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

Αξιολόγηση και Ερμηνεία Συστημάτων Μηχανικής Μάθησης Βασισμένη σε Γράφους Γνώσης

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΕΝΤΜΟΝΤ-ΓΡΗΓΟΡΗΣ Γ. ΝΤΕΡΒΑΚΟΣ

Επιβλέπων : Γιώργος Στάμου
Καθηγητής ΕΜΠ

Αθήνα, Νοέμβριος 2023

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

Αξιολόγηση και ερμηνεία συστημάτων μηχανικής μάθησης βασισμένη σε γράφους γνώσης

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Έντμοντ-Γρηγόρης Ντερβάκος

Συμβουλευτική Επιτροπή : Γεώργιος Στάμου
Αθανάσιος Βουλόδημος
Στέφανος Κόλλιας

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή τη 8η Νοεμβρίου 2023.

.....	.....	.....
Γ. Στάμου	Α. Βουλόδημος	Σ. Κόλλιας
Καθηγητής ΕΜΠ	Επ. Καθηγητής ΕΜΠ	Ομ. Καθηγητής ΕΜΠ
.....	.....	.....
Κ. Νικήτα	Μ. Βαζιργιάννης	Δ. Φωτάκης
Καθηγήτρια ΕΜΠ	Καθηγητής Ecole Polytechnique	Καθηγητής ΕΜΠ
	.....	
	Α. Ροντογιάννης	
	Αν. Καθηγητής ΕΜΠ	

Αθήνα, Νοέμβριος 2023

Έντμοντ-Γρηγόρης Γ. Ντερβάκος
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η τεχνητή νοημοσύνη υπέστη εκρηκτική εξέλιξη τα τελευταία χρόνια. Με κινητήριο δύναμη την τεχνολογία της βαθιάς μάθησης, η τεχνητή νοημοσύνη βρίσκει εφαρμογή σε πληθώρα επιστημονικών πεδίων, στη βιομηχανία, καθως και στις τέχνες. Παρά τα θεαματικά αποτελέσματα, έχουν προκύψει διάφορα ηθικά ζητήματα που εμποδίζουν την αξιοποίηση της βαθιάς μάθησης σε εφαρμογές που επηρεάζουν κρίσιμα τη ζωή των ανθρώπων, όπως είναι οι εφαρμογές στην ιατρική. Κύρια πηγή των ηθικών ζητημάτων αποτελεί η αδιαφάνεια της βαθιάς μάθησης, καθώς τα μοντέλα αδυνατούν να δώσουν επεξηγήσεις για τις αποφάσεις που λαμβάνουν, και η εφαρμογή τους προϋποθέτει την εμπιστοσύνη των χρηστών. Αυτή η έλλειψη εξηγισιμότητας γεννά περεταίρω προβλήματα πέραν των ηθικών, όπως είναι για παράδειγμα η δυσκολία εντοπισμού και διόρθωσης σφαλμάτων σε τέτοια συστήματα.

Τα ζητήματα αυτά έχουν οδηγήσει στο πεδίο έρευνας της ερμηνεύσιμης τεχνητής νοημοσύνης, στο οποίο εντάσσεται η παρούσα διατριβή. Το πεδίο αυτό έχει μεγάλο εύρος, με πλήθος διαφορετικών αλγορίθμων που παράγουν διαφορετικού τύπου επεξηγήσεις, σε διαφορετικά θεωρητικά πλαίσια και διαφορετικούς τύπους δεδομένων. Σε αυτή τη διατριβή διερευνήσαμε τα συστήματα και τις τεχνολογίες τυπικής αναπαράστασης γνώσης ως εργαλείο για την επεξήγηση, καθώς και για την αξιολόγηση της λειτουργίας αδιαφανών συστημάτων βαθιάς μάθησης. Συγκεκριμένα, αναπτύχθηκε το κατάλληλο θεωρητικό πλαίσιο και αλγόριθμοι για την επεξήγηση τέτοιων συστημάτων με βάση σημασιολογικές περιγραφές δεδομένων, χρησιμοποιώντας συγκεκριμένη ορολογία που περιγράφεται σε υποκείμενη οντολογική γνώση.

Το θεωρητικό πλαίσιο, ως ανεξάρτητο του τύπου δεδομένων, και του υποκείμενου μοντέλου, εφαρμόστηκε στα πεδία της εικόνας, των συμβολικών αναπαραστάσεων μουσικής, και του ήχου. Συγκρίθηκε με υπαρχουσες μεθόδους επεξήγησης και αξιολόγησης αδιαφανών συστημάτων, και αναδείχθηκε ως προσέγγιση που δύναται να προσφέρει στο χρήστη πληροφορία που άλλες μέθοδοι αδυνατούν, χάρη στην θεμελίωση των προτεινόμενων εξηγήσεων σε τυπικά αναπαραστημένη γνώση. Η καινοφανής ιδέα μας να χρησιμοποιήσουμε γράφους γνώσης με τον συγκεκριμένο τρόπο, ανοίγει νέα ερευνητικά μονοπάτια προς την υβριδοποίηση των συστημάτων τεχνητής νοημοσύνης, αξιοποιώντας πληροφορία χαμηλού επιπέδου, όπου διαπρέπει η αδιαφανείς βαθιά μάθηση, καθώς και συμβολική πληροφορία υψηλού επιπέδου, η οποία είναι οργανωμένη, δομημένη, και πιο κατανοητή στον άνθρωπο.

