Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας, Πληροφορικής & Υπολογιστών

Εμπλουτισμός δεδομένων και ερμηνεία ταξινομητών Βαθιάς Μάθησης με τη χρήση Περιγραφικών Λογικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ-ΙΑΣΩΝ ΛΙΑΡΤΗΣ

Επιβλέπων : Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας, Πληροφορικής & Υπολογιστών

Εμπλουτισμός δεδομένων και ερμηνεία ταξινομητών Βαθιάς Μάθησης με τη χρήση Περιγραφικών Λογικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ-ΙΑΣΩΝ ΛΙΑΡΤΗΣ

Επιβλέπων : Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 15η Ιουλίου 2021.

Ανδρέας-Γεώργιος Σταφυλοπάτης	Στέφανος Κόλλιας	Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

Γεώργιος-Ιάσων Λιάρτης

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

Περίληψη

Η ραγδαία ανάπτυξη που έχει γνωρίσει ο κλάδος της Βαθιάς Μάθησης την τελευταία δεκαετία έχει οδηγήσει στην ευρεία εφαρμογή του για την αυτοματοποίηση διαδικασιών. Όμως, η αδιαφάνεια των μοντέλων που παράγει δημιουργεί ηθικά ζητήματα και επιφυλάξεις στη χρήση του σε κρίσιμες εφαρμογές, όπως οι ιατρικές, παρά τις σημαντικές του επιδόσεις. Εμείς κινούμαστε στον χώρο της Εξηγήσιμης Τεχνητής Νοημοσύνης και χρησιμοποιούμε εργαλεία από τις Περιγραφικές Λογικές, εύκολα ερμηνεύσιμα από τη φύση τους, για να διαλευκάνουμε τον τρόπο με τον οποίο λαμβάνουν αποφάσεις αυτά τα μοντέλα. Συγκεκριμένα, αντιμετωπίζουμε ταξινομητές ως μαύρα κουτιά και χρησιμοποιούμε δεδομένα εμπλουτισμένα με σημασιολογικές περιγραφές για να παράγουμε συζευκτικά ερωτήματα που μιμούνται τη συμπεριφορά τους.

Λέξεις κλειδιά

Οντολογίες, Βάσεις Γνώσης, Περιγραφικές Λογικές, Εξηγήσιμη Τεχνητή Νοημοσύνη, Συζευκτικά Ερωτήματα, Μηχανική Μάθηση, Βαθιά Μάθηση

Abstract

The rapid growth Machine Learning has met in the last decade has lead to its wide application for automating processes. However, the opaqueness of the models it produces is creating issues of morality and reservations in its usage for critical applications, such as medical ones, despite its significant performance. We work in the area of Explainable Artificial Intelligence and use tools from Description Logics, easily interpretable by their nature, to disambiguate the way in which these models make decisions. In particular, we treat classifiers as black boxes and we use data enriched with semantic descriptions to produce conjunctive queries that mimic their behavior.

Key words

Ontologies, Knowledge Bases, Description Logics, Explainable Artificial Intelligence, Conjunctive Queries, Machine Learning, Deep Learning

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

Κεφάλαιο 1

Εισαγωγή

Μετά από μία σειρά από επιτυχίες που σημείωσε το 2012, ο κλάδος της Βαθιάς Μάθησης έχει γνωρίσει ανανεωμένο ενδιαφέρον, και από την επιστημονική κοινότητα, και από τον επιχειρησιακό τομέα. Η λίστα των επιτευγμάτων του διαρκώς μεγαλώνει, καθώς βρίσκει εφαρμογές σε όλο και μεγαλύτερο εύρος προβλημάτων εμφανίζοντας συχνά υπεράνθρωπες επιδόσεις. Πρόκειται για έναν κλάδο που διαρκώς εξελίσσεται, με καινοτόμες τεχνικές και αρχιτεκτονικές Νευρωνικών Δικτύων να δημοσιεύονται καθημερινά. Αυτή η διαρκής καινοτομία συνεισφέρει στην επιτυχία του κλάδου, αλλά ταυτόχρονα και σε ένα σημαντικό ελάττωμα του, την δυσκολία στην ερμηνεία των μοντέλων που παράγει. Μοντέλα που αν και πολύ αποτελεσματικά, ακόμα και οι σχεδιαστές τους δυσκολεύονται να αιτιολογήσουν τις επιδόσεις τους και τα ερμηνεύουν συχνά με διαισθητικές μεθόδους έως και τα αντιμετωπίζουν ως μαύρα κουτιά.

1.1 Το πρόβλημα της ερμηνείας

Υπάρχουν δύο βασικές προσεγγίσεις για την παραγωγή συστημάτων τεχνητής νοημοσύνης που είναι ερμηνεύσιμα από τον άνθρωπο. Η πρώτη είναι η χρήση συστημάτων που έχουν από τη φύση τους ευνόητο τρόπο λειτουργίας, όπως τα συστήματα κανόνων, ή, από τον κλάδο της μηχανικής μάθησης, τα δέντρα αποφάσεων. Η δεύτερη είναι η δημιουργία τεχνικών για την παραγωγή εξηγήσεων δυσερμήνευτων συστημάτων. Αυτή η προσέγγιση διαιρείται περαιτέρω σε τεχνικές που ασχολούνται με την εσωτερική δομή ενός συστήματος και σε τεχνικές που αντιμετωπίζουν το σύστημα ως ένα μαύρο κουτί και αγνοούν τους εσωτερικούς του μηχανισμούς. Εμείς ασχολούμαστε με αυτού του είδους τις τεχνικές, αντιμετωπίζουμε ταξινομητές ως μαύρα κουτιά και παράγουμε εξηγήσεις για τη συμπεριφορά τους με τη μορφή Συζευκτικών Ερωτημάτων.

1.2 Συνεισφορά

Στη παρούσα διπλωματική εργασία, διερευνούμε την κατασκευή Συνόλων Εξήγησης. Πρόκειται για σύνολα δεδομένων, που χρησιμοποιούνται για την εκπαίδευση ταξινομητών, τα οποία έχουμε εμπλουτίσει με υψηλότερου επιπέδου πληροφορία, η οποία παραμένει ευνόητη από τον άνθρωπο. Χρησιμοποιώντας τεχνικές από τον χώρο των Περιγραφικών Λογικών, συσχετίζουμε αυτήν την υψηλότερη επιπέδου πληροφορία με την συμπεριφορά του ταξινομητή για να παραγάγουμε εξηγήσεις στη μορφή Συζευκτικών Ερωτημάτων, τα οποία μιμούνται τη συμπεριφορά του ταξινομητή. Παρέχουμε σύνολα εξήγησης για δύο σύνολα δεδομένων, το CLEVR-Hans3 και το MNIST, καθώς και δύο αλγορίθμους για την παραγωγή εξηγήσεων.

1.3 Δομή

Στο κεφάλαιο 2 παρουσιάζουμε σύντομα τους κλάδους της Μηχανικής Μάθησης και της Εξηγήσιμης Τεχνητής Νοημοσύνης και στη συνέχεια αναπτύσσουμε τα θεωρητικά εργαλεία που θα χρειαστούμε από τον Χώρο των Περιγραφικών Λογικών και των Συζευκτικών Ερωτημάτων. Στο

κεφάλαιο 3 ορίζουμε ένα πλαίσιο χρήσης των συζευκτικών ερωτημάτων για την εξήγηση ταξινομητών και στη συνέχεια παρέχουμε δύο αλγορίθμους για την εύρεση τέτοιων ερωτημάτων. Ορίζουμε επίσης την έννοια του Συνόλου Εξήγησης και παρουσιάζουμε τα σύνολα δεδομένων CLEVR-Hans3 και MNIST καθώς και τα σύνολα εξήγησης που παραγάγαμε από αυτά. Στο κεφάλαιο 4 χρησιμοποιούμε τους αλγορίθμους του κεφαλαίου 3 για να παραγάγουμε εξηγήσεις ταξινομητών στα σύνολα δεδομένων CLEVR-Hans3 και MNIST και τις αξιολογούμε με ποσοτικά και ποιοτικά κριτήρια. Τέλος, στο κεφάλαιο 5 κάνουμε μία σύνοψη αυτής της διπλωματικής εργασίας, διατυπώνουμε τα συμπεράσματα μας και ορίζουμε μελλοντικές κατευθύνσεις.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

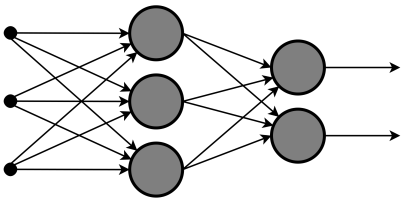
2.1 Μηχανική Μάθηση

Στον χώρο της Τεχνητής Νοημοσύνης διακρίνουμε τον κλάδο της Μηχανικής Μάθησης, ο οποίος ασχολείται με αλγορίθμους που αυτοβελτιώνονται με τη χρήση δεδομένων. Αυτοί οι αλγόριθμοι παράγουν μοντέλα τα οποία δεν έχει σχεδιάσει ο προγραμματιστής κατηγορηματικά, αλλά προκύπτουν μηχανιστικά, με μεθόδους που εντοπίζουν μοτίβα σε σύνολα δεδομένων. Θα λέγαμε ότι αυτά τα μοντέλα έχουν πείρα, παρά ευφυΐα.

Αυτοί οι αλγόριθμοι είναι ιδιαίτερα χρήσιμοι στην αυτοματοποίηση διαδικασιών των οποίων η αναλυτική περιγραφή είναι δύσκολη ή αδύνατη. Για παράδειγμα, στον κλάδο της Όρασης Υπολογιστών, συναντάμε το πρόβλημα του εντοπισμού και της ταξινόμησης αντικειμένων σε εικόνες. Είναι μάλλον ανέφικτο να δώσουμε μια σαφή περιγραφή κάθε είδους αντικειμένου που μας ενδιαφέρει, ειδικά όταν αυτή μπορεί να απαιτεί εξειδικευμένες γνώσεις, παραδείγματος χάριν από τον κλάδο της Βιολογίας για τον διαχωρισμό ποικίλων ειδών. Σε αυτό το παράδειγμα, ένας αλγόριθμος μηχανικής μάθησης θα αξιοποιούσε ένα σύνολο εικόνων, επισημειωμένες με τις προβλέψεις που θέλουμε να αυτοματοποιήσουμε. Αυτή η επισημείωση αρκεί να γίνει μία φορά από γνώστες του αντικειμένου και στη συνέχεια μπορεί να αξιοποιείται διαρκώς. Στη παρούσα διπλωματική εργασία χρησιμοποιούμε το σύνολο δεδομένων MNIST, το οποίο αποτελείται από 70.000 εικόνες, δημιουργήθηκε το 1998 και έκτοτε έχει χρησιμοποιηθεί σε αναρίθμητες εργασίες.

2.1.1 Τεχνητά Νευρωνικά Δίκτυα

Ένα από τα κυριότερα μοντέλα που παράγουν οι αλγόριθμοι της Μηχανικής Μάθησης είναι τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks - ANN). Τα ANN είναι μοντέλα εμπνευσμένα από τους εγκεφάλους των ζώων, έχουν ως δομικό στοιχείο τους τεχνητούς νευρώνες και είναι δομημένα σε μορφή δικτύου, με κόμβους τους τεχνητούς νευρώνες και συνδέσεις που μιμούνται τις συνάψεις των εγκεφάλων. Ένας τεχνητός νευρώνας δέχεται ένα σήμα με τη μορφή αριθμών από τις συνάψεις εισόδου του, το επεξεργάζεται μέσω κάποιας μη γραμμικής πράξης και το αναμεταδίδει επεξεργασμένο στους γείτονες του με τις συνάψεις εξόδου του. Οι τεχνητοί νευρώνες είναι συνήθως δομημένοι σε μία σειρά από στρώματα, με κάθε στρώμα νευρώνων να δέχεται ως είσοδο τις εξόδους του προηγουμένου στρώματος.



Σχήμα 2.1: Ένα Τεχνητό Νευρωνικό Δίκτυο δομημένο σε στρώματα.

2.1.2 Συνελικτικά Νευρωνικά Δίκτυα

Μία πολύ σημαντική κατηγορία ANN είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networs - CNN). Χρησιμοποιούνται ευρέως στην Όραση Υπολογιστών καθώς είναι εμπνευσμένα από τον οπτικό φλοιό των ζώων και χρησιμοποιούν την πράξη της συνέλιξης η οποία είναι

βαθιά καθιερωμένη στην όραση υπολογιστών για το φιλτράρισμα εικόνων. Οι νευρώνες ενός συνελικτικού στρώματος είναι οργανωμένοι σε ένα συνελικτικό φίλτρο, γνωστό και ως πυρήνα, το οποίο είναι συνήθως πολύ μικρότερο σε μέγεθος από την έξοδο του προηγούμενου στρώματος. Η έξοδος του στρώματος προκύπτει από τη συνέλιξη της εισόδου με τον πυρήνα. Για έναν μονοδιάστατο πυρήνα k η συνέλιξη ορίζεται ως

$$(f * k)[n] = \sum_{m=0}^{M} f[n-m]k[m]$$

Όπου f η είσοδος και M το μέγεθος του πυρήνα. Το στρώμα αυτό επομένως φιλτράρει το σήμα εισόδου και εκπαιδεύεται μέσω αλγορίθμων για να μάθει το κατάλληλο φίλτρο.

2.1.3 Βαθιά Μάθηση

Η Βαθιά Μάθηση είναι μία κατηγορία αλγορίθμων Μηχανικής Μάθησης που παράγουν Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks - DNN), δηλαδή, ANN με μεγάλο πλήθος στρωμάτων συνδεδεμένων εν σειρά. Θεωρούμε ότι κάθε στρώμα επιτρέπει στο ANN να παίρνει τα χαρακτηριστικά που δέχεται στην είσοδο του και να δημιουργεί νέα, υψηλότερου επιπέδου χαρακτηριστικά. Στην εικόνα 2.2 βλέπουμε ένα παράδειγμα από τον χώρο της Όρασης Υπολογιστών, όπου ένα βαθύ CNN έχει μάθει στο πρώτο στρώμα απλά χαρακτηριστικά, όπως χρώματα και γραμμές, στο δεύτερο στρώμα απλά σχήματα και υφές ενώ στο τρίτο έχει μάθει σύνθετες υφές και τμήματα αντικειμένων. Παλαιότερα χρησιμοποιούνταν πιο ρηχά δίκτυα, με είσοδο υψηλού επιπέδου χαρακτηριστικά κατασκευασμένα από τον άνθρωπο (π.χ. για εικόνες, Ιστόγραμμα Κατευθυντηρίων Παραγώγων), όμως έχει φανεί πλέον ότι τα ANN είναι πιο αποτελεσματικά από τον άνθρωπο στην κατασκευή χαρακτηριστικών που είναι χρήσιμα για τα ίδια.

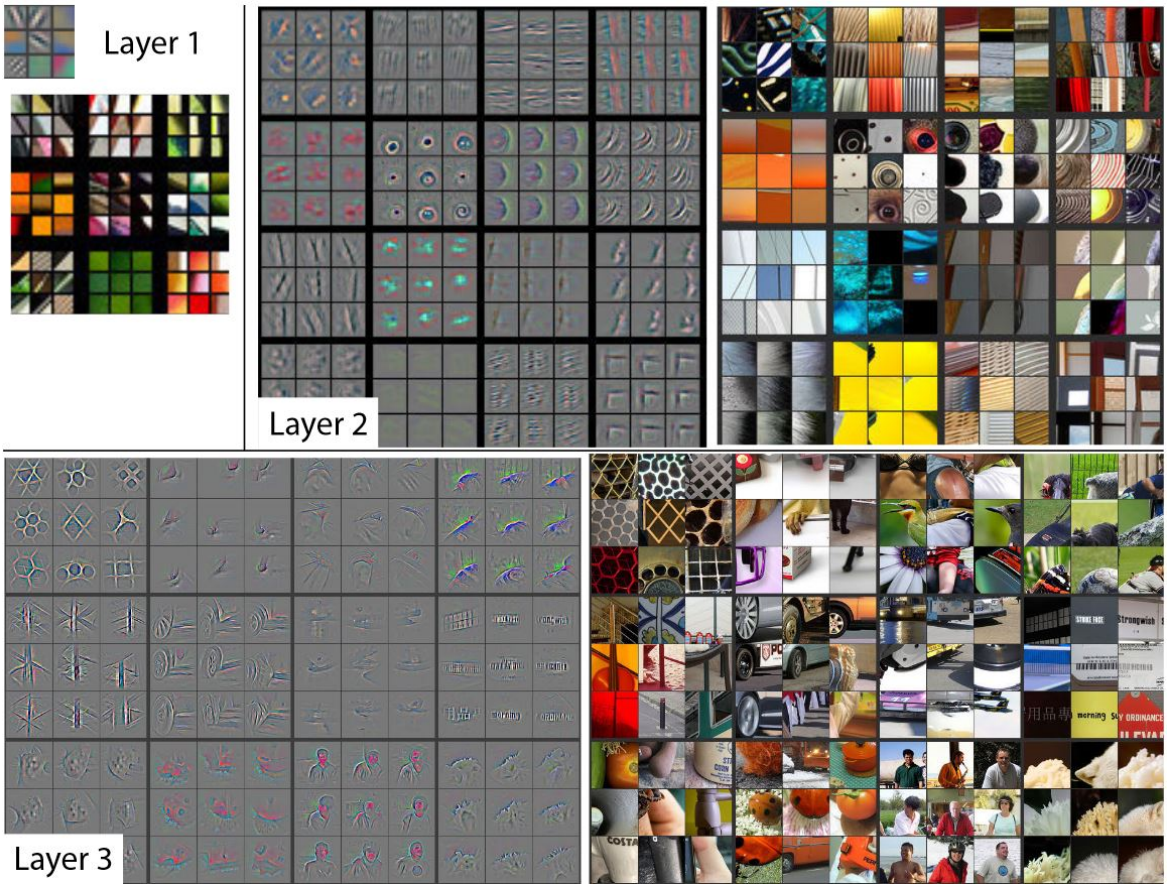
Με αυτήν την μεθοδολογία βέβαια εμφανίζεται το πρόβλημα της επεξηγηματικότητας. Αν και τα μοντέλα που αποκτάμε από αυτούς τους αλγορίθμους είναι πολύ πιο αποτελεσματικά, είναι και πιο δύσκολο να ερμηνευθούν οι αποφάσεις που παίρνουν. Δεν είναι πάντα τετριμμένη εργασία να ερμηνεύσουμε τα χαρακτηριστικά που μαθαίνει ένα DNN, ειδικά τα βαθύτερα στρώματα του.

2.2 Εξηγήσιμη Τεχνητή Νοημοσύνη

Ο κλάδος της Εξηγήσιμης Τεχνητής Νοημοσύνης ασχολείται με τον σχεδιασμό συστημάτων τεχνητής νοημοσύνης των οποίων η συμπεριφορά είναι ευνόητη για τον άνθρωπο, είτε σχεδιάζοντας συστήματα που είναι εξαρχής ερμηνεύσιμα, είτε εξάγοντας εξηγήσεις για τη λειτουργία δυσερμήνευτων συστημάτων. Ο κλάδος προσφέρει πολλές εναλλακτικές προσεγγίσεις στην εξήγηση συστημάτων. Υπάρχουν προσεγγίσεις που αγνοούν την εσωτερική λειτουργία του συστήματος (μαύρα κουτιά) και άλλες που αξιοποιούν την δομή του για την παραγωγή εξηγήσεων. Οι εξηγήσεις μπορεί να παίρνουν τη μορφή κανόνων, αντιπαραδειγμάτων ή να δίνουν πληροφορίες για την συνεισφορά που έχουν τμήματα της εισόδου ενός ταξινομητή στην απόφαση που πήρε. Επίσης οι εξηγήσεις μπορούν να αφορούν είτε την γενική συμπεριφορά του συστήματος είτε τη συμπεριφορά που έχει σε επιμέρους περιπτώσεις. Η μέθοδος με την οποία θα ασχοληθούμε εμείς αντιμετωπίζει ταξινομητές ως μαύρα κουτιά, παράγει εξηγήσεις σε μορφή που είναι κοντινή με αυτήν των κανόνων και αφορούν την συνολική συμπεριφορά του ταξινομητή.

2.3 Οντολογίες

Οι οντολογίες αποτελούν ένα εργαλείο για τη δομημένη αναπαράσταση πληροφορίας. Περιγράφουν αντικείμενα του κόσμου, τα οποία χαρακτηρίζουν με ιδιότητες και τα συνδέουν μεταξύ τους με σχέσεις, χρησιμοποιώντας για γλώσσα τις Περιγραφικές Λογικές (Description Logics). Το λεξιλόγιο μίας οντολογίας αποτελείται από τα ονόματα των αντικειμένων που περιγράφει τα οποία



Σχήμα 2.2: Οπτικοποίηση των χαρακτηριστικών που έχουν μάθει τα τρία πρώτα στρώματα ενός Βαθιού Νευρωνικού Δικτύου. (Zeiler και Fergus 2013)

καλούνται Individual Names (IN), τα ονόματα των εννοιών που αποτυπώνει τα οποία καλούνται Concept Names (CN), και τα ονόματα των ρόλων με τους οποίους συνδέει τα αντικείμενα τα οποία καλούνται Role Names (RN), (Στάμου 2015).

