obozinski. *A latent factor model for highly multi-relational data*. *Advances in Neural Information Processing Systems*F. Pereira, C.J. Burges, L. Bottou και K.Q. Weinberger, επιμελητές, τόμος 25. Curran Associates, Inc., 2012.

[21] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston και Oksana Yakhnenko. *Translating Embeddings for Modeling Multi-Relational Data*. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, σελίδα 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

[22] George A. Miller. *WordNet: A Lexical Database for English*. *Commun. ACM*, 38(11):39–41, 1995.

[23] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge και Jamie Taylor. *Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge*. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, σελίδα 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery.

[24] Zhen Wang, Jianwen Zhang, Jianlin Feng και Zheng Chen. *Knowledge Graph Embedding by Translating on Hyperplanes*. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, σελίδα 1112–1119. AAAI Press, 2014.

[25] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu και Xuan Zhu. *Learning Entity and Relation Embeddings for Knowledge Graph Completion*. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, σελίδα 2181–2187. AAAI Press, 2015.

[26] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky και Jure Leskovec. *Embedding Logical Queries on Knowledge Graphs*. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, σελίδα 2030–2041, Red Hook, NY, USA, 2018. Curran Associates Inc.

[27] Hongyu Ren*, Weihua Hu* και Jure Leskovec. *Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings*. *International Conference on Learning Representations*, 2020.

[28] Bishan Yang, Wentau Yih, Xiaodong He, Jianfeng Gao και Li Deng. *Embedding Entities and Relations for Learning and Inference in Knowledge Bases*, 2015.

[29] Vinh Thinh Ho, Daria Stepanova, Mohamed H. Gad-Elrab, Evgeny Kharlamov και Gerhard Weikum. *Learning Rules from Incomplete KGs using Embeddings*. *International Workshop on the Semantic Web*, 2018.

[30] Fan Yang, Zhilin Yang και William W. Cohen. *Differentiable Learning of Logical Rules for Knowledge Base Reasoning*, 2017.

[31] Dimitrios Alivanistos, Max Berrendorf, Michael Cochez και Mikhail Galkin. *Query Embedding on Hyper-relational Knowledge Graphs*, 2022.

[32] Federico Bianchi, Gaetano Rossiello, Luca Costabello, Matteo Palmonari και Pasquale Minervini. *Knowledge Graph Embeddings and Explainable AI*. *Knowledge Graphs for eXplainable Artificial Intelligence*, 2020.

[33] Daniel Daza και Michael Cochez. *Message Passing Query Embedding*, 2020.

[34] Petar Ristoski, Gerben Klaas Dirkde Vries και Heiko Paulheim. *A Collection of Benchmark Datasets for Systematic Evaluations of Machine Learning on the Semantic Web*. *The Semantic Web - ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, σελίδα 186–194, Berlin, Heidelberg, 2016. Springer-Verlag.

[35] Dean Allemang, Jim Hendler και Fabien Gandon. *Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL*, τόμος 33. Association for Computing Machinery, New York, NY, USA, 3η έκδοση, 2020.

[36] Serge Abiteboul, Richard Hull και Victor Vianu. *Foundations of Databases: The Logical Level*. Addison-Wesley Longman Publishing Co., Inc., USA, 1στη έκδοση, 1995.

[37] Yann LeCun, Léon Bottou, Yoshua Bengio και Patrick Haffner. *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[38] David E. Rumelhart, Geoffrey E. Hinton και Ronald J. Williams. *Learning Representations by Back-propagating Errors*. *Nature*, 323(6088):533–536, 1986.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser και Illia Polosukhin. *Attention is All You Need*. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, σελίδα 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit και Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2021.

[41] Sebastian Ruder. *An overview of gradient descent optimization algorithms*, 2016.

[42] Diederik P. Kingma και Jimmy Ba. *Adam: A Method for Stochastic Optimization*, 2014.

[43] Thomas N. Kipf και Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. *International Conference on Learning Representations*, 2017.

[44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò και Yoshua Bengio. *Graph Attention Networks*, 2018.

[45] Qimai Li, Zhichao Han και Xiao Ming Wu. *Deeper insights into graph convolutional networks for semi-supervised learning*. *Proceedings of the AAAI conference on artificial intelligence*, τόμος 32, 2018.

[46] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2νδη έκδοση, 2005.

[47] Ralph C. Merkle. *A Certified Digital Signature*. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, τόμος 435 στο *Lecture Notes in Computer Science*, σελίδες 218–238. Springer, 1989.

[48] Python Software Foundation. *Python*. 2001.

[49] Ilya Loshchilov και Frank Hutter. *Decoupled Weight Decay Regularization*, 2019.

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai και Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. *Advances*

*in Neural Information Processing Systems 32*H. Wallach, H. Larochelle, A. Beygelzimer, F.d' Alché-Buc, E. Fox και R. Garnett, επιμελητές, σελίδες 8024–8035. Curran Associates, Inc., 2019.

[51] Matthias Fey και Jan Eric Lenssen. *Fast Graph Representation Learning with PyTorch Geometric*, 2019.

[52] MLflow Community. *MLflow: A Platform for Managing the Machine Learning Lifecycle*. https://mlflow.org/.

[53] Takuma Ebisu και Ryutaro Ichise. *Generalized Translation-Based Embedding of Knowledge Graph*. *IEEE Transactions on Knowledge and Data Engineering*, ΠΠ:1–1, 2019.

# Απόδοση ξενόγλωσσων όρων

| Απόδοση | Ξενόγλωσσος όρος |
|---|---|
| Υπόθεση Ανοιχτού Κόσμου | Open World Assumption |
| διπλέτες | doubles |
| τριπλέτες | triplets |
| αντιστοίχηση μοτίβων | pattern matching |
| πρώτης-τάξεως λογική | first order logic |
| τελεστής σύζευξης | conjuction operator |
| υπαρξιακή ποσοτικοποίηση | existential quantification |
| ένθεση | embedding |
| γράφος | graph |
| κατευθυνόμενος άκυκλος γράφος | directed acyclic graph |
| μήτρα γειτνίασης | adjacency matrix |
| ανάκτηση πληροφορίας | information retrieval |
| αντιμεταθετικότητα | commutativity |
| απόγονος | descedant |
| βάση γνώσης | knowledge base |
| γράφος γνώσης | knowledge graph |
| βάση δεδομένων | database |
| οντότητα | entity |
| ερώτημα | query |
| νευρωνικό δίκτυο | neural network |
| στοχαστική κατάβαση κλίσης | stochastic gradient descent |
| σχεσιακό | relational |
| αντίστροφο | inverse |
| κόμβος άγκυρα | anchor node |
| κόμβος ρίζα | root node |
| κόμβος μεταβλητή | variable node |
| κατακερματισμός | hash |
| υπερπαράμετροι | hyperparameters |
| πρόβλεψη συνδέσμου | link prediction |
| απάντηση ερωτημάτων | question answering |
| εκπαίδευση | train |
| επαλήθευση | validation |
| έλεγχος | test |
| συνέλιξη | convolution |