Λέξεις Κλειδιά: Γράφοι Γνώσης, Εξηγισιμότητα, Ερμηνευσιμότητα, Αξιολόγηση

Abstract

Artificial intelligence (AI) has progressed explosively in recent years. Driven by the advent of deep learning, AI is being used in a variety of applications, across multiple scientific fields, in industry as well as in the arts. Despite spectacular results, various ethical issues have arisen that prevent the utilization of deep learning in applications that critically affect people's lives, such as applications in medicine. The main source of ethical issues is the opacity of deep learning, as the models generally do not provide explanations for the decisions they make, and their use presupposes users' trust. This lack of transparency also gives rise to additional problems hindering the development of AI systems, such as for example the difficulty of detecting, and consequently fixing bugs, and mistakes in deep learning based systems.

These issues have led to the emergence of the eXplainable AI (XAI) field of research, which is the overarching context of this dissertation. This field of research has produced a wide range of approaches, with different algorithms that produce different types of explanations, in different theoretical contexts and concerning different types of data. In this dissertation we explored systems and technologies of formal knowledge representation as a tool to explain the operation of opaque deep learning systems. Specifically, we developed a theoretical framework and algorithms for explaining such systems based on semantic descriptions of data, expressed using specific terminology which is described in underlying ontological knowledge.

The proposed framework is domain and model agnostic, and was applied on image classification, symbolic music generation and classification, and audio classification systems. It was compared with existing explainability and evaluation methods, and emerged as a promising approach that can provide high level information to users that other approaches cannot, thanks to the grounding of the explanations on structured represented knowledge. Our novel idea to utilize knowledge graphs for explainability in this way opens new paths to researching hybrid AI systems that utilize both low level sub-symbolic information, such as deep learning systems, in addition to high level symbolic information, that is structured, and more understandable to humans, as are knowledge graphs.

Keywords: Knowledge Graphs, Explainability, Interpretability, Evaluation

Acknowledgements

The work presented in this thesis is the result of a 5-year effort done at the AILS lab, under the supervision of professor Giorgos Stamou. It was an interesting 5-year period, that included a 'pre-COVID' era, quarantines, and 'post-COVID'. Especially during the pandemic, there are not enough words to describe how important it was to collaborate with my colleagues, and a huge thanks is owed to them for helping, and supporting me, and eachother, and to professor Giorgos Stamou, who managed to organize the "virtual version" of the lab in a way that worked for everyone.

The work that is presented in this thesis was mostly a collaborative effort. I would like to thank my co-authors, and acknowledge their contributions to this work. Specifically, Giorgos Filandrianos, and Konstantinos Thomas for coadapting the proposed framework to generating counterfactual explanations, Orfeas Menis-Mastromichalakis, Jason Liartis, and Alexandros Chortaras, for coadapting the framework to generating rule-based explanations, and for developing the KGrules and KGrules-H algorithms, Spyridon Kantarelis and Natalia Kotsani for collaborating and experimenting in the domain of symbolic music, Theofanis Ganitidis, professor Konstantia Zarkogianni, and professor Konastantina Nikita, my colleagues at the smarty4covid project, Maria Lymperaiou, George Manoliadis, Vassilis Lymperatos, and professor Chryssoula Zerva for collaborating outside the scope of this dissertation and teaching me valuable experience, and ofcourse my supervisor professor Giorgos Stamou for his excellent leadership and motivating presence.

Finally, none of this would have been possible without my parents, George and Miranda, my brother Ryan, and my best friends and bandmates Ilias Samartzis and Ian Stratis.

Contents

List of Tables

List of Figures

Glossary - Γλωσσάριο

***ante hoc* explanation**	Εκ των προτέρων εξήγηση
Black box explanation	Εξήγηση μαύρου κουτιού
Certain answers	Βέβαιες απαντήσεις (σημασιολογικού ερωτήματος)
Classifier	Ταξινομητής
Concept attribution	Εξηγήσεις ανάθεσης εννοιών
Convolutional Neural Network	Συνελικτικό Νευρωνικό Δίκτυο
Counterfactual Explanation	Αντιπαραθετική Επεξήγηση
Description Logics	Περιγραφικές Λογικές
Explainability	Εξηγησιμότητα
Explanation Dataset	Σύνολο Δεδομένων Εξήγησης
Feature Importance	Εξηγήσεις Σημαντικότητας Χαρακτηριστικών
Feature Space	Χώρος Χαρακτηριστικών
Global Explanation	Εξήγηση Μοντέλου σε Σύνολο Δεδομένων
Heuristic	Ευριστική
Informativeness	Πληροφορία (που φέρουν εξηγήσεις)
Inherently Interpretable	Εγγενώς Ερμηνεύσιμο
Interpretability	Ερμηνευσιμότητα
Knowledge Base	Βάση Γνώσης
Knowledge Graph	Γράφος Γνώσης
Level of Abstraction	Επίπεδο Αφαίρεσης
Local Explanation	Εξήγηση Μοντέλου σε Ένα Δεδομένο
Model-agnostic Explanation Method	Μέθοδος εξήγησης ανεξάρτητη από το μοντέλο
Model-specific Explanation Method	Μέθοδος εξήγησης για συγκεκριμένη οικογένεια μοντέλων
Opaque AI	Αδιαφανής Τεχνητή Νοημοσύνη
***post hoc* Explanation**	Εκ των υστέρων Εξήγηση
Pianoroll	Πίνακας Αναπαράστασης σε ταινία πλήκτρων πιάνο
Query Reverse Engineering	Αντίστροφος Σχεδιασμός Ερωτημάτων
Receptive Field	Δεκτικό Πεδίο
Rule Based Explanation	Εξηγήσεις σε Μορφή Κανόνων
Semantic Description	Σημασιολογική Περιγραφή
Semantic Explanation	Σημασιολογικές Εξηγήσεις
Semantic Query	Σημασιολογικό Ερώτημα
Understandability	Κατανοητότητα
White box Explanation	Εξήγηση άσπρου κουτιού (με πρόσβαση στο μοντέλο)