Για παράδειγμα, μία οντολογία που αφορά ταινίες μπορεί να χρησιμοποιεί το λεξιλόγιο

$$\mathsf{IN} = \{\mathsf{mulhollandDr}, \mathsf{davidLynch}\}, \quad \mathsf{CN} = \{\mathsf{Movie}, \mathsf{Director}\}, \quad \mathsf{RN} = \{\mathsf{hasDirector}\}$$

Με το συντακτικό που παρέχει μία περιγραφική λογική μπορούμε στη συνέχεια να διατυπώσουμε προτάσεις όπως ότι το Mulholland Dr. είναι ταινία, $\mathsf{Movie}(\mathsf{mulhollandDr})$, και ο David Lynch σκηνοθέτης, $\mathsf{Director}(\mathsf{davidLynch})$, που σκηνοθέτησε το Mulholland Dr., $\mathsf{hasDirector}(\mathsf{mulhollandDr}, \mathsf{davidLynch})$.

Τέτοιες προτάσεις ονομάζονται ισχυρισμοί και εμείς θα συναντήσουμε δύο τύπους. Ισχυρισμός έννοιας ονομάζεται μία έκφραση της μορφής $A(a)$ όπου $A \in \mathsf{CN}$, $a \in \mathsf{IN}$ και εκφράζει ότι το a έχει την ιδιότητα A. Ισχυρισμός ρόλου ονομάζεται μία έκφραση της μορφής $r(a, b)$ όπου $r \in \mathsf{RN}$, $a, b \in \mathsf{IN}$ και εκφράζει ότι το a συνδέεται με το b μέσω του ρόλου r. Ένα σύνολο από ισχυρισμούς καλείται σώμα ισχυρισμών ή Abox (assertion box) και θα το συμβολίζουμε με το $\mathcal{A}$, π.χ.:

$$\mathcal{A} = \{\mathsf{Movie}(\mathsf{mulhollandDr}),\ \mathsf{Director}(\mathsf{davidLynch}),\ \mathsf{hasDirector}(\mathsf{mulhollandDr}, \mathsf{davidLynch})\}$$

Πέρα από την αναπαράσταση του Abox με τη χρήση συνόλων, είναι ιδιαίτερα χρήσιμη και η αναπαράσταση του με την μορφή γράφου. Για κάθε στοιχείο $a \in \mathsf{IN}$ σχεδιάζουμε έναν κόμβο, με ετικέτα που αναγράφει κάθε έννοια $C \in \mathsf{CN}$ για την οποία $C(a) \in \mathcal{A}$. Ενώνουμε δυο κόμβους a, b με μία ακμή με ετικέτα r αν $r(a, b) \in \mathcal{A}$. Αυτός ο γράφος περιγράφεται από μία τριπλέτα (V, E, L) όπου

$V = \mathsf{IN}$ οι κόμβοι a του γράφου, $E \subseteq \mathsf{IN} \times \mathsf{RN} \times \mathsf{IN}$ οι ακμές (a, r, b) του γράφου, και $L : V \to 2^{\mathsf{CN}}$ η συνάρτηση ετικετών $L(a) = \{C, D, \dots\}$ των κόμβων του γράφου.

Οι έννοιες και οι ρόλοι που περιέχονται στα σύνολα CN, RN μπορούν να συνδυαστούν για να εκφράσουν σύνθετες έννοιες. Μπορεί να θέλουμε να αναφερθούμε για παράδειγμα στις γυναίκες σκηνοθέτιδες, Woman $\sqcap$ Director, ή στις ταινίες που έχουν γυναίκα ως σκηνοθέτιδα, $\exists$hasDirector. Woman.

Αυτές ονομάζονται σύνθετες έννοιες και κατασκευάζονται αναδρομικά από άλλες έννοιες και ρόλους με τη χρήση τελεστών. Θα συναντήσουμε δύο τελεστές. Έστω C, D δύο έννοιες, σύνθετες ή απλές και r ένας ρόλος. Μπορούμε να κατασκευάσουμε την έννοια $C \sqcap D$ η οποία χαρακτηρίζει τα αντικείμενα που έχουν και την ιδιότητα C και την D. Μπορούμε να κατασκευάσουμε και την έννοια $\exists r.C$ ή οποία χαρακτηρίζει τα αντικείμενα που μέσω του ρόλου r συνδέονται με αντικείμενα που έχουν την ιδιότητα C. Στην προηγούμενη έκφραση μπορούμε να χρησιμοποιήσουμε και το σύμβολο $\top$(Top), στη θέση του C, $\exists r.\top$, για να αναφερθούμε στα αντικείμενα που μέσω του ρόλου r συνδέονται με αντικείμενα που δεν έχουν αναγκαστικά κάποια συγκεκριμένη ιδιότητα. Θεωρούμε ότι το $\top$ είναι μία ειδική έννοια που χαρακτηρίζει όλα τα αντικείμενα. Το αντίστροφο του Top εκφράζει το σύμβολο $\bot$(Bottom), είναι μία ειδική έννοια στην οποία δεν ανήκει κανένα αντικείμενο και χρησιμοποιείται συνήθως για να εκφράσει ότι κάτι είναι αδύνατον.

Πέρα από τους ισχυρισμούς υπάρχουν και άλλες εκφράσεις οι οποίες ονομάζονται αξιώματα και μας επιτρέπουν να εκφράζουμε κανόνες του κόσμου, όπως π.χ. ότι οι σκηνοθέτες είναι και άνθρωποι, Director $\sqsubseteq$ Human). Εμείς θα συναντήσουμε δύο τύπους αξιωμάτων, τα αξιώματα υπαγωγής εννοιών, που έχουν τη μορφή $C \sqsubseteq D$, όπου C και D είναι έννοιες, σύνθετες ή απλές και τα αξιώματα υπαγωγής ρόλων, που έχουν την μορφή $r \sqsubseteq s$ όπου r και s είναι ονόματα ρόλων. Τα αξιώματα υπαγωγής εννοιών εκφράζουν ότι τα αντικείμενα που χαρακτηρίζονται από την έννοια στην αριστερή πλευρά, χαρακτηρίζονται και από την έννοια στην δεξιά πλευρά, ενώ τα αντικείμενα υπαγωγής ρόλων εκφράζουν ότι τα αντικείμενα που συνδέονται με τον ρόλο της αριστερής πλευράς συνδέονται και με τον ρόλο της δεξιάς πλευράς. Ένα σύνολο από αξιώματα καλείται σώμα ορολογίας ή Tbox (terminological box) και θα το συμβολίζουμε με το $\mathcal{T}$, π.χ.:

$$\mathcal{T} = \{\mathsf{Director} \sqsubseteq \mathsf{Human},\ \mathsf{Movie} \sqsubseteq \exists \mathsf{hasDirector}.\top,\ \mathsf{hasDirector} \sqsubseteq \mathsf{hasCrewMember}\}$$

Ένα ζεύγος Abox, Tbox καλείται βάση γνώσης (knowledge base), $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

Τα αξιώματα που εμφανίζονται σε ένα Tbox περιορίζονται στην μορφή τους από την εκφραστικότητα της περιγραφικής λογικής που χρησιμοποιείται. Εμάς μας ενδιαφέρει κυρίως η περιγραφική λογική RL. Τα αξιώματα της RL έχουν την μορφή

$$C \sqsubseteq D, \quad r \sqsubseteq s, \quad C \sqsubseteq \bot$$

όπου r, s ονόματα ρόλων, D όνομα έννοιας και C σύνθετη έννοια η οποία έχει μορφή που ορίζεται αναδρομικά από την εξής γραμματική:

$$C ::= D \quad | \quad \exists r.\top \quad | \quad \exists r.C \quad | \quad C_1 \sqcap C_2$$

Οι έννοιες των συμβόλων που έχουμε εξηγήσει έως τώρα διαισθητικά, αποτυπώνονται φορμαλιστικά μέσω των ερμηνειών. Ερμηνεία μίας βάσης γνώσης $\mathcal{K}$ είναι ένα ζεύγος $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, όπου $\Delta^{\mathcal{I}}$ ένα μη κενό (πιθανώς άπειρο) σύνολο αντικειμένων που ονομάζεται πεδίο και $\cdot^{\mathcal{I}}$ μία απεικόνιση που ονομάζεται αντιστοίχηση ερμηνείας και ερμηνεύει τα μη λογικά σύμβολα της $\mathcal{K}$ με δομές στοιχείων του $\Delta^{\mathcal{I}}$ ως εξής:

- Τα άτομα ερμηνεύονται ως αντικείμενα του $\Delta^{\mathcal{I}}$, δηλαδή αν $a \in \mathsf{IN}$ ένα άτομο της γνώσης $\mathcal{K}$, τότε $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

- Οι απλές έννοιες ερμηνεύονται ως υποσύνολα του $\Delta^{\mathcal{I}}$, δηλαδή αν $A \in \mathsf{CN}$ μία απλή έννοια της γνώσης $\mathcal{K}$, τότε $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$.

- Οι ρόλοι ερμηνεύονται ως υποσύνολα του καρτεσιανού γινομένου $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, δηλαδή αν $r \in \mathsf{RN}$ ένας ατομικός ρόλος της γνώσης $\mathcal{K}$, τότε $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Οι διαισθητικές ερμηνείες που δώσαμε αποτυπώνονται στους παρακάτω κανόνες:

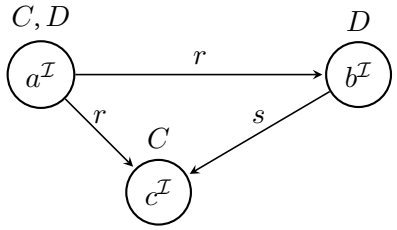
- Για κάθε $x \in \Delta^{\mathcal{I}}$ ισχύει $x \in \top^{\mathcal{I}}$.

- Δεν υπάρχει $x \in \Delta^{\mathcal{I}}$ τέτοιο ώστε $x \in \bot^{\mathcal{I}}$

- $x \in (C \sqcap D)^{\mathcal{I}}$ αν και μόνο αν $x \in C^{\mathcal{I}}$ και $x \in D^{\mathcal{I}}$.

- $x \in (\exists r.C)^{\mathcal{I}}$ αν και μόνο αν υπάρχει $y \in \Delta^{\mathcal{I}}$, τέτοιο ώστε $(x, y) \in r^{\mathcal{I}}$ και $y \in C^{\mathcal{I}}$.

Οι ερμηνείες μπορούν να απαρασταθούν και αυτές σε μορφή γράφου, όπως και τα Abox. Για κάθε στοιχείο $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ σχεδιάζουμε έναν κόμβο, με ετικέτα που αναγράφει κάθε έννοια $C \in \mathsf{CN}$ για την οποία $a^{\mathcal{I}} \in C^{\mathcal{I}}$. Ενώνουμε δυο κόμβους $a^{\mathcal{I}}, b^{\mathcal{I}}$ με μία ακμή με ετικέτα r αν $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. Η τριπλέτα (V, E, L) του γράφου ορίζεται αντίστοιχα με την περίπτωση του Abox.



Λέμε ότι μία ερμηνεία $\mathcal{I}$ της γνώσης $\mathcal{K}$ ικανοποιεί έναν ισχυρισμό του Abox $\mathcal{A}$ της μορφής $C(a)$ αν και μόνο αν $a^{\mathcal{I}} \in C^{\mathcal{I}}$ και έναν ισχυρισμό της μορφής $r(a, b)$ αν και μόνο αν $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. Μία ερμηνεία ικανοποιεί το Abox $\mathcal{A}$ αν και μόνο αν ικανοποιεί όλους του ισχυρισμούς του. Στην περίπτωση αυτή λέμε ότι η $\mathcal{I}$ είναι μοντέλο του $\mathcal{A}$.

Λέμε ότι μία ερμηνεία $\mathcal{I}$ της γνώσης $\mathcal{K}$ ικανοποιεί ένα αξίωμα του Tbox $\mathcal{T}$ της μορφής $C \sqsubseteq D$ αν και μόνο αν $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ και ένα αξίωμα της μορφής $r \sqsubseteq s$ αν και μόνο αν $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. Μία ερμηνεία ικανοποιεί το Tbox $\mathcal{T}$ αν και μόνο αν ικανοποιεί όλα τα αξιώματα του. Στην περίπτωση αυτή λέμε ότι η $\mathcal{I}$ είναι μοντέλο του $\mathcal{T}$.

Λέμε ότι μία ερμηνεία $\mathcal{I}$ ικανοποιεί τη γνώση $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ αν και μόνο αν ικανοποιεί τα $\mathcal{A}$, $\mathcal{T}$. Στην περίπτωση αυτή λέμε ότι η $\mathcal{I}$ είναι μοντέλο της .

Σε πολλές περιγραφικές λογικές, όπως και στην RL, ορίζεται το λεγόμενο κανονικό μοντέλο (canonical model) το οποίο στηρίζεται στην απλούστερη ερμηνεία της βάσης γνώσης. Ως πρότυπο μοντέλο (standard model) ενός Abox ορίζεται η ερμηνεία $\mathcal{I}_{\mathcal{A}}$ με

$$
\begin{aligned}
&\Delta^{\mathcal{I}_{\mathcal{A}}} = \mathsf{IN}, \\
&a^{\mathcal{I}_{\mathcal{A}}} = a, && \forall a \in \mathsf{IN}, \\
&A^{\mathcal{I}_{\mathcal{A}}} = \{a \mid A(a) \in \mathcal{A}\}, && \forall A \in \mathsf{CN}, \\
&r^{\mathcal{I}_{\mathcal{A}}} = \{(a, b) \mid r(a, b) \in \mathcal{A}\}, && \forall r \in \mathsf{RN}.
\end{aligned}
$$

Αυτή η ερμηνεία ικανοποιεί εξ' ορισμού όλους τους ισχυρισμούς του Abox και επομένως είναι μοντέλο του. Επεκτείνουμε το πρότυπο μοντέλο στο κανονικό με βάση κάποιους κανόνες ώστε να ικανοποιεί και το Tbox. Κατασκευάζουμε μία σειρά από ερμηνείες $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_n$, θέτοντας $\mathcal{I}_0 = \mathcal{I}_{\mathcal{A}}$ και εφαρμόζοντας τους εξής κανόνες σε κάθε $\mathcal{I}_k$ για να πάρουμε το $\mathcal{I}_{k+1}$:

(c) Αν $a \in B^{\mathcal{I}_k},\ B \sqsubseteq A \in \mathcal{T}$ όμως $a \notin A^{\mathcal{I}_k}$, τότε προσθέτουμε το a στο $A^{\mathcal{I}_{k+1}}$.
(r) Αν $(a,b) \in r^{\mathcal{I}_k},\ r \sqsubseteq s \in \mathcal{T}$ όμως $(a,b) \notin s^{\mathcal{I}_k}$, τότε προσθέτουμε το (a,b) στο $s^{\mathcal{I}_{k+1}}$.
(b) Αν $a \in B^{\mathcal{I}_k}$ και $B \sqsubseteq \bot \in \mathcal{T}$ τότε η διαδικασία τερματίζει.

Επειδή τα πεδία των $\mathcal{I}_k$ είναι πεπερασμένα και όλα συμπίπτουν με τους individual του $\mathcal{A}$, η διαδικασία τερματίζει μετά από ένα πεπερασμένο αριθμό βημάτων είτε επειδή δεν μπορούν πλέον να εφαρμοστούν οι κανόνες **(c)**, **(r)** είτε επειδή εφαρμόζεται ο **(b)**. Στην πρώτη περίπτωση η ερμηνεία που προκύπτει ικανοποιεί όλα τα αξιώματα του $\mathcal{T}$ και ονομάζεται κανονικό μοντέλο της $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ και συμβολίζεται $\mathcal{C}_{\mathcal{T},\mathcal{A}}$. Στην δεύτερη περίπτωση καμία ερμηνεία δεν είναι μοντέλο της $\mathcal{K}$, (Kontchakov και Zakharyaschev 2014).

Το κανονικό μοντέλο μίας RL βάσης γνώσης είναι ιδιαίτερα χρήσιμο γιατί είναι οικουμενικό μοντέλο (universal model), δηλαδή περιέχεται κάθε άλλο μοντέλο και συγκεκριμένα είναι ομομορφικό σε αυτά, (Baader κ.ά. 2017). Έστω $\mathcal{I}_1, \mathcal{I}_2$ ερμηνείες, μία συνάρτηση $h : \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ καλείται ομομορφισμός αν ισχύουν τα εξής:

$$\textbf{(i)}\ d \in A^{\mathcal{I}_1} \Rightarrow h(d) \in A^{\mathcal{I}_2},\ \forall A \in \mathsf{CN}$$
$$\textbf{(ii)}\ (d,e) \in r^{\mathcal{I}_1} \Rightarrow (h(d), h(e)) \in r^{\mathcal{I}_2},\ \forall r \in \mathsf{RN}$$
$$\textbf{(iii)}\ h(a^{\mathcal{I}_1}) = a^{\mathcal{I}_2},\ \forall a \in \mathsf{IN}$$

Αν υπάρχει ομομορφισμός από την $\mathcal{I}_1$ στην $\mathcal{I}_2$ γράφουμε $\mathcal{I}_1 \to \mathcal{I}_2$.

Για την απόδειξη ότι $\mathcal{C}_{\mathcal{T},\mathcal{A}} \to \mathcal{I}$ για κάθε μοντέλο $\mathcal{I}$ αρκεί να δείξουμε ότι οι ερμηνείες $\mathcal{I}_k$ που χρησιμοποιούνται για την κατασκευή του $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ είναι ομομορφικές προς την $\mathcal{I}$ με ομομορφισμό την $h(a) = a^{\mathcal{I}}$ και επομένως επαγωγικά και το $\mathcal{C}_{\mathcal{T},\mathcal{A}}$. Για την αρχή της επαγωγής αρκεί να παρατηρήσουμε ότι επειδή η $\mathcal{I}$ είναι μοντέλο του Abox, πρέπει να ισχύει ότι $\mathcal{I}_{\mathcal{A}} = \mathcal{I}_0 \to \mathcal{I}$. Για την επαγωγική υπόθεση, ισχύει ότι $\mathcal{I}_k \to \mathcal{I}$. Για το επαγωγικό βήμα, η $\mathcal{I}_{k+1}$ προέκυψε από την εφαρμογή είτε του κανόνα **(c)** είτε του κανόνα **(r)**, εφόσον η διαδικασία τερμάτισε σε μοντέλο.

- Αν εφαρμόστηκε ο κανόνας **(c)** τότε υπάρχει $a \in B^{\mathcal{I}_k}$, και $B \sqsubseteq A \in \mathcal{T}$ τέτοιο ώστε η $\mathcal{I}_{k+1}$ προέκυψε από την $\mathcal{I}_k$ προσθέτοντας το a στην $A^{\mathcal{I}_{k+1}}$. Από την επαγωγική υπόθεση και την ιδιότητα **(i)** των ομομορφισμών έχουμε ότι $h(a) \in B^{\mathcal{I}}$. Αφού η $\mathcal{I}$ είναι μοντέλο του $\mathcal{T}$ τότε πρέπει και $h(a) \in A^{\mathcal{I}}$ επομένως και $\mathcal{I}_{k+1} \to \mathcal{I}$.

- Αν εφαρμόστηκε ο κανόνας **(r)** τότε υπάρχει $(a,b) \in r^{\mathcal{I}_k}$, και $r \sqsubseteq s \in \mathcal{T}$ τέτοιο ώστε η $\mathcal{I}_{k+1}$ προέκυψε από την $\mathcal{I}_k$ προσθέτοντας το (a,b) στο $s^{\mathcal{I}_{k+1}}$. Από την επαγωγική υπόθεση και την ιδιότητα **(ii)** των ομομορφισμών έχουμε ότι $(h(a), h(b)) \in r^{\mathcal{I}}$. Αφού η $\mathcal{I}$ είναι μοντέλο του $\mathcal{T}$ τότε πρέπει και $(h(a), h(b)) \in s^{\mathcal{I}}$ επομένως και $\mathcal{I}_{k+1} \to \mathcal{I}$.

2.3.1 Ερωτήματα

Το βασικό εργαλείο αυτής της διπλωματικής εργασίας είναι τα ερωτήματα (queries) και συγκεκριμένα τα συζευκτικά ερωτήματα (conjunctive queries - CQ). Τα ερωτήματα μας επιτρέπουν να εξάγουμε πληροφορίες που μας ενδιαφέρουν από μία βάση γνώσης. Στο πλαίσιο των προηγουμένων παραδειγμάτων, από μία βάση γνώσης που αφορά ταινίες, θα μπορούσαμε να εντοπίσουμε μέσω ενός ερωτήματος τις ταινίες των οποίων ο σκηνοθέτης έχει και τον ρόλο του ηθοποιού.

Συζευκτικό ερώτημα καλείται μία έκφραση της μορφής $q = \{\langle x_1, \ldots, x_k \rangle | \exists y_1 \ldots \exists y_l (c_1 \wedge \cdots \wedge c_n)\}$ όπου $l \geq 0, n \geq 1$. Τα y_i καλούνται ποσοτικοποιημένες μεταβλητές και τα x_i μεταβλητές απάντησης. Τα c_i είναι της μορφής $C(u)$ ή $r(u,v)$ όπου $C \in \mathsf{CN},\ r \in \mathsf{RN},\ u, v \in \mathsf{IN} \cup \{x_i\}_{i=1}^{k} \cup \{y_i\}_{i=1}^{l}$. Το σύνολο των μεταβλητών που εμφανίζονται στο q συμβολίζεται $\mathsf{VN}(q)$. Το σύνολο των c_i καλείται σώμα του q και συμβολίζεται $\mathsf{body}(q)$. Εμάς μας ενδιαφέρουν τα CQ με μία μεταβλητή απάντησης και χωρίς ονόματα individual στο σώμα του q. Θέλουμε επιπλέον τα queries να μην περιέχουν μεταβλητές που δεν συνδέονται με το x μέσω κάποιου ρόλου γιατί εκείνες δεν εκφράζουν κάποιον περιορισμό για το x αλλά για την βάση γνώσης γενικά. Θέλουμε επομένως τα queries να

είναι συνεκτικά. Για απλότητα μπορούμε να τα γράφουμε $q = \{c_1, \ldots, c_n\}$ θεωρώντας πάντα το x ως τη μεταβλητή απάντησης. Ένα συζευκτικό ερώτημα εκφράζει ουσιαστικά ένα σύνολο απαιτήσεων που θέλουμε να πληροί το x.

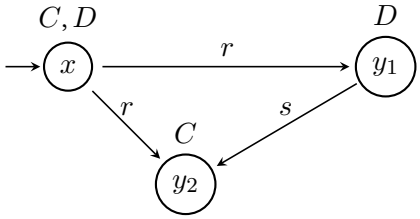
Έστω q ένα CQ και $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ μία ερμηνεία της βάσης γνώσης $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. Ταίριασμα του q στην $\mathcal{I}$ καλείται μία απεικόνιση $\pi : \mathsf{VN}(q) \to \Delta^{\mathcal{I}}$ τέτοια ώστε

(i) $\pi(u) \in C^{\mathcal{I}}$ για κάθε $C(u) \in q$

(ii) $(\pi(u), \pi(v)) \in r^{\mathcal{I}}$ για κάθε $r(u, v) \in q$

Το ταίριασμα αποτελεί δηλαδή μία επιτυχή αντικατάσταση μεταβλητών στο q, τέτοια ώστε όλες οι απαιτήσεις του q να τηρούνται υπό την $\mathcal{I}$. Τα $a \in \mathsf{IN}$ για τα οποία υπάρχει ένα ταίριασμα με $\pi(x) = a^{\mathcal{I}}$ καλούνται απαντήσεις του q στην $\mathcal{K}$ υπό την $\mathcal{I}$. Τα $a \in \mathsf{IN}$ για τα οποία υπάρχει κάποιο ταίριασμα για κάθε μοντέλο της $\mathcal{K}$ καλούνται βέβαιες απαντήσεις (certain answers) και συμβολίζονται $\mathsf{cert}(q, \mathcal{K})$. Η χρησιμότητα του κανονικού μοντέλου στην RL είναι ότι αν ένας invividual αποτελεί απάντηση του q υπό το $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ τότε αποτελεί και βέβαιη απάντηση. Υπάρχουν και άλλες εκφραστικότητες για τις οποίες ισχύει αυτό, όπως η EL και η QL, όμως σε αυτές τα κανονικά μοντέλα μπορεί να είναι άπειρα και η χρήση τους στην εύρεση certain answers είναι σύνθετη και εκτός των πλαισίων αυτής της διπλωματικής.

Μπορούμε να αναπαραστήσουμε και τα queries ως γράφους. Για κάθε όρο $u \in \mathsf{VN}(q)$ σχεδιάζουμε έναν κόμβο, με ετικέτα που αναγράφει κάθε έννοια $C \in \mathsf{CN}$ για την οποία $C(u) \in \mathsf{body}(q)$. Ενώνουμε δυο κόμβους u, v με μία ακμή με ετικέτα r αν $r(u, v) \in \mathsf{body}(q)$. Η τριπλέτα (V, E, L) του γράφου ορίζεται αντίστοιχα με την περίπτωση του Abox. Η αναπαράσταση αυτή είναι ιδιαίτερα χρήσιμη γιατί συνδέεται με τις απαντήσεις του query.



Επισημαίνουμε τον κόμβο x με ένα βέλος γιατί η μεταβλητή απάντησης έχει πάντα ξεχωριστή σημασία.

Ορίζουμε τον ομομορφισμό μεταξύ γράφων με τρόπο παρόμοιο με αυτόν των ερμηνειών. Έστω δύο γράφοι

$G_1 = (V_1, E_1, L_1),\ G_2 = (V_2, E_2, L_2)$, μία συνάρτηση

$h : V_1 \to V_2$ καλείται ομομορφισμός αν σέβεται τη δομή του γράφου G_1, δηλαδή:

(i) $\forall v \in V_1,\ L_1(v) \subseteq L_2(h(v))$

(ii) $\forall u, v \in V_1,\ (u, r, v) \in E_1 \Rightarrow (h(u), r, h(v)) \in E_2$

Αν δύο γράφοι είναι ομομορφικοί γράφουμε $G_1 \to G_2$. Για τους ομομορφισμούς μεταξύ των γράφων query έχουμε την επιπλέον απαίτηση $h(x) = x$.

Από τον ορισμό του ταιριάσματος είναι εμφανές ότι αποτελεί ομομορφισμό από το γράφο του query στον γράφο της ερμηνείας. Επιπλέον, οι ομομορφισμοί μεταξύ ερμηνειών είναι εμφανές ότι είναι και ομομορφισμόι μεταξύ γράφων ικανοποιώντας όμως επιπλέον και την ιδιότητα **(iii)**. Επομένως, αν υπάρχει ομομορφισμός h από την ερμηνεία $\mathcal{I}_1$ στην ερμηνεία $\mathcal{I}_2$ τότε από τη σύνθεση ομομορφισμών κάθε ταίριασμα π_1 του q στην $\mathcal{I}_1$ με $\pi_1(x) = a^{\mathcal{I}_1}$ μας δίνει ένα ταίριασμα pi_2 του q στην $\mathcal{I}_2$ με $\pi_2(x) = h(a^{\mathcal{I}_1}) = a^{\mathcal{I}_2}$. Από αυτό προκύπτει η ιδιότητα του κανονικού μοντέλου, αν υπάρχει ταίριασμα από το query στο κανονικό μοντέλο, από την σύνθεση ομομορφισμών υπάρχει ταίριασμα και σε κάθε άλλο μοντέλο και επομένως οι απαντήσεις του q υπό το κανονικό μοντέλο είναι και βέβαιες απαντήσεις.

2.3.2 Υπαγωγή

Οι ομομορφισμοί μας επιτρέπουν επίσης να συγκρίνουμε queries. Αν υπάρχει ομομορφισμός από ένα query q_1 σε ένα query q_2 λέμε ότι το query q_2 υπάγεται στο (is subsumed by) query q_1 και γράφουμε $q_2 \leq_S q_1$. Αν δυο queries υπάγονται από κοινού τότε λέμε ότι είναι συντακτικά ισοδύναμα και γράφουμε $q_2 \equiv_S q_1$. Λόγω της σύνθεσης ομομορφισμών προκύπτει ότι αν $q_2 \leq_S q_1$ τότε και $\mathsf{cert}(q_2, \mathcal{K}) \subseteq \mathsf{cert}(q_1, \mathcal{K})$ καθώς κάθε ταίριασμα π_2 του q_2 στο κανονικό μοντέλο, με $\pi_2(x) = a$, μας δίνει ένα ταίριασμα π_1 του q_1, με $\pi_1(x) = \pi_2(h(x)) = \pi_2(x) = a$, όμως δεν ισχύει αναγκαστικά το αντίστροφο. Το πρόβλημα εντοπισμού ενός ομομορφισμού μεταξύ δύο γράφων είναι στη γενική περίπτωση NP-complete (Bodirsky 2021) και επομένως το ίδιο ισχύει και για τον έλεγχο subsumption μεταξύ δυο queries.

Αν από ένα query q αφαιρέσουμε κάποια στοιχεία από το σώμα του, τότε το νέο query q' προφανώς θα υπάγεται στο q καθώς η απεικόνιση από τις μεταβλητές του q' στις αντίστοιχες μεταβλητές του q πληροί τις προϋποθέσεις του ομομορφισμού. Αν όμως και το q υπάγεται στο q' τότε είναι συντακτικά ισοδύναμα και οι όροι που αφαιρέσαμε από το query ήτανε περιττοί. Το query q' με τους ελάχιστους δυνατούς όρους αποκαλείται συμπύκνωση (condensation) του q, ενώ αν η συμπύκνωση του q ταυτίζεται με το q τότε λέμε ότι είναι συμπυκνωμένο (condensed). Η συμπύκνωση ενός query απαιτεί ελέγχους subsumption γραμμικούς στο πλήθος ως προς το μέγεθος του $\mathsf{body}(q)$ και επομένως είναι co NP-complete (Gottlob και Fermüller 1993).

Ο ελάχιστος κοινός υπαγωγός (least common subsumer - lcs) δύο queries q_1, q_2 συμβολίζεται $\mathsf{lcs}(q_1, q_2)$ και ορίζεται ως το query q για το οποίο ισχύει

$$q_1, q_2 \leq_S q, \quad \forall q' : q_1,\, q_2 \leq_S q' \Leftrightarrow q \leq_S q'$$

Είναι δηλαδή το ελάχιστο query που υπάγει τα q_1, q_2. Από τη σκοπιά των γράφων, το lcs είναι το query για τον γράφου του οποίου ισχύει $G \to G_1, G_2$ και για κάθε άλλο γράφο G' ισχύει $G' \to G_1, G_2 \Leftrightarrow G' \to G$. Ο υπολογισμός αυτού του γράφου και επομένως και του query είναι γνωστός και προκύπτει από το γινόμενο Kronecker των q_1, q_2 (Kronecker product, γνωστό και ως tensor product ή direct product). Το γινόμενο Kronecker δύο γράφων $G_1 = (V_1, E_1, L_1)$, $G_2 = (V_2, E_2, L_2)$ συμβολίζεται $G_1 \times G_2$ και ορίζεται ως εξής:

$$\begin{aligned}
&G_1 \times G_2 = (V, E, L)\\
&V = V_1 \times V_2 \quad \text{(καρτεσιανό γινόμενο συνόλων)}\\
&(u_1, r, v_1) \in E_1, \quad (u_2, r, v_2) \in E_2 \iff ((u_1, u_2), r, (v_1, v_2)) \in E\\
&L((v_1, v_2)) = L_1(v_1) \cap L_2(v_2)
\end{aligned}$$

Η απόδειξη ότι το γινόμενο Kronecker μας δίνει τον επιθυμητό γράφο είναι η εξής:
$(\Rightarrow)$ Έστω γράφος H ομομορφικός στους G_1, G_2 με ομομορφισμούς $h_1 : H \to G_1,\ h_2 : H \to G_2$. Τότε η απεικόνιση $h : H \to G_1 \times G_2$ με $h(v) = (h_1(v), h_2(v))$ είναι ομομορφισμός:

$$\begin{aligned}
&L_H(v) \subseteq L_1(h_1(v)), L_2(h_2(v)) \Leftrightarrow\\
&L_H(v) \subseteq L_1(h_1(v)) \cap L_2(h_2(v)) = L(h_1(v), h_2(v)) = L(h(v))
\end{aligned}$$

$$\begin{aligned}
&\forall u, v \in V_H, (u, r, v) \in E_H \Rightarrow\\
&(h_1(u), r, h_1(v)) \in E_1, \quad (h_2(u), r, h_2(v)) \in E_2 \Leftrightarrow\\
&((h_1(u), h_2(u)), r, (h_1(v), h_2(v))) \in E \Leftrightarrow\\
&(h(u), r, h(v)) \in E
\end{aligned}$$

$(\Leftarrow)$ Έστω γράφος H ομομορφικός στον $G_1 \times G_2$ με ομομορφισμό $h : H \to G_1 \times G_2$. Τότε μπορούμε να "προβάλουμε" τον h στους G_1, G_2 συνθέτοντας τον με τους ομομορφισμούς

$$\begin{aligned}
&\pi_1 : G_1 \times G_2 \to G_1, \quad \pi_1((v_1, v_2)) = v_1\\
&\pi_2 : G_1 \times G_2 \to G_2, \quad \pi_2((v_1, v_2)) = v_2
\end{aligned}$$

Κάποιες επισημάνσεις που πρέπει να γίνουν:

- Το γινόμενο Kronecker δυο queries έχει ως μεταβλητή απάντησης την (x_1, x_2), δηλαδή τη μεταβλητή που προκύπτει από το ζεύγος των μεταβλητών απάντησης των q_1, q_2. Αυτό συμβαίνει διότι αυτή η μεταβλητή είναι που απεικονίζεται στις μεταβλητές x_1, x_2 μέσω των προβολών π_1, π_2 και επομένως και στα στοιχεία του IN που αποτελούν certain answers.

- Πολλές φορές ο γράφος που προκύπτει από το γινόμενο Kronecker περιέχει περισσότερες από μία συνεκτικές συνιστώσες και επομένως το αντίστοιχο query έχει κάποιες μεταβλητές που δεν συνδέονται με τη συνιστώσα που περιέχει τη μεταβλητή απάντησης, έστω κύρια, μέσω κάποιου ρόλου. Υποθέτοντας ότι ένα από τα q_1, q_2, έστω το q_1, είχε κάποιο certain answer και επομένως κάποιον ομομορφισμό h_1 προς το Abox, τότε η σύνθεση π_1, h_1 μας δίνει έναν σίγουρο ομομορφισμό για τις μη κύριες συνιστώσες του lcs. Τα certain answers μπορούν να βρεθούν ανεξάρτητα εξετάζοντας ομομορφισμούς μόνο για τη κύρια συνιστώσα. Εμείς θα ασχοληθούμε μόνο με queries που γνωρίζουμε εξ' αρχής ότι έχουν κάποια απάντηση και επομένως μας ενδιαφέρει μόνο η κύρια συνιστώσα.

- Ισχύει ότι $G_1 \times G_2 \cong G_2 \times G_1$ και $(G_1 \times G_2) \times G_3 \cong G_1 \times (G_2 \times G_3)$ είναι δηλαδή ισομορφικοί με ισομορφισμό τις αμφιμονοσήμαντες απεικονίσεις $(v_1, v_2) \leftrightarrow (v_2, v_1)$ και $((v_1, v_2), v_3) \leftrightarrow (v_1, (v_2, v_3))$. Από τον ορισμό του γινομένου Kronecker είναι εμφανές ότι αυτές οι απεικονίσεις είναι ισομορφισμοί και επομένως αρκεί να γράφουμε $\prod_{i=1}^{n} G_i$ και μπορούμε να γενικεύσουμε τον ορισμό του lcs σε σύνολα από queries παίρνοντας το γινόμενο των γράφων όλων των queries.

Με βάση το κανονικό μοντέλο και την έννοια της υπαγωγής μπορούμε να ορίσουμε και την έννοια του πιο συγκεκριμένου query (most specific query - msq) ενός individual. Θέλουμε να συμπεριλάβουμε σε ένα query όσο το δυνατόν περισσότερη πληροφορία μας δίνει μία βάση γνώσης ενός individual a. Παίρνουμε αυτό το query μετατρέποντας τη συνεκτική συνιστώσα του γράφου του κανονικού μοντέλου που περιέχει τον a σε γράφο query. Μετονομάζουμε τον κόμβο του a σε x και τους υπόλοιπους σε y_i. Αυτό το query έχει από την κατασκευή του τον a ως certain answer αλλά είναι και το ελάχιστο query που τον έχει ως certain answer. Κάθε άλλο query q αν έχει ως απάντηση τον a είναι ομομορφικό σε εκείνη τη συνεκτική συνιστώσα του κανονικού μοντέλου και επομένως και στο msq αφού έχουν ακριβώς την ίδια δομή.

Κεφάλαιο 3

Σχεδιασμός και υλοποίηση

3.1 Ερωτήματα ως εξηγήσεις

Μπορούμε να χρησιμοποιήσουμε τα συζευκτικά ερωτήματα (CQ) που ορίσαμε στην ενότητα 2.3.1 για να παραγάγουμε εξηγήσεις ταξινομητών που είναι μαύρα κουτιά. Έστω σύνολο δεδομένων $\mathcal{D}$ και F η συνάρτηση ενός ταξινομητή, δηλαδή ενός μοντέλου που κατατάσσει τις εισόδους του σε κλάσεις. Η συνάρτηση F μπορούμε να θεωρήσουμε ότι παίρνει ως τιμή τον αριθμό της κλάσης στην οποία ταξινομεί την είσοδο ή αν έχουμε δυικό πρόβλημα ταξινόμησης, με έξοδο "ναι"/"όχι", παίρνει τη τιμή 1 για όσες εισόδους ταξινομεί θετικά και 0 για όσες ταξινομεί αρνητικά. Έστω και μία γνώση $\mathcal{K}$ η οποία περιέχει μία περιγραφή του συνόλου $\mathcal{D}$ με τα στοιχεία του $\mathcal{D}$ να αποτελούν individuals της γνώσης, $\mathcal{D} \subseteq \mathsf{IN}$. Τότε μπορούμε να ερμηνεύσουμε queries στη γνώση $\mathcal{K}$ ως εξηγήσεις υποσυνόλων του $\mathcal{D}$. Αν δηλαδή έχουμε $\mathsf{cert}(q, \mathcal{K}) = \mathcal{C} \subseteq \mathcal{D}$ τότε μπορούμε να θεωρήσουμε ότι το q είναι μία περιγραφή του συνόλου $\mathcal{C}$ στην γλώσσα των CQ, η οποία μπορεί να μεταφραστεί σε φυσική γλώσσα και εξηγεί τι κοινά σημεία έχουν τα στοιχεία του συνόλου $\mathcal{C}$. Εμείς θα θέλαμε να βρούμε εξηγήσεις που περιγράφουν τέλεια τις εξόδους του ταξινομητή, δηλαδή να βρούμε ένα query για κάθε κλάση i του ταξινομητή τέτοιο ώστε $\mathsf{cert}(q_i, \mathcal{K}) = \mathcal{C}_i = \{a \in \mathcal{D} : \mathsf{F}(a) = i\}$. Αυτά τα queries θα μας έδιναν μία τέλεια περιγραφή του ταξινομητή.

Επειδή συνήθως είναι ανέφικτο να διατυπωθεί ένα CQ που να περιγράφει πλήρως τη συμπεριφορά ενός ταξινομητή, ορίζουμε τις εξής έννοιες:

- Ένα query q_i το οποίο περιγράφει ακριβώς το σύνολο $\mathcal{C}_i$ ($\mathsf{cert}(q_i, \mathcal{K}) = \mathcal{C}_i$), λέμε ότι αποτελεί τέλεια εξήγηση (perfect explanation).

- Ένα query q_i το οποίο περιγράφει όλα τα στοιχεία του συνόλου $\mathcal{C}_i$, ($\mathsf{cert}(q_i, \mathcal{K}) \supseteq \mathcal{C}_i$) λέμε ότι αποτελεί υπέρ-εξήγηση (over-explanation).

- Ένα query q_i το οποίο περιγράφει μόνο στοιχεία του συνόλου $\mathcal{C}_i$, ($\mathsf{cert}(q_i, \mathcal{K}) \subseteq \mathcal{C}_i$) λέμε ότι αποτελεί υπό-εξήγηση (under-explanation).

- Ένα query q_i το οποίο περιγράφει κάποια στοιχεία του συνόλου $\mathcal{C}_i$, ($\mathsf{cert}(q_i, \mathcal{K}) \cap \mathcal{C}_i \neq \varnothing$) λέμε ότι αποτελεί προσεγγιστική εξήγηση (approximate explanation).

Κάθε perfect explanation αποτελεί και over/under-explanation. Τα approximate explanations έχουν οριστεί με ένα πολύ χαλαρό κριτήριο και αξιολογούνται με τη χρήση μετρικών ή με καθαρά ποιοτικά κριτήρια.