Εκτεταμένη Περίληψη

Η εξηγισιμότητα, η αξιολόγηση, και η ερμηνεία των συστημάτων τεχνητής νοημοσύνης, και ειδικότερα εκείνων της μηχανικής μάθησης, έχουν προκύψει ως ζητήματα καθοριστικής σημασίας για την εφαρμογή τους σε πραγματικά προβλήματα. Οι μέθοδοι για την εποπτεία των συστημάτων μπορούν να συνεισφέρουν σημαντικά στη βελτίωση της απόδοσής τους, καθώς διευκολύνουν την αναγνώριση και την ανάλυση σφαλμάτων, και μπορούν να κατευθύνουν αυτούς που τα αναπτύσσουν ώστε να τα αντιμετωπίσουν. Επίσης, ειδικότερα η εξηγισιμότητα θα είναι απαραίτητη για την εφαρμογή κανονισμών και νομοθετικών πλαισίων που αφορούν τις εφαρμογές της τεχνητής νοημοσύνης στη βιομηχανία. Για παράδειγμα, πρέπει να διασφαλίσουμε πως τα συστήματα με τα οποία αλληλεπιδρά η κοινωνία, δεν μεταδίδουν επιβλαβή στερεότυπα, αλλά για να διασφαλιστεί αυτό, θα πρέπει πρώτα να αναπτύξουμε στιβαρές μεθόδους για την αναγνώρισή τους. Επιπρόσθετα, η εξηγισιμότητα δύναται να διευκολύνει την αλληλεπίδραση του ανθρώπου με την τεχνητή νοημοσύνη, για τον απλό λόγο οτι ένας χρήστης θα μπορεί να ρωτάει το σύστημα "γιατί;" και το σύστημα να του απαντά με ικανοποιητικό τρόπο.

Η αναδυόμενη ανάγκη για την εξηγισιμότητα της τεχνητής νοημοσύνης έχει οδηγήσει στην ανάπτυξη του ερευνητικού τομέα της *ερμηνεύσιμης τεχνητής νοημοσύνης* (XAI - eXplainable AI). Ο τομέας αυτός είναι ρευστός, προκύπτουν συνεχώς νέες ιδέες, αλλάζουν τα δεδομένα, και οι ερευνητές δεν συμφωνούν ακόμα και σε στοιχειώδη ζητήματα, όπως είναι οι ορισμοί των όρων εξηγισιμότητα και ερμηνευσιμότητα, ή το ποιες είναι οι απαιτήσεις για μια "καλή" εξήγηση. Σε διάφορες επισκοπήσεις της περιοχής, έχουν οριστεί ταξονομίες μεθόδων, στις οποίες σε γενικές γραμμές συμφωνεί η ερευνητική κοινότητα. Μία από τις σημαντικές διακρίσεις μεθόδων αφορά το αν αυτές είναι *ante hoc*, όπου η εξηγισιμότητα λαμβάνεται υπ'όψη κατά την ανάπτυξη και εκπαίδευση ενός μοντέλου ή *post hoc*, όπου επιχειρείται η εξήγηση ενός συστήματος αφού αυτό έχει αναπτυχθεί και εκπαιδευτεί. Όμοια, υπάρχει διάκριση μεθόδων σε αυτές του "μαύρου κουτιού" (black box), οι οποίες εφαρμόζονται σε οποιοδήποτε σύστημα, αρκεί να υπάρχει πρόσβαση σε ένα σύνολο εισόδων και εξόδων, και στις μεθόδους του "άσπρου κουτιού" (white box) όπου για την εξήγηση απαιτείται πρόσβαση στο εσωτερικό του μοντέλου (για παράδειγμα στα βάρη ενός νευρωνικού δικτύου).