Μπορούμε να επεκτείνουμε τις παραπάνω έννοιες, και γενικά την έννοια της εξήγησης, και σε σύνολα από queries $Q_i = \{q_{i1}, q_{i2}, \ldots q_{in}\}$, τέτοια ώστε $\mathsf{cert}(Q_i, \mathcal{K}) = \mathcal{C}_i$. Είναι σχετικά τετριμμένο να κατασκευάσουμε ένα perfect explanation αυτού του είδους αν πάρουμε ως Q_i το σύνολο των msq των individual του συνόλου $\mathcal{C}_i$ (υπό την προϋπόθεση ότι κανένα από αυτά τα queries δεν επιστρέφει individuals που ανήκουν σε κάποια άλλη κλάση). Όμως το πλήθος των queries που θα άνηκαν σε εκείνο το σύνολο κάνει την εξήγηση δυσνόητη και επομένως μικρής χρησιμότητας. Στόχος μας είναι να βρούμε ένα σύνολο queries το οποίο με ποιοτικά κριτήρια θεωρούμε καλή περιγραφή του $\mathcal{C}_i$, αρκετά μικρό και αρκετά ευνόητο.

Ορίζουμε και κάποιες μετρικές για την ποσοτική αξιολόγηση εξηγήσεων. Με τον όρο positives αναφερόμαστε στους individuals που επέστρεψε ένα query και ανήκουν στην επιθυμητή κλάση ενώ με negatives στου υπόλοιπους.

- Με τον όρο precision θα αναφερόμαστε στην αναλογία $|\mathsf{cert}(q_i, \mathcal{K}) \cap \mathcal{C}_i|/|\mathsf{cert}(q_i, \mathcal{K})|$, την οποία μπορούμε να εκφράσουμε και ως $|\text{positives}|/(|\text{positives}| + |\text{negatives}|)$, δηλαδή το ποσοστό των απαντήσεων που είναι positive. Αυτή η μετρική δίνει βάρος στο πλήθος των negatives που επέστρεψε ένα query. Παίρνει τιμές από το 0 εώς το 1, με μεγαλύτερη τιμή να θεωρείται καλύτερη και κάθε under-explanation παίρνει τιμή 1.

- Με τον όρο recall θα αναφερόμαστε στην αναλογία $|\mathsf{cert}(q_i, \mathcal{K}) \cap \mathcal{C}_i|/|\mathcal{C}_i|$, δηλαδή των individual της επιθυμητής κλάσης που επέστρεψε το query. Αυτή η μετρική δίνει βάρος στο πλήθος των individual ανήκουν στην επιθυμητή κλάση και δεν επέστρεψε το query. Παίρνει τιμές από το 0 εώς το 1, με μεγαλύτερη τιμή να θεωρείται καλύτερη και κάθε over-explanation παίρνει τιμή 1.

- Με τον όρο degree θα αναφερόμαστε στην Jaccard ομοιότητα των συνόλων $\mathsf{cert}(q_i, \mathcal{K})$, $\mathcal{C}_i$, δηλαδή την ποσότητα $|\mathsf{cert}(q_i, \mathcal{K}) \cap \mathcal{C}_i|/|\mathsf{cert}(q_i, \mathcal{K}) \cup \mathcal{C}_i|$, την οποία μπορούμε να εκφράσουμε και ως $|\text{positives}|/(|\mathcal{C}_i| + |\text{negatives}|)$. Αυτή η μετρική δίνει εξίσου βάρος και στο πλήθος των negatives καθώς και στο πλήθος των individual ανήκουν στην επιθυμητή κλάση και δεν επέστρεψε το query. Παίρνει τιμές από το 0 εώς το 1, με μεγαλύτερη τιμή να θεωρείται καλύτερη και μόνο τα perfect-explanations παίρνουν τιμή 1.

3.2 Αλγόριθμοι

Algorithm 1: LCS

Input: Queries $q_1 = (L_1, E_1)$, $q_2 = (L_2, E_2)$, each represented as a vector of sets containing the node labels of the query graph and a set of adjacency matrices, one for each role.
Output: Query $q = (L, E)$, the least common subsummer of $q_1,\ q_2$.

```
1 foreach l_1 ∈ L_1 do
2 |   foreach l_2 ∈ L_2 do
3 |   |   Append l_1 ∩ l_2 to L
4 |   end
5 end
6 foreach role r, common to q_1, q_2 do
7 |   Add to E the Kronecker product of the adjacency matrices of r in E_1, E_2
8 end
9 return q = (V, E)
```

Έχουμε ορίσει τον least common subsumer και έχουμε εξηγήσει ότι υπολογίζεται από το γινόμενο Kronecker των γράφων των query αλλά με τον αλγόριθμο 1 δίνουμε κάποιες επιπλέον πληροφορίες για το πώς τον υπολογίζουμε στην πράξη. Για τα labels των γράφων χρησιμοποιούμε ένα διάνυσμα που περιέχει στη πρώτη θέση το $L(x)$ και στη θέση $i+1$ το $L(y_i)$. Οι ακμές των γράφων αναπαρίστανται ως ένα σύνολο από πίνακες γειτνίασης (adjacency matrices), έναν για κάθε ρόλο. Θεωρώντας το μέγεθος των CN, RN σταθερό και το πλήθος των μεταβλητών των q_1, q_2 ίσο με n, m αντίστοιχα, ο διπλός βρόχος στις γραμμές 1,2 περιέχει $O(nm)$ πράξεις, ενώ ο βρόχος 6 έχει στη χειρότερη περίπτωση $|\mathsf{RN}|$ υπολογισμούς γινομένου Kronecker πινάκων το οποίο έχει πολυπλοκότητα $O(n^2m^2)$. Άρα συνολικά, αυτή η υλοποίηση του lcs έχει πολυπλοκότητα $O(n^2m^2)$.

Αν και ο lcs δύο queries είναι ελάχιστος ως προς το subsumption, δεν είναι ελάχιστος και ως προς το πλήθος των μεταβλητών. Για δύο queries με πλήθος μεταβλητών n, m ο lcs ορίζεται ως ένα query με $n \cdot m$ μεταβλητές. Είναι επομένως μία πράξη που γρήγορα αυξάνει το μέγεθος των

Algorithm 2: MINIMIZE

Input: Query $q = (V, E, L)$ to be minimized.
Output: The minimized query q'.

```
1  n ← V
2  q' ← q
3  do
4  |  q ← q'
5  |  foreach pair 0 < i, j ≤ n, i ≠ j do
6  |  |  Check if unifying variable j of q' with variable i of q' would be the same as deleting
   |  |   it:
7  |  |  if L(v_j) ⊆ L(v_i) and ((v_j, r, v_k) ∈ E ⇒ (v_i, r, v_k) ∈ E, k ≠ j) and
   |  |   ((v_k, r, v_j) ∈ E ⇒ (v_k, r, v_i) ∈ E, k ≠ j) and ((v_j, r, v_j) ∈ E ⇒ (v_i, r, v_i) ∈ E)
   |  |   then
8  |  |  |  Delete variable j from q'.
9  |  |  end
10 |  end
11 while q' ≠ q
12 return q'
```

queries. Για να αντιμετωπίσουμε αυτό το φαινόμενο χρησιμοποιούμε έναν αλγόριθμο ελαχιστοποίησης. Όπως είχαμε αναφέρει στην ενότητα 2.3.2 το condensation ενός query, δηλαδή η ελαχιστοποίηση των όρων του, είναι NP-complete διαδικασία. Για το πλαίσιο των πειραμάτων μας όμως χρησιμοποιούμε τον πολυωνυμικό αλγόριθμο MINIMIZE ο οποίος αφαιρεί κάποιους όρους χωρίς απαραίτητα να εντοπίζει το ελάχιστο δυνατό query.

Για την ορθότητα του αλγορίθμου αρκεί να επιβεβαιώσουμε τον ισχυρισμό της γραμμής 6, δηλαδή ότι αφαιρούνται μεταβλητές των οποίων η αφαίρεση ισοδυναμεί με την ενοποίηση τους με κάποια άλλη. Κατά την ενοποίηση των μεταβλητών όλα τα conjuncts της μορφής $C(v_j)$ θα γινόντουσαν $C(v_i)$, ενώ τα conjunct των μορφών $r(v_j, v_k), r(v_k, v_j), r(v_j, v_j)$ θα γινόντουσαν $r(v_i, v_k)$, $r(v_k, v_i), r(v_i, v_i)$. Η γραμμή 7 ελέγχει ότι όλα τα νέα conjuncts υπάρχουν ήδη στο query και επομένως ότι η ενοποίηση των μεταβλητών αντιστοιχεί απλά στην διαγραφή των conjunct που περιέχουν τη μεταβλητή v_j. Η διαγραφή όρων από το q μας δίνει ένα νέο query q' που είναι ομομορφικό στο q με τον ταυτοτικό ομομορφισμό, άρα $q \leq_S q'$. Όμως και το q είναι ομομορφικό στο q' με ομομορφισμό τον $h : q \to q', \quad h(v_k) = v_k, k \neq j, \quad h(v_j) = v_i$, επομένως $q \geq_S q'$ και άρα $q \equiv_S q'$.

Ο αλγόριθμος σίγουρα τερματίζει αφού κάθε εκτέλεση του βρόχου 5 έχει πεπερασμένο πλήθος επαναλήψεων, ενώ ο βρόχος 3 εκτελείται μόνο μετά από μία διαγραφή μεταβλητής και εφόσον δουλεύουμε με queries με πεπερασμένο πλήθος μεταβλητών, έστω n, θα σταματήσει μετά από το πολύ n επαναλήψεις. Κάθε εκτέλεση του βρόχου 5 περιέχει $O(n^2)$ επαναλήψεις ενώ η συνθήκη 7 περιέχει $O(n)$ συγκρίσεις συνόλων και $2 \cdot |\mathsf{RN}|$ συγκρίσεις στηλών και γραμμών των πινάκων γειτνίασης του q. Αν θεωρήσουμε σταθερά σε μέγεθος τα CN, RN τότε η συνθήκη 7 περιέχει $O(n)$ πράξεις. Εφόσον ο βρόχος 3 θα εκτελεστεί το πολύ n φορές έχουμε συνολικά πολυπλοκότητα $O(n^4)$.

Ο αλγόριθμος 3 (EXPLAIN) παράγει υποψήφιες εξηγήσεις για σύνολα από individuals. Ξεκινάει με μία λίστα που περιέχει τα most specific queries για κάθε individual και στη συνέχεια με ευριστικά κριτήρια τα ομαδοποιεί ανά δύο. Κάθε νέο query που φτιάχνει αποθηκεύεται ως μία πιθανή εξήγηση. Με την ομαδοποίηση των queries, ομαδοποιεί στην ουσία τα msq από τα οποία προέκυψαν και επομένως και τους αντίστοιχους individual. Λόγω της επιμεριστικότητας του lcs κάθε query που παράγει είναι ο lcs του αντίστοιχου υποσυνόλου από individual. Επειδή για την εκφραστικότητα RL οι απαντήσεις ενός query αντιστοιχούν σε ομομορφισμούς προς το κανονικό μοντέλο, κάθε query που περιέχει στις απαντήσεις του αυτό το υποσύνολο από individual πρέπει

Algorithm 3: EXPLAIN

Input: A set of individuals $\{i_1, i_2, \dots i_n\} \subseteq \mathsf{IN}$ and a threshold t of maximum query size.
Output: A set of queries as explanations of the individuals.

```
1  explanations ← ∅
2  queries ← {msq(i_j)}_{j=1}^n
3  while 'queries' has two or more elements do
4  |   Find the least disjoint pair of queries:
5  |   q_1, q_2 ← arg min{ Disj(q, q') | q, q' ∈ queries}
6  |   Remove q_1, q_2 from 'queries'.
7  |   q ← Minimize(LCS(q_1, q_2))
8  |   if the number of variables in q is ≤ t then
9  |   |   explanations ← explanations ∪ {q}
10 |   |   queries ← queries ∪ {q}
11 |   end
12 end
13 return explanations
```

να είναι ομομορφικό προς τον lcs του και επομένως τα queries που υπολογίζουμε είναι κατά αυτήν την έννοια τα ελάχιστα πιθανά. Επομένως, αν το κατώφλι t της εισόδου είναι αρκετά μεγάλο και κάποιο από αυτά τα υποσύνολα ή ολόκληρο το σύνολο των individual της εισόδου έχει perfect explanation τότε ο αλγόριθμος το εντοπίζει. Το κατώφλι χρησιμοποιείται για πρακτικούς λόγους γιατί η εύρεση απαντήσεων queries είναι NP-complete ως προς το πλήθος των μεταβλητών και επομένως πολύ μεγάλα queries δεν έχουν μεγάλη πρακτική χρησιμότητα ως εξηγήσεις

Η ευριστική ομαδοποίηση που χρησιμοποιεί ο αλγόριθμος EXPLAIN στηρίζεται στον αλγόριθμο 4 (DISJ) που κάνει μία πολύ γενική εκτίμηση για το πόσο ξένα είναι δύο queries. Συγκεκριμένα, συγκρίνει κάθε μεταβλητή v_1 του q_1 με κάθε μεταβλητή v_2 του q_2 μετρώντας πόσα conjuncts εννοιών και ρόλων θα έχανε σίγουρα η v_1 αν ενοποιούνταν με τη v_2 και στη συνέχεια κάνει την αντίστροφη διαδικασία. Αν θεωρήσουμε ότι $|V_1| = n$, $|V_2| = m$ και τα μεγέθη των CN, RN σταθερά, ο διπλός βρόχος 3,5 έχει $n \cdot m$ το πλήθος επαναλήψεις και η γραμμή 6 έχει σταθερό χρόνο εκτέλεσης. Ο βρόχος 7 έχει $|\mathsf{RN}|$ το πλήθος επαναλήψεις, ενώ οι γραμμές 8 και 9 έχουν σταθερό χρόνο εκτέλεσης χάρη σε μία προεπεξεργασία που κάνουμε στα queries που αποθηκεύει το πλήθος των εισερχόμενων και εξερχόμενων ακμών που έχει κάθε μεταβλητή για κάθε ρόλο. Η γραμμή 15 έχει το ίδιο πλήθος πράξεων με τα παραπάνω. Επομένως, συνολικά ο αλγόριθμος έχει πολυπλοκότητα $O(nm)$.

Έχοντας τη πολυπλοκότητα του αλγορίθμου DISJ μπορούμε να αναλύσουμε και την πολυπλοκότητα του αλγορίθμου EXPLAIN. Θα θεωρήσουμε ότι το πλήθος των μεταβλητών που έχουν τα msq των query της εισόδου είναι φραγμένο από μία σταθερά m. Η δημιουργία των msq στη γραμμή 2 γίνεται με μία απλή συμπλήρωση του διανύσματος των labels και των πινάκων γειτνίασης του κάθε query και χρειάζεται συνολικά $O(nm^2)$ πράξεις. Το σύνολο 'queries' έχει μέγεθος n πριν την εκτέλεση του βρόχου 3 ενώ σε κάθε επανάληψη του βρόχου αφαιρούνται δύο στοιχεία και προστίθεται ένα αν εκτελεστεί η γραμμή 10. Επομένως, συνολικά εκτελούνται στη χειρότερη περίπτωση $n - 1$ επαναλήψεις. Για την γραμμή 5, αν αποθηκευτούν οι τιμές $disj(q, q')$ την πρώτη φορά που υπολογίζονται, αρκούν $O(n^2)$ πράξεις για να βρεθεί κάθε φορά τα q_1, q_2. Για την πρώτη φορά που θα υπολογιστούν οι τιμές χρειάζονται $O(m^2)$ πράξεις για κάθε ζευγάρι από queries ενώ υπάρχουν $n(n-1)/2$ ζευγάρια αρχικά και στην επανάληψη i με τη προσθήκη του νέου query δημιουργούνται $n - i$ νέα ζευγάρια όπου τα queries τώρα έχουν μέγιστο πλήθος μεταβλητών $t \geq m$. Άρα, όλες οι επαναλήψεις του βήματος 5 θα στοιχίσουν συνολικά $O(n^3 + n^2m^2 + n^2t^2) = O(n^3 + n^2t^2)$. Το βήμα 7 στοιχίζει $O(t^4)$ πράξεις λόγω του lcs και $O(t^4)$ πράξεις λόγω του minimization και τα βήματα 8-11 μπορούμε να θεωρήσουμε ότι εκτελούνται σε σταθερό χρόνο. Συνολικά ο αλγόριθμος έχει πολυπλοκότητα $O(n^3 + n^2t^2 + nt^4)$.

Algorithm 4: DISJ

Input: A pair of queries $q_1 = (V_1, E_1, L_1)$, $q_2 = (V_2, E_2, L_2)$ for which to calculate a very rough estimate of how disjoint they are.

Output: The estimate of the disjointness.

```
1  disj ← 0
2  For every variable in q1 find how much it differs in terms of labels and number of edges
     from its closest match in q2:
3  foreach v1 ∈ V1 do
4      min_diff ← +∞
5      foreach v2 ∈ V2 do
6          diff ← |L1(v1) \ L2(v2)|
7          for r ∈ RN do
8              diff ← diff
                 + max{|{(v1, r, u1) ∈ E1, u1 ∈ V1}| − |{(v2, r, u2) ∈ E2, u2 ∈ V2}|, 0}
9              diff ← diff
                 + max{|{(u1, r, v1) ∈ E1, u1 ∈ V1}| − |{(u2, r, v2) ∈ E2, u2 ∈ V2}|, 0}
10         end
11         min_diff ← min(min_diff, diff)
12     end
13     disj ← disj + min_diff
14 end
15 Repeat the above but with q1 and q2 reversed.
16 return disj
```

Επειδή τα queries που παράγονται μέσω του lcs αυξάνουν πολύ το πλήθος των μεταβλητών τους, και ο αλγόριθμος MINIMIZE πολλές φορές δεν παράγει ικανοποιητικές ελαχιστοποιήσεις, όπως θα δούμε και στα πειράματα, χρησιμοποιούμε και μία ακόμα τεχνική ενοποίησης query. Κρατάμε τα κοινά conjuncts δύο query αφού πρώτα κάνουμε κάποια μετονομασία των μεταβλητών του ενός χρησιμοποιώντας τα ονόματα των μεταβλητών του άλλου. Έστω δύο queries q_1, q_2 όπου το q_1 έχει μεταβλητές τα $y_1, y_2, \ldots, y_n$ και το q_2 έχει μεταβλητές τα $z_1, z_2, \ldots, z_m$ με $m \leq n$. Μία αντιστοίχηση (matching) των μεταβλητών του q_2 στις μεταβλητές του q_1 είναι μία 1-1 απεικόνιση από τα z_i στα y_i, π.χ. $z_1 \leftrightarrow y_3$, $z_2 \leftrightarrow y_6, \ldots$. Αντικαθιστούμε στα conjuncts του q_1 τα y_i με τα z_i που αντιστοιχούν σε αυτά και αν $m < n$ τους όρους που περιέχουν y_i στα οποία δεν αντιστοιχεί κανένα z_i τους διαγράφουμε και έπειτα κρατάμε τα κοινά conjuncts των δύο queries. Η αντιστοίχηση είναι στην ουσία μια μερική αντιμετάθεση (partial permutation) των μεταβλητών του q_1 με μήκος m, ενώ η συνολική διαδικασία είναι η εύρεση ενός κοινού υπογράφου των γράφων των q_1, q_2.

Για την εύρεση τέτοιων αντιστοιχίσεων χρησιμοποιούμε τον αλγόριθμο 5 (BESTMATCHING) ο οποίος εξαντλητικά εξετάζει κάθε πιθανή αντιστοίχιση, εκτός και αν είναι πάρα πολλές στο πλήθος οπότε και εξετάζει τις πρώτες t. Ο αλγόριθμος εκτελεί $n!/(n-m)!$ ή t επαναλήψεις του βρόχου 8, ότι είναι μικρότερο. Ο βρόχος 8 διατρέχει τα στοιχεία των L_1, E_1, L_2, E_2 και οπότε κάθε επανάληψη εκτελεί $O(n^2)$ πράξεις. Η πολυπλοκότητα του αλγορίθμου είναι επομένως $O(tn^2)$, αν και για μεγάλα t ο χρόνος εκτέλεσης θα εξαρτάται κυρίως από τις $n!/(n-m)!$ επαναλήψεις του βρόχου 8.

Ο αλγόριθμος 6 (EXPLAIN2) προκύπτει με την χρήση του αλγορίθμου BESTMATCHING αντί του DISJ στον αλγορίθμο EXPLAIN με κάποιες απαραίτητες αλλαγές. Τώρα αντικείμενο είναι η μεγιστοποίηση του matching αντί της ελαχιστοποίησης του disjointment. Έπειτα, αν και η προηγούμενη μέθοδος χρησιμοποιούσε ένα ευριστικό κριτήριο και υπολόγιζε λίγα queries, το BESTMATCHING δεν συγκρίνει απλά δύο queries αλλά υπολογίζει και την ένωση τους, οπότε παράγονται και πολλά περισσότερα queries τα οποία αποθηκεύονται γιατί θα ήταν ανώφελο απλά να απορρίπτονται. Με αυτές τις αλλαγές ο αλγόριθμος θυμίζει περισσότερο γενετικό παρά ευριστικό αλγόριθμο. Όλα τα

Algorithm 5: BestMatching

Input: A pair of queries $q_1 = (V_1, E_1, L_1)$, $q_2 = (V_2, E_2, L_2)$ for which to find the best matching of their variables and a threshold t for the maximum number of matchings to consider.