Στα πλαίσια της διατριβής, μας απασχολούν κυρίως οι *post hoc* μέθοδοι "μαύρου κουτιού", καθώς αυτές έχουν το μεγαλύτερο εύρος εφαρμογής, και έχουν τις λιγότερες απαιτήσεις για να μπορέσουν να λειτουργήσουν στην πράξη. Αξίζει να σημειωθεί πως αυτές οι μέθοδοι έχουν υποστεί σημαντική κριτική από κάποιους ερευνητές της ερμηνεύσιμης τεχνητής νοημοσύνης. Συγκεκριμένα, επιχειρηματολογούν πως για σημαντικά προβλήματα στα οποία η εξηγισιμότητα είναι απαραίτητη, όπως είναι τα προβλήματα στον τομέα της ιατρικής, θα πρέπει να αναπτύσσουμε λύσεις οι οποίες να είναι εν γενώς εξηγίσιμες, αξιοποιώντας μεθόδους εξαγωγής χαρακτηριστικών, και εκπαιδεύοντας απλά και κατανοητά συστήματα μηχανικής μάθησης, όπως είναι τα δέντρα αποφάσεων. Ένα άλλο επιχείρημα είναι πως αν καταφέρουμε να εξηγήσουμε με ακρίβεια ένα μαύρο κουτί με *post hoc* προσέγγιση, τότε γιατί να μην χρησιμοποιήσουμε απευθίας την μέθοδο εξήγησης για να λύσουμε το πρόβλημα που λύνει το μαύρο κουτί; Αφού στη διατριβή ασχολούμαστε και προτείνουμε *post hoc* μεθόδους μαύρου κουτιού, θεωρούμε σημαντικό να απαντήσουμε σε αυτά τα επιχειρήματα. Ως προς το πρώτο, που αφορά την εξαγωγή χαρακτηριστικών και εξηγίσιμων απλών μεθόδων, απ'ότι φαίνεται από τις συγκρίσεις προσεγγίσεων στους περισσότερους πλέον τομείς, φαίνεται πως η αδιαφανής βαθιά μάθηση συστηματικά έχει καλύτερη απόδοση στα προβλήματα, η οποία αναγκαστικά θα θυσιαστεί εάν ακολουθήσουμε διαφορετική προσέγγιση, κάτι το οποίο ενδεχομέ-

νως να είναι επιθυμητό σε κάποιους τομείς, αλλά σίγουρα δεν είναι ένα γενικό συμπέρασμα για όλους τους τομείς. Σε ό,τι αφορά το δεύτερο επιχείρημα, το οποίο στηρίζει πως αν είχαμε ακριβείς *post hoc* μεθόδους εξήγησης μαύρου κουτιού, δεν θα χρειαζόμασταν το ίδιο το μαύρο κουτί, η απάντησή μας είναι πως στις περισσότερες περιπτώσεις ένα καλό σύστημα εξηγισιμότητας εκφράζει τις εξηγήσεις σε **διαφορετικό επίπεδο αφαίρεσης** από τα χαρακτηριστικά που βλέπει στην είσοδό του το μαύρο κουτί, και χρησιμοποιεί **συγκεκριμένη ορολογία**, η οποία είναι προσαρμοσμένη στον τελικό χρήστη.

Για να το πετύχουμε αυτό, αξιοποιούμε γράφους γνώσης, με τους οποίους μπορεί να αναπαρασταθεί σύνθετη πληροφορία με τρόπο ο οποίος είναι αναγνώσιμος από μηχανές και ταυτόχρονα κατανοητός από ανθρώπους. Ο τρόπος που το πετυχαίνουν αυτό είναι η ελευθερία που υπάρχει στον ορισμό της σημασιολογίας των στοιχείων του γράφου, καθώς και στην εκφραστικότητα. Για παράδειγμα στον γράφο των WikiData υπάρχουν ακμές "τόποςΓέννησης" που συνδέουν ανθρώπους με τοποθεσίες, ακμές "έχειΣυνεργαστείΜε" που συνδέουν ανθρώπους με ανθρώπους, και μεγάλο πλήθος διαφορετικών τύπων κόμβων, και ακμών, συντάσσοντας έτσι έναν γράφο γνώσης που φέρει τεράστιο όγκο πληροφορίας, σε εύκολα αναγνώσιμη μορφή. Τέτοιου τύπου πληροφορία είναι ιδανική για το πρόβλημα της εξηγισιμότητας, αρκεί να βρούμε τρόπους να περιγράψουμε τα συστήματα που θέλουμε να εξηγήσουμε αξιοποιόντας την πληροφορία αυτή. Με αυτόν τον τρόπο μπορεί κανείς να παίρνει εξηγήσεις εκφρασμένες με την ορολογία και αξιοποιώντας τη σημασιολογία του γράφου γνώσης.

Η σημαντικότητα της ορολογίας σε ό,τι αφορά την εξηγισιμότητα, την ερμηνεία, την αξιολόγηση και την εποπτεία μοντέλων είναι στον πυρήνα της παρούσας διατριβής, και δεν είναι ένα ζήτημα το οποίο έχει μελετηθεί εκτενώς από την ερευνητική κοινότητα. Για αυτό το λόγο, αναπτύξαμε ένα θεωρητικό πλαίσιο, που παρουσιάζεται στο **Πρώτο Κεφάλαιο** του βασικού σώματος της διατριβής, το οποίο αξιοποιεί τον φορμαλισμό των περιγραφικών λογικών και ορίζει τυπικά μεθόδους για την αξιολόγηση και την ερμηνεία ταξινομητών μηχανικής μάθησης, βασισμένη σε γράφους γνώσης. Στο **Δεύτερο Κεφάλαιο** της διατριβής αναδεικνύουμε τη χρησιμότητα του θεωρητικού μας πλαισίου, εφαρμόζο- ντας το και εκτελώντας διεξοδικά πειράματα για την αξιολόγηση και την ερμηνεία ταξινομητών εικόνας. Ύστερα, στο **Τρίτο** και το **Τέταρτο Κεφάλαιο**, μελετάμε πιο οριζόντια δύο διαφορετικά πεδία: αυτό των συμβολικών αναπαραστάσεων μουσικής, και της αναγνώρισης COVID-19 από ηχητικά αρχεία, πάντα υπό το πρίσμα της σημαντικότητας της ορολογίας σε ό,τι αφορά την εξηγισμότητα και την ερμηνεία συστημάτων μηχανικής μάθησης.