Output: The query that corresponds to the best variable matching of the input queries, and the count of its conjuncts.

1 **if** $|V_1| < |V_2|$ **then**
2 | swap q_1, q_2
3 **end**
4 $n \leftarrow |V_1|$
5 $m \leftarrow |V_2|$
6 max_count $\leftarrow 0$
7 best_matching $\leftarrow (1, 2, \ldots, m)$
8 **foreach** *partial permutation* p *of the variables of* $|V_1|$ *of size* m*, or for the first* t *such permutations if there are more than* t **do**
9 | $E_1' \leftarrow p \circ E_1,\ L_1' \leftarrow p \circ L_1$
10 | concept_count $\leftarrow \sum |L_1'(v) \cap L_2(v)|_{v \in V_2}$
11 | role_count $\leftarrow |E_1' \cap E_2|$
12 | count $\leftarrow$ concept_count + role_count
13 | **if** *count* $>$ *max_count* **then**
14 | | max_count $\leftarrow$ count
15 | | best_matching $\leftarrow p$
16 | **end**
17 **end**
18 $V \leftarrow V_2$, $E \leftarrow$ (best_matching$\circ E_1) \cap E_2$, $L \leftarrow$ (best_matching$\circ L_1) \cap L_2$
19 $q \leftarrow (V, E, L)$
20 **return** q, *max_count*

queries διασταυρώνονται αλλά μόνο τα επιτυχή ζεύγη διατηρούνται.

Ο βρόχος 4 του αλγορίθμου περιέχει $n(n-1)/2$ κλήσεις του αλγορίθμου BestMatching και επομένως εκτελεί $O(tn^4)$ πράξεις. Ο βρόχος 9 εκτελείται $n-2$ φορές αφού σε κάθε επανάληψη αφαιρείται ένα στοιχείο από το σύνολο 'queries'. Κάθε επανάληψη περιλαμβάνει $O(n)$ κλήσεις του αλγορίθμου BestMatching και επομένως η συνολική πολυπλοκότητα του αλγορίθμου είναι $O(tn^4)$.

3.3 Σύνολο Εξήγησης

Για να εντάξουμε την μεθοδολογία μας στο πλαίσιο της μηχανικής μάθησης, εισάγουμε την έννοια του συνόλου εξήγησης (explanation dataset). Το σύνολο εξήγησης είναι ένα σύνολο δεδομένων, συμβατό με τα σύνολα δεδομένων εκπαίδευσης, επικύρωσης και ελέγχου ενός ταξινομητή. Ο ταξινομητής μπορεί να δεχθεί δείγματα από αυτό το σύνολο δεδομένων στην είσοδο και να δώσει στην έξοδο προβλέψεις με τον ίδιο τρόπο που δίνει προβλέψεις για δείγματα από τα άλλα σύνολα δεδομένων. Όμως, το σύνολο αυτό είναι εμπλουτισμένο και με πληροφορία υψηλότερου επιπέδου, διατυπωμένη σε κάποια περιγραφική λογική, η οποία μπορεί να διαφωτίσει τον τρόπο με τον οποίο λειτουργεί ο ταξινομητής.

Δεν υπάρχει διαδεδομένη χρήση των περιγραφικών λογικών για τον εμπλουτισμό συνόλων δεδομένων στο πεδίο της μηχανικής μάθησης. Ένα κοντινό παράδειγμα είναι το Visual Genome (Krishna κ.ά. 2016), ένα σύνολο δεδομένων αποτελούμενο από εικόνες εμπλουτισμένες με πληροφορίες για τα αντικείμενα που περιέχουν, με τις ιδιότητές τους και τις σχέσεις που τα συνδέουν. Οι πληροφορίες αυτές δεν είναι διατυπωμένες σε κάποια περιγραφική λογική, αλλά εύκολα με-

Algorithm 6: EXPLAIN2

Input: A set of individuals $\{i_1, i_2, \dots i_n\} \subseteq \mathsf{IN}$ and a threshold t.
Output: A set of queries as explanations of the individuals.

explanations $\leftarrow \varnothing$
queries $\leftarrow \{\text{msq}(i_j)\}_{j=1}^{n}$
best_matchings $\leftarrow$ an $n \times n$ matrix
foreach *pair of queries,* $q_i, q_j, i \neq j$ **do**
    $q, c \leftarrow$ BESTMATCHING(q_i, q_j, t)
    explanations $\leftarrow$ explanations $\cup \{q\}$
    best_matching$[i, j] \leftarrow (q, c)$
end
while *'queries' has more than two elements* **do**
    Find the indices of the queries with the best variable matching:
    $k, l \leftarrow \arg\max\{\text{best\_matching}[i, j][1],\ i \neq j\}$
    Remove q_k from 'queries' and its row and column from 'best_matchings'.
    $q \leftarrow$best_matching$[i, j][0]$
    Replace q_l with q
    foreach *query* $q_i \in$ *'queries'* $\setminus \{q_l\}$ **do**
        $q', c' \leftarrow$ BESTMATCHING(q_l, q_i, t)
        explanations $\leftarrow$ explanations $\cup \{q'\}$
        best_matching$[l, i] \leftarrow (q', c')$
    end
end
return *explanations*

τατρέπονται σε τέτοια μορφή για να δημιουργηθεί ένα Abox. Ταυτόχρονα, οι όροι που χρησιμοποιούνται στις περιγραφές έχουν παρθεί από την οντολογία WordNet (Miller κ.ά. 1991) η οποία είναι taxonomy, δηλαδή περιέχει μόνο αξιώματα υπαγωγής εννοιών και μπορεί να χρησιμοποιηθεί ως Tbox. Υπάρχει επίσης το σύνολο δεδομένων CLEVR-Hans3, το οποίο χρησιμοποιούμε και στα πειράματα. Όπως και στο Visual Genome, οι εικόνες του CLEVR-Hans3 συνοδεύονται από πληροφορία η οποία εύκολα μπορεί να μετατραπεί σε κάποιο Abox, όμως δεν υπάρχει κάποιο σύνολο αξιωμάτων που να τη συνοδεύει για να λειτουργήσει ως Tbox.

Μία μεθοδολογία για την δημιουργία νέων συνόλων εξήγησης, την οποία έχουμε χρησιμοποιήσει και εμείς για το σύνολο MNIST, είναι η αξιοποίηση των παραδοσιακών μεθόδων εξαγωγής χαρακτηριστικών στη μηχανική μάθηση. Όπως είχαμε αναφέρει στην ενότητα 2.1.3, παλαιότερα χρησιμοποιούνταν ρηχά δίκτυα που είχαν όμως υψηλού επιπέδου πληροφορία στην είσοδο, την οποία κατασκεύαζε ο άνθρωπος. Οι τεχνικές εξαγωγής πληροφορίας που έχουν υποχωρήσει για την εκπαίδευση νευρωνικών δικτύων μπορούν να αξιοποιηθούν για την κατασκευή των υψηλού επιπέδου περιγραφών ενός συνόλου εξήγησης το οποίο θα βοηθήσει στην κατανόηση της λειτουργίας βαθιών δικτύων.

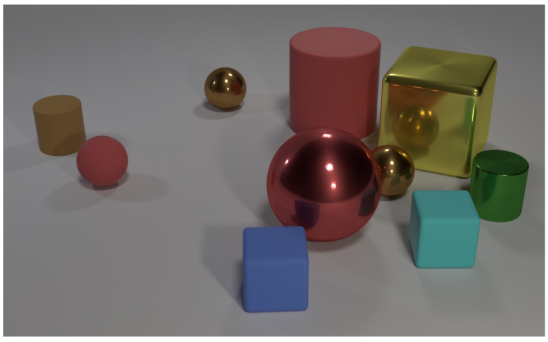
3.4 Σύνολα δεδομένων

3.4.1 CLEVR-Hans3

Το CLEVR-Hans3 είναι ένα σύνολο δεδομένων 13.500 εικόνων που εισηχθεί από τη δημοσίευση Stammer, Schramowski και Kersting 2021. Οι ερευνητές εργάζονταν και αυτοί στη περιοχή του XAI και μελετούσαν μεθόδους επέμβασης σε μοντέλα μηχανικής μάθησης που μαθαίνουν να παίρνουν αποφάσεις για τους λάθος λόγους. Το σύνολο δεδομένων CLEVR-Hans3 σχεδιάστηκε με τέτοιο τρόπο ώστε να είναι εύκολο για ένα μοντέλο να μάθει μια ανεπιθύμητη συμπεριφορά και

στη συνέχεια να φανεί αν αυτή η συμπεριφορά διορθώνεται με τις μεθόδους επέμβασης.

Πρόκειται για μία παραλλαγή του συνόλου δεδομένων CLEVR (Johnson κ.ά. 2016). Αποτελούνται και τα δύο από εικόνες που αναπαριστούν γεωμετρικά στερεά, τα οποία διαφέρουν μεταξύ τους ως προς το σχήμα, το μέγεθος, το χρώμα, το υλικό και τη διάταξη στο χώρο. Το CLEVR σχεδιάστηκε για εφαρμογές στη περιοχή της Απάντησης Οπτικών Ερωτήσεων (Visual Question Answering - VQA), ώστε να εκπαιδεύονται μοντέλα που μαθαίνουν να απαντούν ερωτήσεις για τα αντικείμενα που περιέχει μια εικόνα. Αντιθέτως, το CLEVR-Hans3 είναι ένα σύνολο δεδομένων για ένα πρόβλημα ταξινόμησης. Οι εικόνες που περιέχει μοιράζονται σε τρεις κλάσεις που κάθε μία περιέχει 3000 εικόνες εκπαίδευσης (training), 750 εικόνες επικύρωσης (validation) και 750 εικόνες ελέγχου (test). Οι εικόνες της πρώτης κλάσης περιέχουν πάντα έναν μεγάλο κύβο και έναν μεγάλο κύλινδρο, οι εικόνες της δεύτερης περιέχουν πάντα έναν μικρό μεταλλικό κύβο και μία μικρή σφαίρα και οι εικόνες της τρίτης μία μεγάλη μπλε σφαίρα και μία μικρή κίτρινη σφαίρα. Οι κλάσεις δεν είναι επικαλυπτόμενες, δηλαδή δεν υπάρχει εικόνα που να περιέχει έναν μεγάλο κύβο, έναν μεγάλο κύλινδρο (τα χαρακτηριστικά της πρώτης κλάσης), έναν μικρό μεταλλικό κύβο και μία μικρή σφαίρα (τα χαρακτηριστικά της δεύτερης). Οι εικόνες συνοδεύονται από μία αναλυτική περιγραφή σε μορφή αρχείου .json που αναφέρει πόσα και ποια αντικείμενα βρίσκονται σε κάθε εικόνα καθώς και πληροφορίες για τη διάταξη τους στο χώρο.

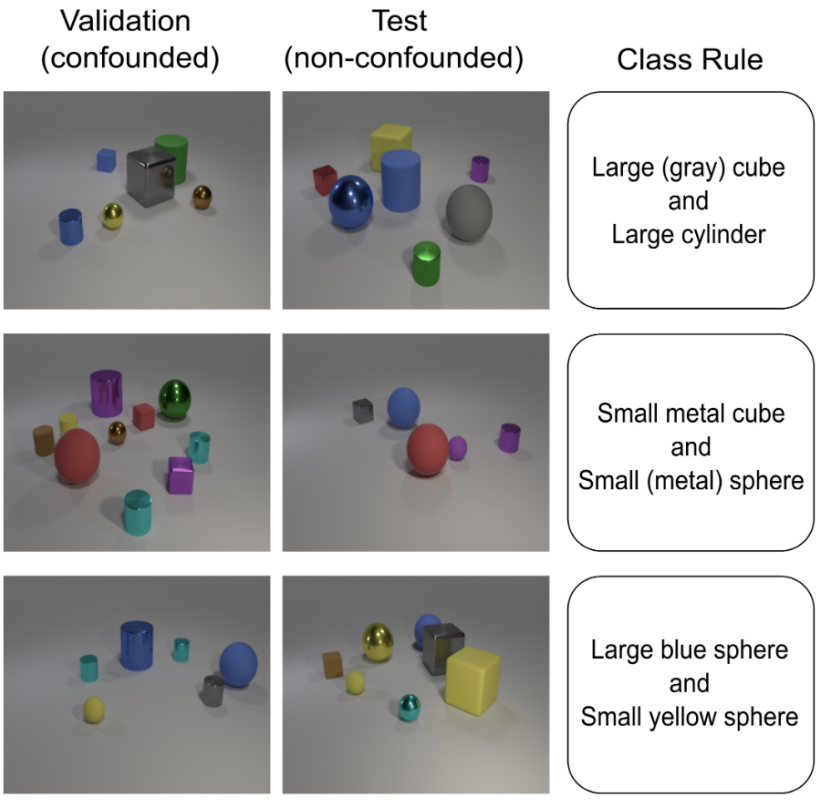


Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or metal things?

Σχήμα 3.1: Παράδειγμα VQA σε μία εικόνα του συνόλου δεδομένων CLEVR.

Η χρησιμότητα του CLEVR-Hans3 προέρχεται από το γεγονός ότι οι κλάσεις 1 και 2 περιέχουν ένα επιπλέον χαρακτηριστικό στα σύνολα εκπαίδευσης και επικύρωσης. Η κλάση 1 περιέχει μόνο γκρι κύβους και η κλάση 2 περιέχει μόνο μεταλλικές σφαίρες (σχήμα 3.2). Οι δημιουργοί του CLEVR-Hans3 εισήγαγαν αυτούς τους συγχυτικούς παράγοντες (confounding factors) στα σύνολα εκπαίδευσης και επικύρωσης ώστε να φαίνεται μια σημαντική διαφορά επίδοσης στο σύνολο ελέγχου για αυτές τις κλάσεις. Εμείς δεν μελετάμε μεθόδους επέμβασης στα μοντέλα που μαθαίνουν τους συγχυτικούς παράγοντες, μελετάμε όμως μεθόδους διάγνωσης. Αυτά τα στοιχεία μπορούμε να θεωρήσουμε ότι αντανακλούν λανθασμένη δειγματοληψία που συμβαίνει πολλές φορές σε πραγματικά σύνολα δεδομένων. Παραδείγματος χάριν, πολλά μοντέλα αναγνώρισης ανθρωπίνων προσώπων σε εικόνες εκπαιδεύονται με σύνολα δεδομένων που περιέχουν αποκλειστικά, ή σχεδόν αποκλειστικά ανθρώπους με ευρωπαϊκά χαρακτηριστικά. Ως αποτέλεσμα εργαλεία που χρησιμοποιούν αυτά τα μοντέλα δεν λειτουργούν σωστά για όλες τις πληθυσμιακές ομάδες όταν αρχίζουν να χρησιμοποιούνται από τον ευρύ πληθυσμό (Klare κ.ά. 2012).

Για τα δικά μας πειράματα έχουμε δημιουργήσει μια οντολογία που περιέχει τις περιγραφές των εικόνων του test set. Έχουμε δημιουργήσει έννοιες για τα διαφορετικά χαρακτηριστικά των



Σχήμα 3.2: Παραδείγματα από τις τρεις κλάσεις του CLEVR-Hans3. Σημειώνονται οι κανόνες που ακολουθούν οι κλάσεις με τους συγχυτικούς παράγοντες σε παρενθέσεις.

αντικειμένων. Συγκεκριμένα, η οντολογία περιλαμβάνει τις εξής έννοιες, ρόλους και αξιώματα:

$$
\begin{aligned}
\mathsf{CN} = \{&\mathsf{Image}, \mathsf{Object}, \mathsf{Cube}, \mathsf{Cylinder}, \mathsf{Sphere}, \mathsf{Metal}, \mathsf{Rubber}, \mathsf{Blue},\\
&\mathsf{Brown}, \mathsf{Cyan}, \mathsf{Gray}, \mathsf{Green}, \mathsf{Purple}, \mathsf{Red}, \mathsf{Yellow}, \mathsf{Large}, \mathsf{Small}\},\\
\mathsf{RN} = \{&\mathsf{contains}(\mathsf{Image}, \mathsf{Object})\}\\
\mathcal{T} = \{&\mathsf{Cube} \sqsubseteq \mathsf{Object}, \mathsf{Cylinder} \sqsubseteq \mathsf{Object}, \mathsf{Sphere} \sqsubseteq \mathsf{Object}, \mathsf{Metal} \sqsubseteq \mathsf{Object},\\
&\mathsf{Rubber} \sqsubseteq \mathsf{Object}, \mathsf{Blue} \sqsubseteq \mathsf{Object}, \mathsf{Brown} \sqsubseteq \mathsf{Object}, \mathsf{Cyan} \sqsubseteq \mathsf{Object},\\
&\mathsf{Gray} \sqsubseteq \mathsf{Object}, \mathsf{Green} \sqsubseteq \mathsf{Object}, \mathsf{Purple} \sqsubseteq \mathsf{Object}, \mathsf{Red} \sqsubseteq \mathsf{Object},\\
&\mathsf{Yellow} \sqsubseteq \mathsf{Object}, \mathsf{Large} \sqsubseteq \mathsf{Object}, \mathsf{Small} \sqsubseteq \mathsf{Object}\}
\end{aligned}
$$

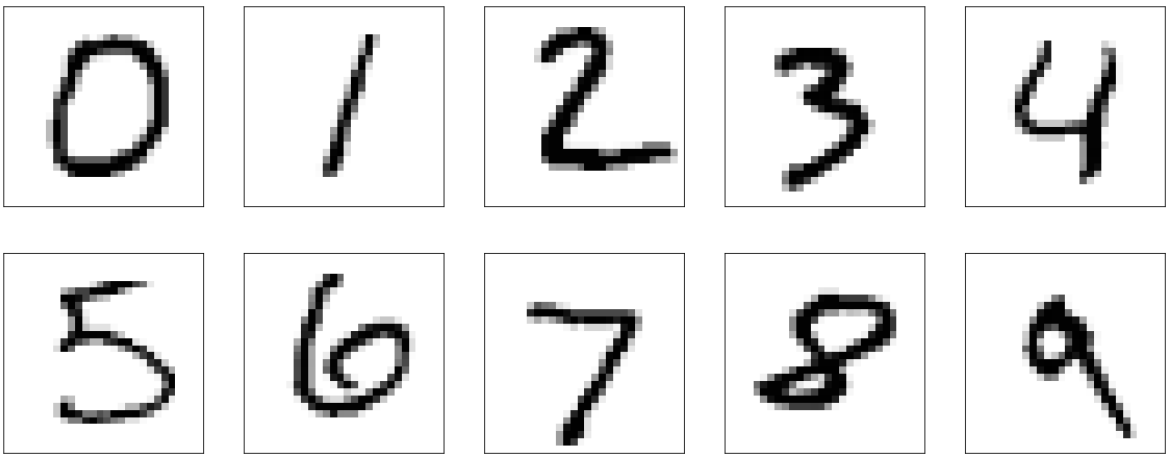
Επιπλέον, το IN περιέχει έναν individual για κάθε εικόνα και αντικείμενο.

Για τη κατασκευή του Abox εξάγουμε τις περιγραφές των εικόνων από το αρχείο .json του CLEVR-Hans3. Ο individual της κάθε εικόνας σημειώνεται ως Image και ο individual κάθε αντικειμένου ως Object. Ο individual κάθε εικόνας σημειώνεται ότι περιέχει (contains) τους individual των αντικειμένων της. Ο individual κάθε αντικειμένου χαρακτηρίζεται με βάση το σχήμα του με τις έννοιες Cube, Cylinder, Sphere, το υλικό του με τις έννοιες Metal, Rubber, το χρώμα του με τις έννοιες Blue, Brown, Cyan, Gray, Green, Purple, Red, Yellow, και το μέγεθος του με τις έννοιες Large, Small.

3.4.2 MNIST

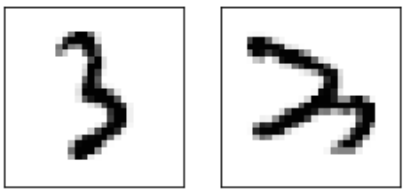
Η βάση δεδομένων MNIST (Modified National Institute of Standards and Technology database) είναι μία βάση δεδομένων που αποτελείται από 70.000 εικόνες χειρόγραφων ψηφίων, 0-9. Δημιουργήθηκε το 1998 για την εκπαίδευση ενός CNN στην Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition - OCR), LeCun κ.ά. 1998. Οι εικόνες έχουν όλες διαστάσεις 28×28 πίξελ και

είναι ασπρόμαυρες. Το σύνολο δεδομένων είναι μοιρασμένο σε ένα σύνολο εκπαίδευσης 60.000 εικόνων και ένα σύνολο ελέγχου 10.000 εικόνων.



Σχήμα 3.3: Δείγμα 10 ψηφίων από το test set του MNIST.