Κεφάλαιο 1: Εξηγήσεις βασισμένες σε γράφους γνώσης

Σε αυτό το κεφάλαιο εισάγουμε ένα θεωρητικό πλαίσιο το οποίο βασίζεται στον φορμαλισμό των περιγραφικών λογικών, και με βάση το οποίο μπορεί κανείς να ορίσει εξηγήσεις και μεθόδους αξιολόγησης, οι οποίες αξιοποιούν σημασιολογική πληροφορία, και εκφράζονται χρησιμοποιόντας συγκεκριμένη ορολογία.

Το κεφάλαιο ξεκινάει με τον ορισμό του **συνόλου δεδομένων εξήγησης** (explanation dataset), το οποίο είναι στον πυρήνα του του μια βάση γνώσης περιγραφικών λογικών. Συγκεκριμένα, ορίζοντας: α) ένα λεξιλόγιο (vocabulary) που αποτελείται από τρία ξένα μεταξύ τους σύνολα ονομάτων ατόμων, εννοιών, και ρόλων, β) ένα σύνολο αξιωμάτων (TBox) στο οποίο ορίζονται οι σημασιολογικές σχέσεις μεταξύ των διάφορων όρων που υπάρχουν στο λεξιλόγιο, και γ) ένα σύνολο από ισχυρισμούς (ABox) που περιγράφουν έναν κόσμο, χρησιμοποιώντας την ορολογία που έχει ορισθεί, έχουμε μία βάση γνώσης περιγραφικών λογικών. Το επιπλέον στοιχείο που χρειάζεται για να αποτελέσει αυτή ένα σύνολο δεδομένων εξήγησης, είναι ο ορισμός μίας ειδικής έννοιας, την οποία την ονομάζουμε Exemplar, η οποία δεν εμπλέκεται στο σύνολο των αξιομάτων (TBox), και δεν επηρεάζει τη σημασιολογία, ή το αποτέλσμα των αλγορίθμων συλλογιστικής. Αντ'αυτού, η έννοια αυτή χρησιμοποιείται στο σύνολο των ισχυρισμών (ABox) για να χαρακτηρίσει τα αντικείμενα εκείνα, τα οποία δεδομένου ενός ταξινομητή μπορούν να απεικονιστούν στο χώρο των χαρακτηριστικών ενός ταξινομητή που επιθυμούμε να εξηγήσουμε.

Έχοντας ένα σύνολο δεδομένων εξήγησης, έχουμε χτίσει ουσιαστικά μια γέφυρα μεταξύ ενός μαύρου κουτιού ταξινομητή, με μία τυπικά ορισμένη βάση γνώσης. Σε αυτό το πλαίσιο ορίζουμε εξηγήσεις κανόνων, που επιχειρούν να περιγράψουν τη λειτουργία του ταξινομητή χρησιμοποιώντας κανόνες που είναι εκφρασμένοι με την ορολογία του συνόλου δεδομένων εξήγησης, καθώς και αντιπαραθετικές εξηγήσεις, οι οποίες επιχειρούν να βρουν σημασιολογικά κοντά αντιπαραδείγματα για ένα δεδομένο, που παρά τη σημασιολογική ομοιότητα, ταξινομούνται σε διαφορετική κατηγορία. Και για τους δύο τύπους εξηγήσεων αναπτύσσουμε αλγορίθμους για τον υπολογισμό τους, και δείχνουμε πως τα αποτελέσματα αυτών μπορεί να είναι χρήσιμα, όχι μόνο για την εξηγησιμότητα, αλλά και για την στοχευμένη αξιολόγηση ταξινομητών.

Οι εξηγήσεις σε μορφή κανόνων βασίζονται στα σημασιολογικά ερωτήματα. Συγκεκριμένα, σε ένα σύνολο δεδομένων εξήγησης, κάθε στοιχείο που μπορεί να τροφοδοτηθεί στον ταξινομητή (exemplar) έχει μια όσο το δυνατόν αναλυτικότερη περιγραφή. Επίσης, κάθε σύνολο από τέτοια στοιχεία, έχει μια όσο το δυνατόν συνοπτική περιγραφή, η οποία έχει τη μορφή σημασιολογικού ερωτήματος που έχει τα συγκεκριμένα στοιχεία ως απαντήσεις. Επομένως, για να περιγράψουμε τη λειτουργία ενός ταξινομητή σε αυτό το πλαίσιο, καλούμαστε να βρούμε σημασιολογικά ερωτήματα, οι απαντήσεις των οποίων είναι υποσύνολο των στοιχείων που ο ταξινομητής ταξινομεί σε μια συγκεκριμένη κατηγορία. Όταν αυτό ισχύει, τότε το ερώτημα μπορεί να μεταφραστεί σε κανόνα της μορφής AN <σώμα του σημασιολογικού ερωτήματος> TOTE <ταξινομείται από τον ταξινομητή στη συγκεκριμένη κατηγορία>. Επίσης ορίζονται κάποιες μετρικές, οι οποίες λαμβάνουν υπ'όψη την επικάλυψη μεταξύ των απαντήσεων ενός σημασιολογικού ερωτήματος, και των στοιχείων που ο ταξινομητής ταξινομεί σε μια συγκεκριμένη κατηγορία, με στόχο την ποσοτικοποίηση της ικανότητας ενός κανόνα να περιγράφει τη λειτουργία του ταξινομητή για τη συγκεκριμένη κατηγορία. Για τον υπολογισμό τέτοιων κανόνων, στη διατριβή παρουσιάζονται δύο αλγόριθμοι (KGrules και KGrules-H), οι λεπτομέρειες των οποίων είναι εκτός του πεδίου μελέτης της διατριβής. Ο πρώτος είναι εξαντλητικός αλγόριθμος εκθετικής πολυπλοκότητας, ενώ ο δεύτερος εξερευνά με ευριστικά κριτήρια τον χώρο των σημασιολογικών ερωτημάτων προσπαθώντας να βρει καλές εξηγήσεις.

Ο δεύτερος τύπος εξηγήσεων που παρουσιάζεται στο θεωρητικό πλαίσιο είναι οι αντιπαραθετικές εξηγήσεις (counterfactual explanations). Αυτές τυπικά ορίζονται ως ένα ελάχιστο σύνολο αλλαγών που εφαρμόζονται σε ένα δεδομένο, ώστε να αλλάξει η πρόβλεψη ενός ταξινομητή στο συγκεκριμένο δεδομένο. Στο θεωρητικό πλαίσιο που αναπτύξαμε, η έννοια της "ελαχιστότητας" αφορά τη σημασιολογική απόσταση των δεδομένων, η οποία ορίζεται με βάση τα αξιώματα που υπάρχουν στο TBox του συνόλου δεδομένων εξήγησης. Παρόμοια με τις εξηγήσεις σε μορφή κανόνων, η ιδέα βασίζεται στο γεγονός πως κάθε στοιχείο του συνόλου δεδομένων έχει μια όσο το δυνατόν αναλυτική περιγραφή γίνεται, και πως μεταξύ αυτών των αναλυτικών περιγραφών μπορούμε να ορίσουμε μέτρα απόστασης. Επομένως, ψάχνουμε για το κοντινότερο σημασιολογικά δεδομένο το οποίο ταξινομείται σε διαφορετική κατηγορία από τον ταξινομητή, και το αποτέλεσμα έχει τη μορφή αλλαγών στους ισχυρισμούς (ABox) του συνόλου δεδομένων. Για τον υπολογισμό τέτοιων εξηγήσεων, στη διατριβή προτείνουμε έναν αλγόριθμο, ο οποίος προϋποθέτει κάποιες απλοποιήσεις στην υποκείμενη βάση γνώσης. Συγκεκριμένα θεωρούμε πως και τα αξιώματα (TBox), καθώς και οι ισχυρισμοί (ABox) του συνόλου δεδομένων, μπορούν να αναπαραστηθούν ως κατευθυνόμενοι γράφοι. Έπειτα το πρόβλημα που καλούμαστε να λύσουμε είναι ένα πρόβλημα απόστασης γράφων, το οποίο δυστυχώς είναι γνωστό πως ανήκει στην κλάση πολυπλοκότητας NP-hard, και δεν θα είναι εφικτό για μεγάλα σύνολα δεδομένων. Έτσι κάνουμε μια περετάιρω απλοποίηση, κωδικοποιώντας την πληροφορία από εξερχόμενες ακμές στους κόμβους προέλευσής τους, μετατρέπωντας το πρόβημα σε ένα πρόβλημα απόστασης συνόλων από σύνολα, το οποίο είναι πολύ ευκολότερο να λυθεί.

Συνοψίζοντας, το πρώτο κεφάλαιο εισάγει ένα κενοφανές θεωρητικό πλαίσιο, το οποίο μέσω των περιγραφικών λογικών ορίζει σημασιολογικές επεξηγήσεις ταξινομητών σε μορφή κανόνων και σε μορφή αντιπαραθέσεων, και προτείνει τρόπους για τον υπολογισμό τους. Αξιοποιώντας αυτήν την ιδέα, μπορεί κανείς να παράξει εξηγήσεις για ταξινομητές, χρησιμοποιώντας όμως ό,τι

ορολογία εκείνος θέλει, και έχοντας ορίσει τη σημασιολογία της ορολογίας αυτής με μαθηματική αυστηρότητα σε μια βάση γνώσης περιγραφικών λογικών. Έτσι, μπορούμε να κατασκευάσουμε σύνολα δεδομένων εξήγησης στο οποία τα δεδομένα περιγράφονται με ορολογία προσαρμοσμένη στον τελικό χρήστη που αναμένουμε αλληλεπιδράσει με το σύστημα, και να δώσουμε εξηγήσεις στο κατάλληλο επίπεδο αφαίρεσης. Για παράδειγμα οι εξηγήσεις που θα δωθούν σε ένα γιατρό θα χρησιμοποιούν διαφορετική ορολογία από αυτές που θα δωθούν σε έναν μηχανικό, ή έναν προγραμματιστή.

Κεφάλαιο 2: Σημασιολογικές εξηγήσεις για ταξινομητές εικόνας

Στο δεύτερο κεφάλαιο των περιεχομένων της διατριβής εφαρμόζουμε εκτενώς το θεωρητικό πλαίσιο που αναπτύχθηκε στο προηγούμενο κεφάλαιο, για την εξήγηση και την αξιολόγηση ταξινομητών εικόνας. Ο κύριος λόγος που επιλέξαμε τους ταξινομητές εικόνας για αυτό το κεφάλαιο είναι η ύπαρξη σχετικών δουλειών στην περιοχή, στις οποίες έχει αναδειχθεί η σημαντικότητα της σημασιολογίας σε ό,τι αφορά την εξηγισιμότητα. Αυτό γίνεται αρκετά προφανές αν σκεφτούμε πως ο χώρος των εικονοκυττάρων (pixels), στον οποίον δρουν οι ταξινομητές εικόνας, δεν είναι η αναπαράσταση που χτίζουμε στο μυαλό μας σαν άνθρωποι όταν κοιτάμε μια φωτογραφία. Αντιθέτως, εμείς αντιλαμβανόμαστε τις εικόνες ως ένα σύνολο πιο ασαφών εννοιών, και σχέσεων μεταξύ τους, που περιγράφουν το τί απεικονίζεται.