Στα πλαίσια των δικών μας πειραμάτων έχουμε επιλέξει ένα υποσύνολο 250 εικόνων από το test set και το έχουμε εμπλουτίσει με μία σχηματική περιγραφή των ψηφίων. Οι εικόνες επιλέχθηκαν με στόχο τη ποικιλία και έναν συνδυασμό τυπικών και δύσκολων παραδειγμάτων (σχήμα 3.4). Τα ψη-



Σχήμα 3.4: Δύο δείγματα του ψηφίου 3 που επιλέχθηκαν.

φία περιγράφονται ως ένα σύνολο γραμμών που περιέχει η εικόνα με χαρακτηριστικά τη γωνία, τη θέση και το μήκος τους. Περιγράφεται ακόμα ποιες γραμμές τέμνονται μεταξύ τους. Συγκεκριμένα, η οντολογία περιλαμβάνει τις εξής έννοιες, ρόλους και αξιώματα:

$$
\begin{aligned}
\mathsf{CN} = \{&\mathsf{Image}, \mathsf{Line}, \mathsf{Line0deg}, \mathsf{Line45deg}, \mathsf{Line90deg}, \mathsf{Line135deg}, \mathsf{TopLeft}, \mathsf{TopCenter}, \mathsf{TopRight},\\
&\mathsf{MidLeft}, \mathsf{MidCenter}, \mathsf{MidRight}, \mathsf{BotLeft}, \mathsf{BotCenter}, \mathsf{BotRight}, \mathsf{Short}, \mathsf{Medium}, \mathsf{Long}\}\\
\mathsf{RN} = \{&\mathsf{contains}(\mathsf{Image}, \mathsf{Line}), \mathsf{intersects}(\mathsf{Line}, \mathsf{Line})\}\\
\mathcal{T} = \{&\mathsf{Line0deg} \sqsubseteq \mathsf{Line}, \mathsf{Line45deg} \sqsubseteq \mathsf{Line}, \mathsf{Line90deg} \sqsubseteq \mathsf{Line}, \mathsf{Line135deg} \sqsubseteq \mathsf{Line},\\
&\mathsf{TopLeft} \sqsubseteq \mathsf{Line}, \mathsf{TopCenter} \sqsubseteq \mathsf{Line}, \mathsf{TopRight} \sqsubseteq \mathsf{Line}, \mathsf{MidLeft} \sqsubseteq \mathsf{Line},\\
&\mathsf{MidCenter} \sqsubseteq \mathsf{Line}, \mathsf{MidRight} \sqsubseteq \mathsf{Line}, \mathsf{BotLeft} \sqsubseteq \mathsf{Line}, \mathsf{BotCenter} \sqsubseteq \mathsf{Line},\\
&\mathsf{BotRight} \sqsubseteq \mathsf{Line}, \mathsf{Short} \sqsubseteq \mathsf{Line}, \mathsf{Medium} \sqsubseteq \mathsf{Line}, \mathsf{Long} \sqsubseteq \mathsf{Line}\}
\end{aligned}
$$

Ο ρόλος intersects είναι συμμετρικός, δηλαδή αν $\mathsf{intersects}(i_1, i_2)$ τότε και $\mathsf{intersects}(i_2, i_1)$. Επιπλέον, το IN περιέχει έναν individual για κάθε εικόνα και γραμμή.

Για την εξαγωγή της περιγραφής πρώτα χαρακτηρίζονται τα πίξελ της εικόνας ως προς το αν ανήκουν σε κάποια γραμμή του ψηφίου και με τη γωνία της αντίστοιχης γραμμής. Αυτό γίνεται με αυτόματο τρόπο, με τη διαδικασία που περιγράφεται στην ενότητα 3.5 και με επεμβάσεις με το χέρι όπου χρειάζεται διόρθωση. Στη συνέχεια οι γωνίες των πίξελ κβαντοποιούνται στις γωνίες $0°, 45°, 90°$ και $135°$ και τα πίξελ ομαδοποιούνται σε συνεκτικές συνιστώσες με βάση τη κβαντοποιημένη γωνία. Οι πολύ μικρές συνεκτικές συνιστώσες απορρίπτονται ως ατέλειες και οι κοντινές συνιστώσες ίδιας γωνίας θεωρούνται τεμαχισμένες γραμμές και οπότε συνενώνονται. Στη

συνέχεια, υπολογίζεται το μήκος της κάθε γραμμής με βάση τις διαστάσεις του κουτιού οριοθέτησης. Τέλος, οι πληροφορίες της γωνίας, του μήκους και των πίξελ που ανήκουν σε μία γραμμή χρησιμοποιούνται για να τη χαρακτηρίσουν ως προς τις έννοιες της οντολογίας.

Στο Abox της γνώσης ο individual της κάθε εικόνας σημειώνεται ως Image και ο individual κάθε γραμμής ως Line. Ο individual κάθε εικόνας σημειώνεται ότι περιέχει (contains) τους individuals των γραμμών της. Όλες οι γραμμές χαρακτηρίζονται με βάση την γωνία τους με τις έννοιες Line0deg, Line45deg, Line90deg, Line135deg και με βάση το μήκος τους με τις έννοιες Short, Medium, Long. Για τη θέση των γραμμών, οι εικόνες χωρίζονται σε τρεις οριζόντιες (Top , Mid, Bot-tom) και τρεις κατακόρυφες ζώνες (Left, Center, Right), εννέα περιοχές συνολικά και κάθε γραμμή χαρακτηρίζεται με την αντίστοιχη έννοια για κάθε περιοχή την οποία διασχίζει, ακόμα και για ένα μόνο πίξελ. Τέλος, δύο γραμμές σημειώνεται να τέμνονται (intersects) αν τα πίξελ τους είναι γειτονικά. Γίνεται όμως πρώτα μία πάχυνση των γραμμών γιατί κατά το στάδιο της αφαίρεσης των μικρών συνεκτικών συνιστωσών μπορεί να δημιουργηθούν μικρά κενά. Η συνολική διαδικασία απεικονίζεται στο σχήμα 3.5.

3.5 Εντοπισμός Γραμμών

Για τον εντοπισμό των γραμμών στις εικόνες του MNIST χρησιμοποιούμε την τεχνική του εντοπισμού κορυφογραμμής (ridge detection). Έστω $I(x, y)$ η εικόνα στην οποία θέλουμε να εντοπίσουμε γραμμές και $L(x, y)$ η ίδια εικόνα συνελιγμένη με ένα γκαουσσιανό φίλτρο.

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

Η μορφή μιας εξιδανικευμένης γραμμής είναι αυτή που εμφανίζεται στο σχήμα 3.6c. Μπορούμε να εξετάσουμε αν μια περιοχή της εικόνας προσεγγίζει τοπικά αυτό το σχήμα χρησιμοποιώντας το ανάπτυγμα Taylor δευτέρου βαθμού.

$$L(x, y) \approx \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} L_x & L_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + c \tag{3.1}$$

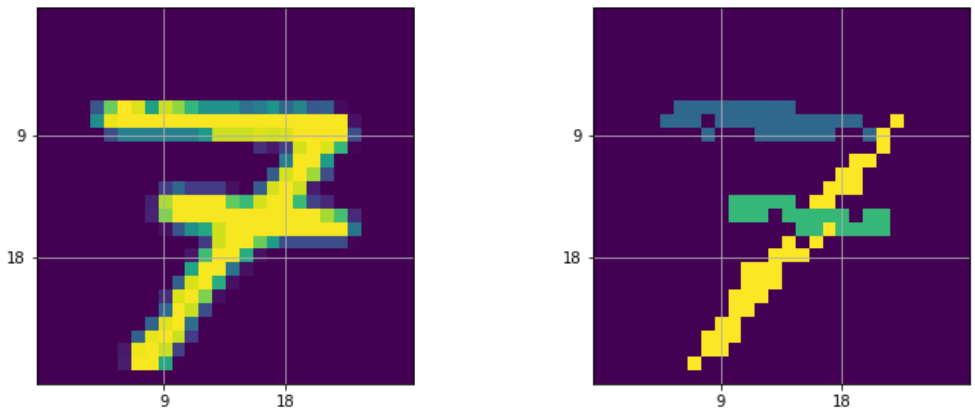
$L(x, y) = L_{xx} \cdot x^2 + L_x \cdot x + L_{yy} \cdot y^2 + L_y \cdot y + c = L_{xx}(x + c_x)^2 + L_{yy}(y + c_y)^2 + c'$. Σε αυτή τη μορφή είναι εύκολο να καταλάβουμε ποιο από τα σχήματα 3.6 (a)-(d) προσεγγίζει τοπικά εκείνο το σημείο με βάση το πρόσημο και το μέγεθος των L_{xx}, L_{yy}. Αυτό το σύστημα συντεταγμένων υπάρχει πάντα επειδή ο Εσσιανός πίνακας της εξίσωσης 3.1 είναι πραγματικός συμμετρικός και επομένως αναλύσιμος σε ορθοκανονικά ιδιοδιανύσματα.

$$\begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} = \mathbf{Q} \cdot \mathbf{\Lambda} \cdot \mathbf{Q^T}, \quad \mathbf{Q} = \begin{bmatrix} \sin\phi & \cos\phi \\ -\cos\phi & \sin\phi \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} L_{pp} & 0 \\ 0 & L_{qq} \end{bmatrix} \tag{3.2}$$

Οι 3.1 και 3.2 μας δίνουν:

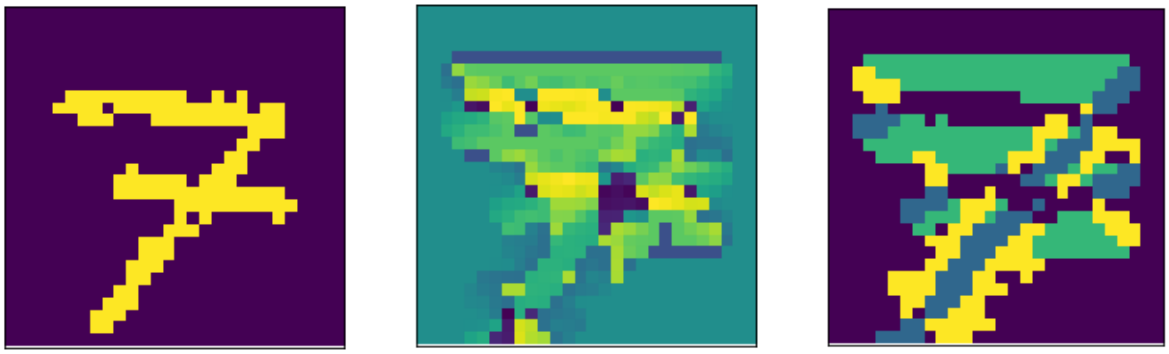
$$\begin{aligned}
&L(x, y) \approx \begin{bmatrix} x & y \end{bmatrix} \cdot \mathbf{Q} \cdot \mathbf{\Lambda} \cdot \mathbf{Q}^T \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} L_x & L_y \end{bmatrix} \cdot \mathbf{Q}^T \cdot \mathbf{Q} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + c \Leftrightarrow \\
&L(p, q) \approx \begin{bmatrix} p & q \end{bmatrix} \begin{bmatrix} L_{pp} & 0 \\ 0 & L_{qq} \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + \begin{bmatrix} L_p & L_q \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + c \Leftrightarrow \\
&L(p, q) \approx L_{pp} \cdot p^2 + L_p \cdot p + L_{qq} \cdot q^2 + L_q \cdot q + c, \\
&L_{pp} = \partial_{pp} L, \quad L_{qq} = \partial_{qq} L, \quad L_p = \partial_p L, \quad L_q = \partial_q L, \\
&\partial_p = \sin\phi \partial_x - \cos\phi \partial_y, \quad \partial_q = \cos\phi \partial_x + \sin\phi \partial_y
\end{aligned}$$

Θεωρόντας ότι η L_{pp} είναι η μικρή ιδιοτιμή και επειδή το σχήμα της γραμμής θέλουμε να έχει τα κοίλα προς τα κάτω, θέλουμε να ισχύει $L_{pp} < 0$ και $L_{qq} \approx 0$. Επιπλέον, για να κρατήσουμε μόνο

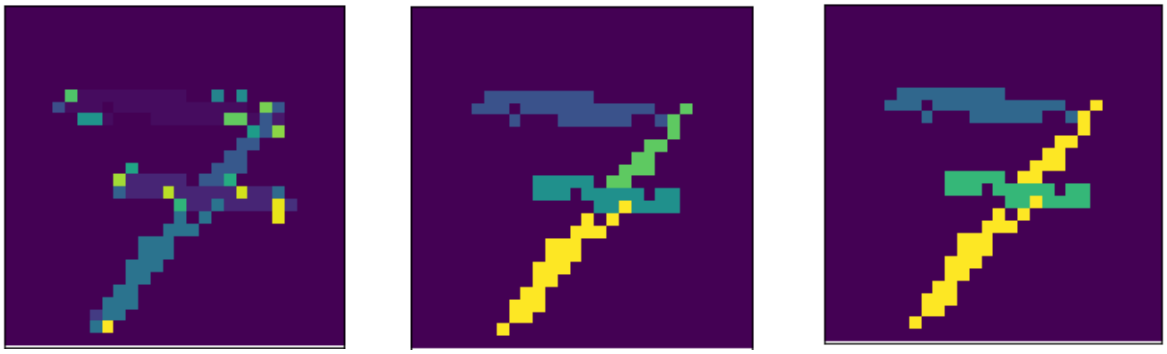



```
image contains 3 lines
line 0 is: Line0deg, TopLeft, TopCenter, TopRight, MidLeft, MidCenter, MidRight, Long.
line 1 is: Line0deg, MidCenter, MidRight, Long.
line 2 is: Line45deg, TopRight, MidCenter, MidRight, BotLeft, BotCenter, Long.
line 0 intersects line 2
line 1 intersects line 2
```

(a)



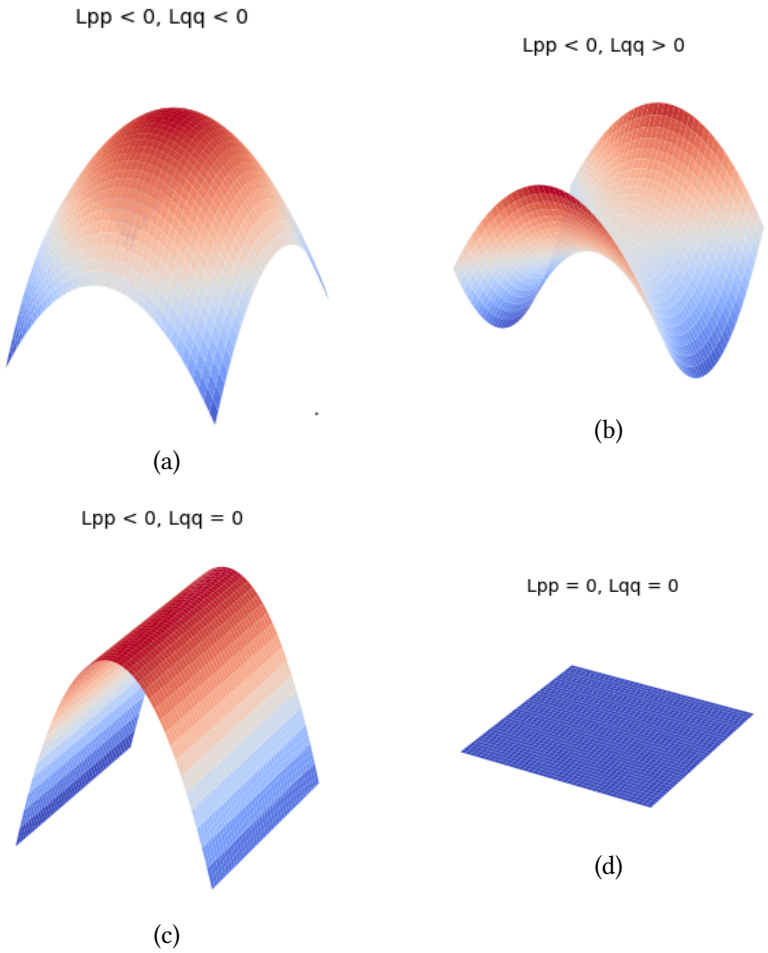
(b) (c) (d)



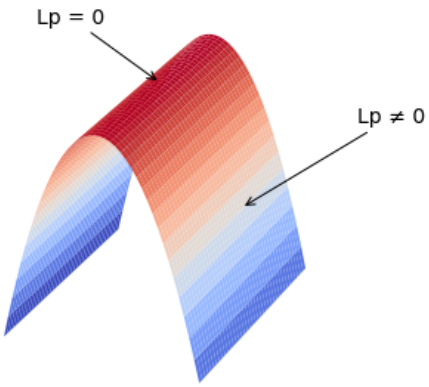
(e) (f) (g)

Σχήμα 3.5: Η περιγραφή που εξάγεται από ένα ψηφίο (a) και τα ενδιάμεσα στάδια (b-g). Εντοπισμός πίξελ γραμμών (b) και γωνιών (c), κβαντοποίηση των γωνιών (d), οι συνεκτικές συνιστώσες πριν (e) και μετά (f) την αφαίρεση των μικρών συνιστωσών, και οι συνιστώσες μετά τη συνένωση των τεμαχισμένων γραμμών (g).

τα σημεία στη κορυφή της γραμμής θέλουμε $L_p = 0$, σχήμα 3.7. Στο Lindeberg 2008 αναφέρεται ότι αρκούν οι σχέσεις $L_{pp} < 0$, $L_{pp} < L_{qq}$ και $L_p \approx 0$, τις οποίες εφαρμόζουμε και εμείς με μεγάλη επιτυχία. Η γωνία της γραμμής δίνεται από τα $\cos\phi, \sin\phi$.



Σχήμα 3.6: Τα πιθανά σχήματα του αναπτύγματος Taylor δευτέρου βαθμού μίας συνάρτησης δύο μεταβλητών, μοναδικά έως κάποιας ανάκλασης ή περιστροφής.



Σχήμα 3.7: Τα σημεία στο κέντρο της γραμμής είναι αυτά με $L_p = 0$.

Κεφάλαιο 4

Πειράματα

4.1 CLEVR-Hans3

4.1.1 Ταξινομητές

Το CLEVR-Hans3 έχει ιδιαίτερο ενδιαφέρον στη παραγωγή queries ως εξηγήσεις επειδή οι κλάσεις του περιγράφονται από κανόνες που είναι διατυπώσιμοι στην γλώσσα των queries. Σαν πρώτο ταξινομητή έχουμε επιλέξει έναν εικονικό τέλειο ταξινομητή που κατατάσσει όλα τα δείγματα στις κλάσεις που πραγματικά ανήκουν. Έχουμε κάνει αυτήν την επιλογή για να δούμε σαν ένα πρώτο εύκολο πείραμα όταν ξέρουμε εκ των προτέρων ότι υπάρχει κάποιο query που αποτελεί τέλεια εξήγηση αν ο αλγόριθμος Explain μπορεί να το εντοπίσει.

Έχουμε επιλέξει και έναν πραγματικό ταξινομητή, το CNN που χρησιμοποίησαν στα πειράματα τους οι ερευνητές που δημιούργησαν το CLEVR-Hans3 (Stammer, Schramowski και Kersting 2021). Ο ταξινομητής είναι ένα βαθύ CNN τύπου ResNet και συγκεκριμένα η αρχιτεκτονική ResNet34 (He κ.ά. 2015). Η διαδικασία εκπαίδευσης που ακολουθήσαμε ήταν η ίδια με αυτή των Stammer, Schramowski και Kersting 2021. Η αρχιτεκτονική ResNet34 υπάρχει ευρέως διαθέσιμη προεκπαιδευμένη στο σύνολο δεδομένων ImageNet (Deng κ.ά. 2009). Χρησιμοποιούμε το προεκπαιδευμένο δίκτυο στο ImageNet για να βελτιώσουμε την απόδοση του ταξινομητή στο CLEVR-Hans3 στα πλαίσια του transfer learning (Goodfellow, Bengio και Courville 2016). Αφαιρούμε τα τελευταία 6 στρώματα του δικτύου και προσθέτουμε ένα γραμμικό στρώμα στην έξοδο. Το μοντέλο εκπαιδεύτηκε για 100 εποχές με την τεχνική ελαχιστοποίησης της εγκάρσιας εντροπίας (cross-entropy minimization, Rubinstein και Kroese 2004) και τον αλγόριθμο βελτιστοποίησης Adam (Kingma και Ba 2017) με τις εξής τιμές παραμέτρων, batch size $= 64$, learning rate $= 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay $= 0$.