Το πρώτο σύνολο πειραμάτων που παρουσιάζεται σε αυτό το κεφάλαιο αφορά το σύνολο δεδομένων CLEVR-Hans3. Αυτό περιέχει συνθετικές εικόνες που απεικονίζουν τρισδιάστατα σχήματα, ενώ κάθε εικόνα συνοδεύεται από μια σημασιολογική περιγραφή, που αφορά το χρώμα, το σχήμα, το υλικό, το μέγεθος, και την τοποθεσία των αντικειμένων που απεικονίζονται. Οι εικόνες του συνόλου δεδομένων είναι χωρισμένες σε τρεις ξένες μεταξύ τους κατηγορίες, ορισμένες με βάση συγκεκριμένα αντικείμενα που απεικονίζονται. Για παράδειγμα η μία από τις τρεις κατηγορίες ορίζεται ως "Οι εικόνες που απεικονίζουν έναν μεγάλο κύβο και έναν μεγάλο κύλινδρο". Η ιδιαιτερότητα που έχει αυτό το σύνολο δεδομένων είναι πως οι δύο από τις τρεις κατηγορίες έχουν κάποιους εσκεμμένα ορισμένους παράγοντες σύγχυσης (confounding factors). Για παράδειγμα, στα σύνολα εκπαίδευσης και επαλήθευσης, στις εικόνες της πρώτης κατηγορίας (μεγάλος κύβος και μεγάλος κύλινδρος), ο μεγάλος κύβος είναι πάντα χρώματος γκρι. Έτσι αναμένουμε μοντέλα που είναι εκπαιδευμένα σε αυτό το σύνολο δεδομένων να υιοθετήσουν αυτούς τους παράγοντες σύγχυσης, το οποίο ύστερα, με τις μεθόδους εξηγησιμότητας μπορούμε να προσπαθήσουμε να το εντοπίσουμε. Μέσω των πειραμάτων μας δείχνουμε πως και οι εξηγήσεις σε μορφή κανόνων, και οι αντιπαραθετικές εξηγήσεις, μπορούν να αναδείξουν τις πολώσεις του υποκείμενου ταξινομητή, με τα καλύτερα ποιοτικά αποτελέσματα να προκύπτουν όταν συσσορεύουμε το σύνολο όλων των αντιπαραθετικών εξηγήσεων (global counterfactuals) σε ένα σύνολο δεδομένων εξήσγησης.

Το δεύτερο πείραμα που παρουσιάζεται έγινε στο σύνολο δεδομένων CUB, το οποίο περιέχει εικόνες πτηνών, χωρισμένες σε είδη πτηνών, οι οποίες συνοδεύονται από σημασιολογικές περιγραφές που αφορούν τα χαρακτηριστικά του κάθε πτηνού. Σε αυτό το σύνολο δεδομένων αναπαράχθηκε ένα πείραμα με τελικούς χρήστες, με σκοπό τη σύγκριση της προτεινόμενης μεθόδου για αντιπαραθετικές εξηγήσεις, με υπάρχουσες δουλειές που αφορούν σημασιολογικές εξηγήσεις. Συγκεκριμένα στους χρήστες δινόταν μια αρχική εικόνα, και δύο αντιπαραθετικές εικόνες, και οι χρήστες καλούνταν να επιλέξουν ποια από τις αντιπαραθετικές εικόνες είναι σημασιολογικά κοντινότερη στην αρχική. Ενώ το γενικό συμπέρασμα του πειράματος είναι πως οι περισσότεροι χρήστες δεν μπορούσαν να επιλέξουν με σιγουριά μια από τις δύο εικόνες, η δική μας μέθοδος αναδείχθηκε ως προτιμότερη για τους χρήστες. Κάτι αξιοσημείωτο εδώ είναι πως οι δουλειές με τις οποίες συγκριθήκαμε ανήκουν στην οικογένεια των μεθόδων "άσπρου κουτιού", καθώς χρειάζονται πρόσβαση στα βάρη του υποκείμενου μοντέλου, ενώ η δική μας προτεινόμενη μέθοδος είναι "μαύρου κουτιού".

Το τελευταίο σύνολο πειραμάτων που αφορούν ταξινομητές εικόνας, έγινε σε ένα πιο ρεαλιστικό πλαίσιο, όπου επιχειρούμε να εξηγήσουμε γενικού σκοπού ταξινομητές, εκπαιδευμένους σε