Στον πίνακα 4.1 βλέπουμε κάποιες μετρικές για τον ταξινομητή. Ως accuracy ορίζεται το ποσοστό των ταξινομήσεων που ήταν σωστές, ως precision τι ποσοστό από τα δείγματα που ταξινομήθηκαν στην εκάστοτε κλάση ανήκουν όντως σε εκείνη τη κλάση, ως recall τι ποσοστό των δειγμάτων της εκάστοτε κλάσης ταξινομήθηκαν σε εκείνη την κλάση και F1-score ο μέσος όρος του precision και του recall. Ως πίνακας σύγχυσης (confusion matrix) ορίζεται ο πίνακας που συγκρίνει τις πραγματικές κλάσεις στις οποίες ανήκουν τα δείγματα με αυτές στις οποίες ταξινομήθηκαν. Παρατηρούμε την αναμενόμενη πτώση στο accuracy στο σύνολο ελέγχου σε σχέση με τα εκπαίδευσης και επικύρωσης λόγω των συγχυτικών παραγόντων το οποίο επιβεβαιώνεται και από το πίνακα (b) καθώς η κλάση 3 που δεν έχει συγχυτικό παράγοντα διατηρεί υψηλές επιδόσεις. Η κλάση με τη μεγαλύτερη πτώση είναι η 1, αυτό εικάζουμε ότι οφείλεται στο ότι ο συγχυτικός παράγοντας του γκρι χρώματος είναι πολύ πιο απλός από τον συγχυτικό παράγοντα της μεταλλικής όψης και επομένως φαίνεται να στηριζόταν αρκετά σε αυτόν ο ταξινομητής κατά την εκπαίδευση. Παρατηρούμε επίσης πολύ υψηλό precision στην κλάση 1 και πολύ υψηλό recall στις κλάσεις 2 και 3. Θα μπορούσαμε να πούμε κατά αναλογία με το δικό μας πλαίσιο ότι ο ταξινομητής έχει βρει under-explanation για τη κλάση 1 και over-explanations για τις κλάσεις 2 και 3.

	Accuracy
Training set	100%
Validation set	99.4%
Test set	71.2%

(a) Η ακρίβεια του ταξινομητή στα σύνολα εκπαίδευσης, επικύρωσης και ελέγχου.

	Precision	Recall	F1-score
Class 1	0.94	0.16	0.27
Class 2	0.59	0.98	0.54
Class 3	0.85	1.00	0.92

(b) Οι μετρικές precision, recall και F1-score για τις τρεις διαφορετικές κλάσεις στο σύνολο ελέγχου.

	Predicted label		
True label	Class 1	Class 2	Class 3
Class 1	118	511	121
Class 2	5	736	9
Class 3	2	0	748

(c) Ο πίνακας σύγχυσης του ResNet34.

Πίνακας 4.1: Οι μετρικές του ResNet34 στο CLEVR-Hans3.

	Ground Truth Classifier			ResNet34 Classifier		
	Query Creation	Query Answering	Total	Query Creation	Query Answering	Total
Class 1	35s	43s	78s	1.3s	5.3s	6.6s
Class 2	35s	25s	60s	95s	119s	214s
Class 3	35s	22s	57s	50s	36s	86s

Πίνακας 4.2: Οι χρόνοι υπολογισμού των queries με τον αλγόριθμο Explain και εύρεσης των απαντήσεων με το GraphDB για τον εικονικό ταξινομητή και το ResNet34.

4.1.2 Παραγωγή εξηγήσεων

Οι εξηγήσεις παράγονται με την χρήση του αλγορίθμου Explain και στη συνέχεια τα certain answers των query υπολογίζονται με την χρήση του προγράμματος GraphDB. Στον πίνακα 4.2 παρουσιάζουμε τους χρόνους υπολογισμού. Για τον αλγόριθμο Explain η παράμετρος t είχε τεθεί ίση με 20, όμως δεν επηρέασε τους χρόνους εκτέλεσης καθώς κανένα από τα queries που δημιουργήθηκαν δεν ξεπέρασε αυτό το πλήθος μεταβλητών. Μάλιστα, κανένα query δεν ξεπέρασε τις 17 μεταβλητές. Παρατηρούμε μεγάλη διαφορά στους χρόνους εκτέλεσης του αλγορίθμου Explain μεταξύ των κλάσεων 1 και 2 για τις εξόδους του ResNet34. Αυτό οφείλεται στο γεγονός ότι το ResNet34 ταξινόμησε λίγα δείγματα στη κλάση 1 (125) και πολλά περισσότερα στη κλάση 2 (1247), καθώς και ότι ο αλγόριθμος έχει κυβική πολυπλοκότητα ως προς το πλήθος των individual της εισόδου. Μικρή αύξηση για τον ίδιο λόγο είχε και η κλάση 3.

4.1.3 Αξιολόγηση

Στον πίνακα 4.1 βλέπουμε τα καλύτερα query ως προς τις μετρικές που ορίσαμε στην ενότητα 3.1. Παρατηρούμε ότι για τον εικονικό ταξινομητή τα query που παίρνουμε περιγράφουν ακριβώς τους κανόνες που ακολουθούν οι τρεις κλάσεις, χωρίς τους συγχυτικούς παράγοντες καθώς ερ-

Metric	Query	Precision	Recall	Degree	Positives	Negatives
Class 1						
Best Precision, Recall, Degree	x contains y1, y2. y1 is Large, Cylinder, Object. y2 is Large, Object, Cube.	1.00	1.00	1.00	750	0
Class 2						
Best Precision, Recall, Degree	x contains y1, y2. y1 is Small, Object, Sphere. y2 is Small, Metal, Object, Cube.	1.00	1.00	1.00	750	0
Class 3						
Best Precision, Recall, Degree	x contains y1, y2. y1 is Yellow, Small, Object, Sphere. y2 is Large, Object, Blue, Sphere.	1.00	1.00	1.00	750	0

(a) Ground Truth

γαζόμαστε στο σύνολο ελέγχου και προφανώς μεγιστοποιούν ταυτόχρονα όλες τις μετρικές αφού αποτελούν τέλειες εξηγήσεις. Για το ResNet34 και την κλάση 1 παρατηρούμε ότι όλες οι εικόνες που ταξινομεί σε αυτήν περιέχουν μεγάλους κύβους καθώς και ότι οι περισσότερες περιέχουν συγκεκριμένα μεγάλους γκρι κύβους και μεγάλα μεταλλικά αντικείμενα. Αυτά τα χαρακτηριστικά ξεφεύγουν από τον κανόνα και επομένως μπορούν να αποτελέσουν την αρχή μιας διαγνωστικής διαδικασίας που θα εντοπίσει σε ποια χαρακτηριστικά στοχεύει λανθασμένα ο ταξινομητής. Πρέπει να σημειωθεί ότι οι μεταβλητές $y_1, y_2, \ldots$ δεν αναφέρονται αναγκαστικά σε διαφορετικά αντικείμενα. Ενδεχομένως σε κάποιες από τις εικόνες που είναι απαντήσεις του under-explanation το y_3 να αναφέρεται στο ίδιο αντικείμενο με το y_1. Η απουσία των κυλίνδρων στο over-explanation υπονοεί ότι κατά την εκπαίδευση στηριζόταν περισσότερο στους κύβους για να κάνει τις ταξινομήσεις. Για τη κλάση 2 παρατηρούμε ότι στο over-explanation εμφανίζεται και πάλι μόνο ο κύβος και μάλιστα αυτή τη φορά χωρίς το μικρό μέγεθος που περιέχει ο κανόνας. Μία πρώτη ερμηνεία που στηρίζεται και στο confusion matrix είναι ότι τις εικόνες της κλάσης 1 που δεν περιέχουν γκρι κύβους τις ταξινόμησε κυρίως ως δείγματα της κλάσης 2 και για αυτόν τον λόγο δεν διατηρήθηκαν τα χαρακτηριστικά μικρός και μεταλλικός για τους κύβους. Για τη κλάση 3 έχει ενδιαφέρον ότι εμφανίζεται ο πραγματικός κανόνας ως approximate explanation. Όπως έχουμε δει και από το confusion matrix δύο δείγματα που τηρούν αυτόν τον κανόνα ταξινομούνται λανθασμένως ως δείγματα της κλάσης 1 και θα είχε ενδιαφέρον να διερευνήσουμε το γιατί.

Μπορούμε να εξετάσουμε και πιο ποιοτικά κριτήρια. Καταρχάς, εξετάζοντας το under-explanation της κλάσης 1 είχαμε επισημάνει ότι εμφανίζεται το γκρι χρώμα στους κύβους και κάποιο μεγάλο μεταλλικό αντικείμενο, χαρακτηριστικά που δεν περιέχονται στον επιθυμητό κανόνα. Εξετάζουμε την σημασία που έχουν αυτά τα χαρακτηριστικά στον ταξινομητή κατασκευάζοντας κάποια δικά μας query, πίνακας 4.4. Βλέπουμε ότι αν αφαιρέσουμε την απαίτηση να είναι γκρι ο κύβος αυξάνονται πολύ οι negatives και λίγο οι positives, επομένως φαίνεται ότι το γκρι χρώμα ήταν πολύ σημαντικό κομμάτι του under-explanation. Από την άλλη, αφαιρώντας το μεγάλο μεταλλικό αντικείμενο οι negatives δεν αυξάνονται καθόλου ενώ εμφανίζονται μερικοί ακόμα positives. Ανακαλύψαμε ένα καλύτερο under-explanation που δεν εντόπισε ο αλγόριθμος με τα ευριστικά κριτήρια. Βέβαια, γνωρίζαμε ήδη ότι μάλλον αυτόν τον κανόνα τηρούσε ο ταξινομητής από τον σχεδιασμό

Metric	Query	Precision	Recall	Degree	Positives	Negatives
Class 1						
Best Precision	x contains y1, y2, y3. y1 is Large, Object, Cube, Gray. y2 is Large, Cylinder, Object. y3 is Large, Object, Metal.	1.00	0.66	0.66	83	0
Best Recall	x contains y1. y1 is Large, Object, Cube.	0.09	1.00	0.09	125	1216
Best Degree	x contains y1, y2, y3. y1 is Large, Object, Cube, Gray. y2 is Large, Cylinder, Object. y3 is Large, Object, Metal.	1.00	0.66	0.66	83	0
Class 2						
Best Precision	x contains y1, y2, y3, y4, y5. y1 is Small, Object, Sphere. y2 is Large, Object, Rubber. y3 is Small, Metal, Object, Cube. y4 is Small, Object, Brown. y5 is Small, Object, Rubber, Cylinder.	1.00	0.09	0.09	116	0
Best Recall	x contains y1. y1 is Object, Cube.	0.63	1.00	0.63	1247	735
Best Degree	x contains y1, y2. y1 is Metal, Object, Cube. y2 is Small, Metal, Object.	0.78	0.8	0.65	1005	285
Class 3						
Best Precision	x contains y1, y2, y3, y4, y5. y1 is Metal, Blue, Object. y2 is Large, Object, Blue, Sphere. y3 is Yellow, Small, Object, Sphere. y4 is Small, Object, Rubber. y5 is Metal, Object, Sphere.	1.00	0.42	0.42	365	0
Best Recall	x contains y1, y2. y1 is Large, Object. y2 is Object, Sphere.	0.42	1.00	0.42	878	1200
Best Degree	x contains y1, y2. y1 is Yellow, Small, Object, Sphere. y2 is Large, Object, Blue, Sphere.	0.99	0.85	0.85	748	2

(b) ResNet34

Πίνακας 4.3: Οι βέλτιστες εξηγήσεις ως προς τις 3 μετρικές. Δίνεται το query, η τιμή της μετρικής, το πλήθος των positives και το πλήθος των negatives.

του CLEVR-Hans3, όμως αυτό το σενάριο αποτελεί μία ενδιαφέρουσα προσομοίωση ανίχνευσης τέτοιων χαρακτηριστικών σε πραγματικές εφαρμογές.

Είχαμε επισημάνει ότι δύο individuals της κλάσης 3 ταξινομούνται ως δείγματα της κλάσης 1. Χρησιμοποιώντας στοχευμένα το query "x contains y1. y1 is Large, Cube, Object, Gray." είδαμε ότι έχει ως απαντήσεις αυτούς τους δύο individual. Βλέπουμε επομένως ότι και πάλι η παρουσία ενός γκρι κύβου είναι που συγχέει τον ταξινομητή.

Query	Positives	Negatives
x contains y1, y2, y3. y1 is Large, Object, Cube. y2 is Large, Cylinder, Object. y3 is Large, Object, Metal.	108	547
x contains y1, y2. y1 is Large, Object, Cube, Gray. y2 is Large, Cylinder, Object.	93	0

Πίνακας 4.4: Το πλήθος των positives και των negatives που επέστρεψαν δύο τροποποιήσεις του under-explanation της κλάσης 1.

4.2 MNIST

4.2.1 Ταξινομητής

Για το MNIST δουλεύουμε με έναν μόνο ταξινομητή. Έχουμε επιλέξει ένα αρκετά απλούστερο μοντέλο καθώς είναι και αρκετά απλούστερο το σύνολο δεδομένων. Χρησιμοποιούμε ένα CNN αρχιτεκτονικής παρόμοιας με αυτή του AlexNet (Krizhevsky, Sutskever και Hinton 2017) η οποία βρίσκεται στα παραδείγματα της βιβλιοθήκης PyTorch [1]. Επιλέξαμε αυτή την υλοποίηση γιατί είναι απλή αλλά παράγει έναν καλό ταξινομητή καθώς και επειδή η δικιά μας δουλειά δεν εστιάζει στην δομή του ταξινομητή αλλά στην εξήγηση του τρόπου λειτουργίας του. Το δίκτυο εκπαιδεύτηκε για 14 εποχές με την τεχνική του cross-entropy minimization και τον αλγόριθμο βελιστοποίησης Adadelta με τις εξής τιμές παραμέτρων, batch size $= 64$, learning rate $= 1.0, \rho = 0.9, \epsilon = 10^{-6}$, weight decay $= 0$.

Στον πίνακα 4.5 βλέπουμε κάποιες μετρικές για τον ταξινομητή. Ο ταξινομητής έχει υψηλό accuracy και στο σύνολο εκπαίδευσης και στο σύνολο ελέγχου με f1-score 0.99 ή 1.00 για κάθε ψηφίο, ενώ στο confusion matrix βλέπουμε ότι ελάχιστα ψηφία έχει κατηγοριοποιηθεί λανθασμένα, με τα υψηλότερα λάθη να είναι 5 για τα ζεύγη (2,7), (7,2), (9,5). Είναι γεγονός ότι αυτά τα ψηφία έχουν κάποιες σχηματικές ομοιότητες οι οποίες εμφανίζονται και στα παραδείγματα του σχήματος 4.1. Βλέπουμε επίσης πόσες εικόνες από το explanation dataset του MNIST έχει ταξινομήσει σε κάθε ψηφίο.

[1] https://github.com/pytorch/examples/tree/master/mnist

	Accuracy
Training set	99.8%
Test set	99.2%

(a) Η ακρίβεια του ταξινομητή στα σύνολα εκπαίδευσης και ελέγχου.

Digit	Precision	Recall	F1-score
0	0.99	1.00	0.99
1	0.99	1.00	1.00
2	0.99	0.99	0.99
3	0.99	1.00	0.99
4	0.99	0.99	0.99
5	0.99	0.99	0.99
6	1.00	0.99	0.99
7	0.99	0.99	0.99
8	0.99	0.99	0.99
9	0.99	0.98	0.99

(b) Οι μετρικές precision, recall και F1-score για τα 10 ψηφία στο σύνολο ελέγχου.

	Predicted label									
True label	0	1	2	3	4	5	6	7	8	9
0	978	0	0	0	0	0	1	1	0	0
1	0	1133	1	1	0	0	0	0	0	0
2	1	2	1023	0	0	0	1	5	0	0
3	0	0	1	1005	0	3	0	0	1	0
4	1	0	0	0	975	0	1	0	2	3
5	2	0	0	3	0	886	1	0	0	0
6	4	2	0	1	1	3	946	0	1	0
7	0	2	5	1	0	0	0	1018	1	1
8	1	1	1	1	0	0	0	1	966	3
9	1	1	0	0	4	5	0	3	2	993

(c) Ο πίνακας σύγχυσης για το MNIST.

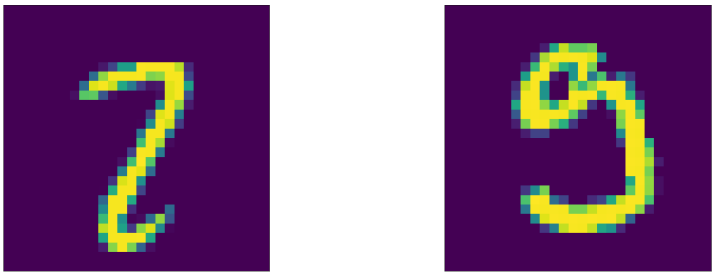
	Predicted label									
	0	1	2	3	4	5	6	7	8	9
Count	31	30	25	26	22	31	18	24	20	15

(d) Το πλήθος των εικόνων του explanation dataset που ταξινομήθηκαν σε κάθε ψηφίο.

Πίνακας 4.5: Οι μετρικές του ταξινομητή του MNIST.

4.2.2 Παραγωγή εξηγήσεων

Πρώτα παράγονται εξηγήσεις με την χρήση του αλγορίθμου Explain και στην συνέχεια με τον αλγόριθμο Explain2 καθώς, όπως θα δούμε στην αξιολόγηση, τα πρώτα αποτελέσματα δεν εί-



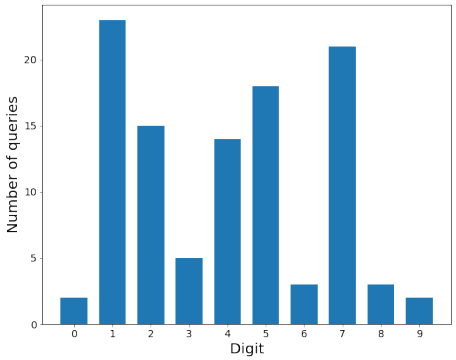
Σχήμα 4.1: Ένα 2 που έχει ταξινομηθεί ως 7 και ένα 9 που έχει ταξινομηθεί ως 5.

	Explain			Explain2		
Digit	Query Creation	Query Answering	Total	Query Creation	Query Answering	Total
0	0.64s	0.82s	1.46s	24m53s	20s	25m13s
1	0.25s	4.0s	4.25s	30s	7s	37s
2	0.72s	122s	123s	6m33s	14s	6m47s
3	0.68s	0.69s	1.37s	22m39s	11s	22m50s
4	0.43s	2.4s	2.83s	19s	10s	29s
5	1.2s	8.5s	9.7s	21m15s	41s	21m56s
6	0.35s	74s	74s	5m49s	5s	5m54s
7	0.69s	27s	28s	25s	12s	37s
8	0.72s	0.15s	0.87s	16m32s	70s	17m42s
9	0.50s	1.8s	2.3s	1m55s	4s	1m59s

Πίνακας 4.6: Οι χρόνοι υπολογισμού των queries με τους αλγορίθμους Explain και Explain2 και της εύρεσης των απαντήσεων με το GraphDB για το MNIST.

ναι ικανοποιητικά. Τα certain answers των query υπολογίζονται με την χρήση του προγράμματος GraphDB. Στον πίνακα 4.6 παρουσιάζουμε τους χρόνους υπολογισμού. Οι χρόνοι υπολογισμού για τον αλγόριθμο Explain είναι πολύ μικρότεροι σε σχέση με το CLEVR-Hans3 καθώς έχουμε πολύ λιγότερα δείγματα σε κάθε κλάση. Για τον αλγόριθμο Explain η παράμετρος t είχε τεθεί ίση με 20 και επηρέασε αρκετά τα αποτελέσματα. Λόγω του μεγάλου πλήθους των γραμμών που έχει η περιγραφή κάποιων ψηφίων στην οντολογία, όπως το 8, το μέγεθος των query για αυτά αυξήθηκε γρήγορα και η μέθοδος Minimize δεν τα ελαχιστοποίησε επαρκώς. Στο σχήμα 4.2 βλέπουμε το πλήθος των queries που επέστρεψε ο αλγόριθμος Explain για κάθε ψηφίο. Ο χρόνος υπολογισμού των απαντήσεων ποικίλει πολύ, εν μέρει λόγω των διαφορετικών πληθών των queries αλλά κυρίως λόγω του πλήθους των μεταβλητών που έχουν τα queries. Τα ψηφία 2 και 6 που έχουν μακράν τους μεγαλύτερους χρόνους περιέχουν queries με 20 μεταβλητές και όπως έχουμε αναφέρει ο υπολογισμός των απαντήσεων είναι εκθετικός ως προς το πλήθος των μεταβλητών.

Ο αλγόριθμος Explain2 έχει τους μεγαλύτερους χρόνους υπολογισμούς των queries με μεγάλη διαφορά και αυτό οφείλεται στην brute force υλοποίηση του αλγορίθμου. Η παράμετρος t όμως, που τέθηκε ίση με 200.000, ορίζει ένα όριο και περιόρισε τον χρόνο υπολογισμού των queries για τα ψηφία που έχουν αναπαραστάσεις με έως και 13 γραμμές. Ο χρόνος απάντησης των queries είναι πολύ καλός δεδομένου το πολύ μεγαλύτερο πλήθος των queries που παράγει ο αλγόριθμος. Αυτό οφείλεται στο μικρό πλήθος των μεταβλητών που έχουν τα queries που παράγει.



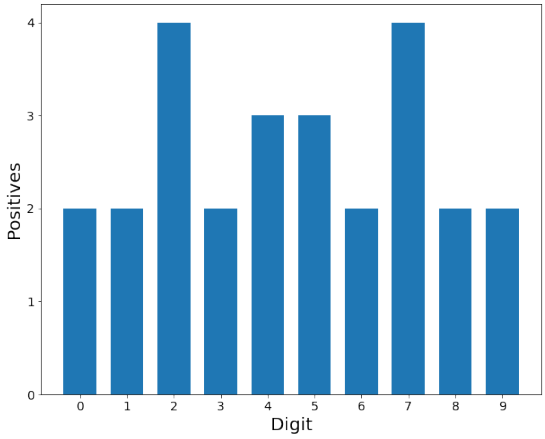
Σχήμα 4.2: Το πλήθος των query που παρήγαγε ο αλγόριθμος Explain για κάθε ψηφίο.

4.2.3 Αξιολόγηση

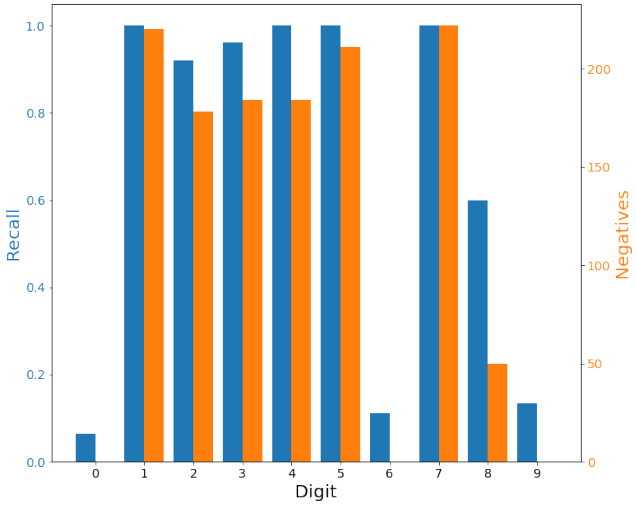
Στο σχήμα 4.3 βλέπουμε πληροφορίες για τα queries του αλγορίθμου Explain που μεγιστοποιούν τις μετρικές precision, recall, degree για κάθε ψηφίο. Επειδή κάθε ψηφίο είχε κάποιο query που να έχει precision 1.00, δηλαδή under-explanation, παρουσιάζουμε τον μέγιστο αριθμό positives μεταξύ των under-explanati-ons. Σε γενικές γραμμές κανένα ψηφίο δεν επιστρέφει ικανοποιητικό αριθμό positives, με τα περισσότερα ψηφία να επιστρέφουν μόλις 2. Πέρα από τη τιμή του μέγιστου recall παρουσιάζουμε και το αντίστοιχο πλήθος των negatives που επέστρεψε το κάθε query. Έχοντας υπ' όψιν μας ότι κάθε κλάση περιέχει περίπου 25 individuals ο αριθμός των negatives είναι δραματικά μεγάλος για τα περισσότερα ψηφία, με αρκετά να προσεγγίζουν τους 250 που είναι και το μέγεθος ολόκληρου του explanation dataset. Ακόμα και τα queries που έχουν υποσχόμενες μετρικές δεν αποτελούν ικανοποιητικές εξηγήσεις λόγω του πλήθους των μεταβλητών τους και την διφορούμενη ερμηνεία των μεταβλητών οι οποίες επιτρέπεται να ενοποιηθούν. Παραδείγματος χάρην το εξής είναι ένα απόσπσμα από το query που είναι το καλύτερο ως προς το precision για το ψηφίο 2: "y1 is BotLeft. y2 is Short. y3 is MidCenter. y4 is MidRight. y5 is BotLeft. y6 is MidCenter. y7 is MidRight. y9 is TopCenter, MidCenter. y10 is Medium, MidCenter. y11 is MidCenter. y12 is BotCenter. y13 is MidCenter. y14 is MidCenter. y15 is Line45deg. y17 is BotCenter. y18 is BotCenter. y19 is Line90deg, Short. y20 is BotCenter. x contains y1. x contains y2. x contains y3. x contains y4. x contains y5. x contains y6. x contains y7. x contains y8. x contains y9. x contains y10. x contains y11...".

Στο σχήμα 4.4 βλέπουμε τις αντίστοιχες μετρικές για τα queries του αλγορίθμου Explain2. Βλέπουμε πολύ σημαντική αύξηση στο πλήθος των positives στα under-explanation και στο μέγιστο degree. Βλέπουμε επίσης ότι πλέον κάθε ψηφίο έχει over-explanations αν και σε πολλά ο αριθμός των negatives παραμένει υψηλός.

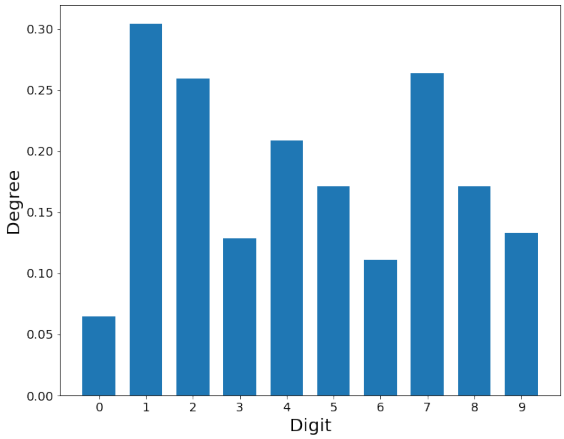
Σε μία πιο ποιοτική αξιολόγηση των αποτελεσμάτων, έχουμε συγκεντρώσει κάποια ενδιαφέρονται under-explanations που μας δείχνουν πρότυπα παραδείγματα για κάθε ψηφίο. Στο σχήμα 4.5 βλέπουμε μία σχηματική απεικόνιση του ψηφίου που περιγράφει το query "y1 is BotLeft, BotCenter. y2 is TopCenter, MidLeft, Line45deg, MidCenter. y3 is MidRight, Line45deg, BotCenter. y4 is Line90deg, MidRight. y5 is Line90deg. y6 is TopRight, Line135deg. y1 intersects y5. y2 intersects y5. y3 intersects y4. y4 intersects y6." και κάποια παραδείγματα από το explanation dataset που αποτελούν απαντήσεις. Στα σχήματα 4.6 και 4.7 βλέπουμε τα query "y1 is Line0deg, Medium, MidCenter. y2 is Line45deg, MidCenter. y3 is Line90deg, BotCenter, Long, MidCenter, TopCenter. y1 intersects y2. y1 intersects y3." και "y1 is Line0deg, TopLeft, Long, TopCenter, MidCenter. y2 is Line0deg, MidCenter. y3 is MidRight, Long, MidCenter, BotCenter. y2 intersects y3." αντίστοιχα. Είναι αξιοσημείωτο το τελευταίο 7 του σχήματος 4.7 φαίνεται να λείπει μία γραμμή από το πρότυπο παράδειγμα. Αυτό που συμβαίνει όμως, είναι ότι η οριζόντια γραμμή βρίσκεται στο κατάλληλο ύψος ώστε και οι ματαβλητή "y1" και η "y2" να μπορούν να αντιστοιχιστούν σε αυτήν και επομένως αναδεικνύεται μία



(a) Το μέγιστο πλήθος positives μεταξύ των queries που έχουν precision 1.00 για κάθε ψηφίο.

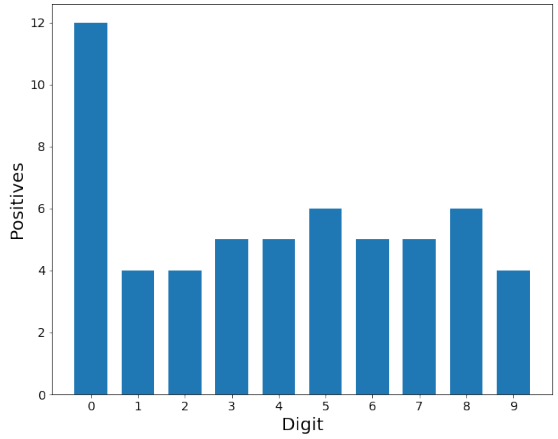


(b) Το μέγιστο recall για κάθε ψηφίο σε σύγκριση με τον αριθμό των negatives που επιστρέφει το αντίστοιχο query.

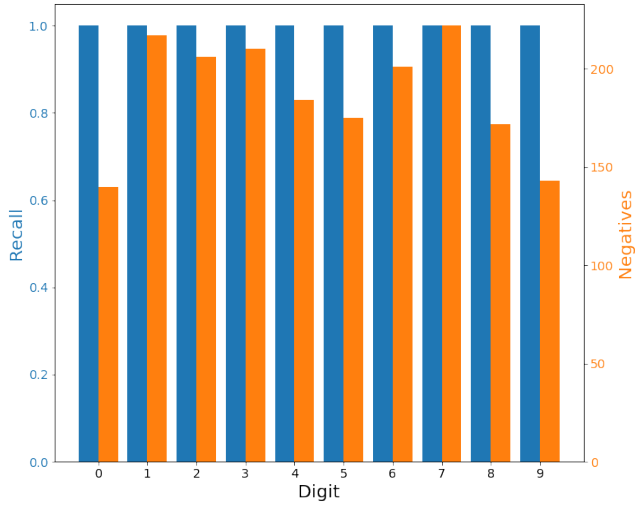


(c) Το μέγιστο degree για κάθε ψηφίο.

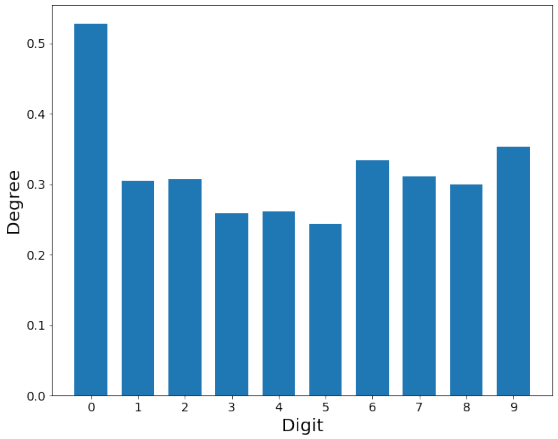
Σχήμα 4.3: Πληροφορίες για τα βέλτιστα query που παρήγαγε ο αλγόριθμος Explain ως προς την εκάστοτε μετρική για κάθε ψηφίο.



(a) Το μέγιστο πλήθος positives μεταξύ των queries που έχουν precision 1.00 για κάθε ψηφίο.



(b) Το μέγιστο recall για κάθε ψηφίο σε σύγκριση με τον αριθμό των negatives που επιστρέφει το αντίστοιχο query.



(c) Το μέγιστο degree για κάθε ψηφίο.

Σχήμα 4.4: Πληροφορίες για τα βέλτιστα query που παρήγαγε ο αλγόριθμος Explain2 ως προς την εκάστοτε μετρική για κάθε ψηφίο.

αδυναμία στην εκφραστικότητα των queries που χρησιμοποιούμε.

Σχήμα 4.5: Πρότυπο παράδειγμα και αντίστοιχα δείγματα του συνόλου εξήγησης για το ψηφίο 0.

Σχήμα 4.6: Πρότυπο παράδειγμα και αντίστοιχα δείγματα του συνόλου εξήγησης για το ψηφίο 4.

Σχήμα 4.7: Πρότυπο παράδειγμα και αντίστοιχα δείγματα του συνόλου εξήγησης για το ψηφίο 7.

Ένα ακόμα αδύναμο σημείο αναδεικνύεται από τα ψηφία του σχήματος 4.8. Ψηφία όπως το 1 στα αριστερά χαρακτηρίζονται κυρίως από την απουσία γραμμών και όχι από την παρουσία. Στην εκφραστικότητα που χρησιμοποιούμε εμείς είναι πολλές φορές αδύνατον να τα περιγράψουμε χωρίς να συμπεριλάβουμε άλλα ανεπιθύμητα ψηφία. Αντίστοιχα στην δεξιά εικόνα βλέπουμε ένα 3 που ταξινομήθηκε ως 5. Αυτό εικάζουμε ότι έγινε και πάλι λόγω της απουσίας γραμμών που έχουν τα 3 στην κορυφή τους και επομένως είναι αδύνατον να περιγράψουμε τι ξεχωρίζει αυτό το 3 από τα υπόλοιπα, αυτή η διατύπωση μπορεί να γίνει μόνο από την αντίθετη πλευρά.

Σχήμα 4.8: Δύο παραδείγματα ψηφίων στα οποία η απουσία γραμμών έχει κρίσιμο ρόλο.

Κεφάλαιο 5

Επίλογος

5.1 Σύνοψη

Στην παρούσα διπλωματική εργασία, διερευνήσαμε την χρήση των Συζευκτικών Ερωτημάτων στην εξήγηση ταξινομητών που αντιμετωπίζονται ως μαύρα κουτιά. Μελετήσαμε κάποιες θεωρητικές ιδιότητες που έχουν τα Συζευκτικά Ερωτήματα ως αναφορά τις βέβαιες απαντήσεις τους και ιδιαίτερα η έννοια της Υπαγωγής. Είδαμε την έννοια του Ελάχιστου Κοινού Υπαγωγού, την χρησιμότητα του για την ενοποίηση δύο Συζευκτικών Ερωτημάτων και πως μπορούμε να τον υπολογίσουμε. Μελετήσαμε και μία εναλλακτική μέθοδο ενοποίησης Συζευκτικών Ερωτημάτων, την εύρεση του βέλτιστου ταριάσματος, που αν και χαλαρότερη και περισσότερο κοστοβόρα, δημιουργεί Συζευκτικά Ερωτήματα με λιγότερες μεταβλητές, τα οποία αναμένεται να είναι πιο ευανάγνωστα. Έπειτα κατασκευάσαμε και αλγορίθμους για την παραγωγή εξηγήσεων στη μορφή Συζευκτικών Ερωτημάτων, αξιοποιώντας αυτές τις τεχνικές ενοποίησης. Στη συνέχεια, ορίσαμε την έννοια του Συνόλου Εξήγησης, ένα σύνολο δεδομένων που είναι εμπλουτισμένο με υψηλού επιπέδου πληροφορία, εκφρασμένη σε κάποια Περιγραφική Λογική. Μελετήσαμε την κατασκευή του για δύο σύνολα δεδομένων, το CLEVR-Hans3 και το MNIST, βλέποντας πώς μπορούμε να το κατασκευάσουμε χρησιμοποιώντας ήδη υπάρχουσα πληροφορία ή εξάγοντας υψηλού επιπέδου χαρακτηριστικά αξιοποιώντας παραδοσιακές τεχνικές. Τέλος, εφαρμόσαμε τους αλγορίθμους μας στα Σύνολα Εξήγησης που κατασκευάσαμε και λάβαμε εξηγήσεις για ταξινομητές, τις οποίες αξιολογήσαμε με ποσοτικά και ποιοτικά κριτήρια.

5.2 Τελικά συμπεράσματα

Μέσα από τα πειράματα μας, είδαμε ότι ο Ελάχιστος Κοινός Υπαγωγός είναι ένα δυνατό εργαλείο για την παραγωγή εξηγήσεων, όταν όμως μπορεί να αντιμετωπιστεί το ζήτημα της αύξησης των μεταβλητών. Ως εναλλακτική, το βέλτιστο ταίριασμα μπορεί να υποκαταστήσει τον Ελάχιστο Κοινό Υπαγωγό παράγοντας ικανοποιητικά αποτελέσματα. Συναντήσαμε βέβαια και κάποιες αδυναμίες στην χρήση των Συζευκτικών Ερωτημάτων που σχετίζονται με την εκφραστική τους ικανότητα, οι οποίες μπορούν να καταστήσουν δύσκολη ή αδύνατη την παραγωγή ικανοποιητικών εξηγήσεων σε ορισμένες περιπτώσεις.

5.3 Μελλοντικές κατευθύνσεις

Το πλαίσιο εργασίας μας αφήνει κατευθύνσεις προς εξερεύνηση σε κάθε του στάδιο. Μελλοντικές εργασίες μπορούν να πειραματιστούν με διαφορετικές εκφραστικότητες στις Περιγραφικές Λογικές, που αν και πιο ισχυρές δημιουργούν εμπόδια στην εύρεση αποδοτικών αλγορίθμων. Πέρα από τις Περιγραφικές Λογικές, αφήνεται ανοιχτός ο πειραματισμός στην εκφραστικότητα των Συζευκτικών Ερωτημάτων η οποία ενδεχομένως να μπορεί να εμπλουτιστεί ανεξάρτητα, χωρίς να επιβαρυνθεί η πολυπλοκότητα των αλγορίθμων. Όσον αφορά την ενοποίηση των Συζευκτικών Ερωτημάτων, μπορούν να εξερευνηθούν τεχνικές πέρα από τον Ελάχιστο Κοινό Υπαγωγό και το βέλτιστο ταίριασμα. Το βέλτιστο ταίριασμα μπορεί να μελετηθεί και με πιο λεπτές μεθόδους, καθώς

εμείς αρκεστήκαμε σε μία απλοϊκή εξαντλητική μέθοδο. Το πρόβλημα ανήκει στην κατηγορία των προβλημάτων εύρεσης μέγιστου κοινού υπογράφου και επομένως υπάρχει αρκετή βιβλιογραφία ως σημείο αφετηρίας. Όμως, και οι τεχνικές που μελετήσαμε μπορούν να αποτελέσουν αντικείμενο περαιτέρω μελέτης για την εφαρμογή τους σε αλγορίθμους παραγωγής εξηγήσεων πέρα από αυτούς που κατασκευάσαμε εμείς. Τέλος, μπορούν να μελετηθούν διαφορετικές σχεδιαστικές επιλογές στην εξαγωγή περιγραφών για τα Σύνολα Εξήγησης που κατασκευάσαμε, ειδικά σε αυτό του MNIST.

Βιβλιογραφία

Baader, Franz κ.ά. (2017). *An Introduction to Description Logic*. Cambridge University Press. DOI: 10.1017/9781139025355.

Bodirsky, Manuel (2021). *Graph Homomorphisms and Universal Algebra, Course Notes*.

Deng, J. κ.ά. (2009). «ImageNet: A Large-Scale Hierarchical Image Database». Στο: *CVPR09*.

Goodfellow, Ian J., Yoshua Bengio και Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press.

Gottlob, Georg και Christian G. Fermüller (1993). *Removing redundancy from a clause*.

He, Kaiming κ.ά. (2015). «Deep Residual Learning for Image Recognition». Στο: *CoRR* abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

Johnson, Justin κ.ά. (2016). *CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning*. arXiv: 1612.06890.

Kingma, Diederik P. και Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].

Klare, Brendan F. κ.ά. (2012). «Face Recognition Performance: Role of Demographic Information». Στο: *IEEE Transactions on Information Forensics and Security* 7.6, σσ. 1789–1801. DOI: 10.1109/TIFS.2012.2214212.

Kontchakov, Roman και Michael Zakharyaschev (2014). «An Introduction to Description Logics and Query Rewriting». Στο: *Reasoning Web. Reasoning on the Web in the Big Data Era: 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*. Επιμέλεια υπό Manolis Koubarakis κ.ά. Cham: Springer International Publishing, σσ. 195–244. ISBN: 978-3-319-10587-1. DOI: 10.1007/978-3-319-10587-1_5. URL: https://doi.org/10.1007/978-3-319-10587-1_5.

Krishna, Ranjay κ.ά. (2016). «Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations». Στο: URL: https://arxiv.org/abs/1602.07332.

Krizhevsky, Alex, Ilya Sutskever και Geoffrey E. Hinton (Μάι. 2017). «ImageNet Classification with Deep Convolutional Neural Networks». Στο: *Commun. ACM* 60.6, σσ. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

LeCun, Yann κ.ά. (Νοέ. 1998). «Gradient-Based Learning Applied to Document Recognition». Στο: *Proceedings of the IEEE, 86(11):2278-2324*.

Lindeberg, Tony (2008). «Scale-Space». Στο: *Wiley Encyclopedia of Computer Science and Engineering*. American Cancer Society, σσ. 2495–2504. ISBN: 9780470050118. DOI: https://doi.org/10.1002/9780470050118.ecse609.

Miller, George κ.ά. (Ιαν. 1991). «Introduction to WordNet: An On-line Lexical Database*». Στο: 3. DOI: 10.1093/ijl/3.4.235.

Rubinstein, Reuven Y. και Dirk P. Kroese (2004). *The Cross-Entropy Method*. Springer New York. DOI: 10.1007/978-1-4757-4321-0. URL: https://doi.org/10.1007%2F978-1-4757-4321-0.

Stammer, Wolfgang, Patrick Schramowski και Kristian Kersting (2021). *Right for the Right Concept: Revising Neuro-Symbolic Concepts by Interacting with their Explanations*. arXiv: 2011.12854.

Zeiler, Matthew D και Rob Fergus (2013). *Visualizing and Understanding Convolutional Networks*. arXiv: 1311.2901.

Στάμου, Γεώργιος (2015). *Αναπαράσταση οντολογικής γνώσης και συλλογιστική*.