διαδεδομένα σύνολα δεδομένων όπως είναι το ImageNet και το PLACES, χρησιμοποιώντας ως σύνολα δεδομένων εξήγησης, σύνολα δεδομένων που φέρουν σημασιολογική πληροφορία όπως είναι το COCO και το Visual Genome. Σε όλες τις περιπτώσεις σε αυτό το σύνολο πειραμάτων, η ορολογία είναι βασισμένη στην ιεραρχία υπερωνύμων και υπονύμων του WordNet. Το αποτέλεσμα αυτών των πειραμάτων είναι μια ποιοτική ανάλυση των αποτελεσμάτων των προτεινόμενων αλγορίθμων, και μία συζήτηση για τη σημαντικότητα της ορολογίας και της σημασιολογίας για το πρόβλημα της εξήγησης και της αξιολόγησης αδιαφανών συστημάτων. Για παράδειγμα, ένας κανόνας εξήγησης ενός ταξινομητή εκπαιδευμένου στο ImageNet που προέκυψε ήταν: "Αν η εικόνα απεικονίζει ζώο το οποίο φοράει κάτι, τότε η εικόνα ταξινομείται ως κατοικίδιο ζώο". Τέτοιοι κανόνες μπορούν να είναι πολύ χρήσιμοι για την αποσφαλμάτωση συστημάτων. Για παράδειγμα ο συγκεκριμένος κανόνας θα παρότρυνε τους προγραμματιστές του μοντέλου να ελέγξουν αν δώσουν στον ταξινομητή εικόνες από άγρια ζώα που φορούν αντικείμενα, αν αυτές θα ταξινομηθούν (εσφαλμένα) ως κατοικίδια ζώα. Ένα άλλο παράδειγμα από τα πειράματα του κεφαλαίου, αφορά κανόνες εξήγησης για ταξινομητή που είχε εκπαιδευτεί στο σύνολο δεδομένων PLACES. Σε αυτό το πείραμα, επιλέξαμε με βάση τον πίνακα σύγχυσης, τις δύο πιο μπερδεμένες για τον ταξινομητή κατηγορίες, οι οποίες ήταν "δρόμος ερήμου" (desert road) και "άμμος ερήμου" (desert sand). Οι κανόνες που προέκυψαν έχουν εξαιρετικό ενδιαφέρον, καθώς για την κατηγορία "άμμος ερήμου", σε όλους τους κανόνες που μεγιστοποιούσαν τις μετρικές, εμφανιζόταν ως στοιχείο μέσα η έννοια "δρόμος′′. Με βάση αυτό, καθώς και τη χαμηλή απόδοση του ταξινομητή, και τη σύγχυση των δύο αυτών κατηγοριών, υποθέτουμε πως οι ετικέτες των δύο αυτών κατηγοριών δίνονται ανάποδα από τους δημιουργούς του συνόλου. Αντίστοιχα ενδιαφέροντα αποτελέσματα προκύπτουν και από την εφαρμογή του αλγορίθμου για τις αντιπαραθετικές εξηγήσεις, όπου για παράδειγμα εντοπίσαμε πως τα μοντέλα εκπαιδευμένα στο PLACES είναι πολωμένα όταν υπάρχει ζώο στην εικόνα, την ταξινομούν συχνά ως κτηνιατρίο, ακόμα και αν η εικόνα είναι εντελώς διαφορετικός χώρος (π.χ. κρεβατοκάμαρα ή κουζίνα).

Συνοψίζοντας, το κεφάλαιο αυτό αναδεικνύει κυρίως ποιοτικά, αλλά σε περιπτώσεις και ποσοτικά, τη χρησιμότητα του προτεινόμενου θεωρητικού πλαισίου. Μεταξύ των τύπων εξηγήσεων (κανόνες, αντιπαραθέσεις), όλες φανέρωσαν χρήσιμη πληροφορία για τους ταξινομητές που μελετήθηκαν, ενώ έγιναν εμφανή και τα θετικά και τα αρνητικά των διαφόρων τύπων αλγορίθμων, οι οποίοι έχουν μεγάλο περιθώριο για να βελτιστοποιηθούν. Το βασικό συμπέρασμα αυτού του κεφαλαίου είναι η σημαντικότητα του ορισμού ενός "καλού" συνόλου δεδομένων εξήγησης, και της κατάλληλης ορολογίας, καθώς είναι ουσιαστικά ο μόνος παράγοντας που επηρρεάζει την ποιότητα των εξηγήσεων, και το τί πληροφορία αυτές μεταδίδουν. Τέλος, συζητάμε πως οι κανόνες και οι γενικεύσεις των αντιπαραθέσεων (global counterfactuals) μπορούν να χρησιμοποιηθούν για τη στοχευμένη αξιολόγηση ταξινομητών, όπως είναι για παράδειγμα ο εντοπισμός συγκεκριμένων πολώσεων.

Κεφάλαιο 3: Εξηγισιμότητα και ερμηνεία στο πεδίο των συμβολικών αναπαραστάσεων μουσικής

Στο τρίτο κεφάλαιο των περιεχομένων της διατριβής μελετάται πιο οριζόντια το πεδίο των συμβολικών αναπαραστάσεων μουσικής, πάντα υπό το πρίσμα της σημαντικότητας της ορολογίας σε ό,τι αφορά την αξιολόγηση και την εξηγησιμότητα. Ο λόγος που επελέχθηκαν οι συμβολικές αναπαραστάσεις είναι η ευκολία διαχείρισης και ανάκτησης δεδομένων, σε σχέση με ηχητικά αρχεία μουσικής, καθώς αυτά είναι ευρέως διαθέσιμα και δεν υπόκεινται στους ίδιους νομικούς περιορισμούς σχετικά με πνευματικά δικαιώματα. Επιπρόσθετα, από συμβολικές αναπαραστάσεις μπορεί κανείς ευκολότερα να εξάγει πληροφορία σχετική με τη θεωρία της μουσικής. Γενικά η υπολογιστική μουσικολογία έχει μελετηθεί εκτενώς και είναι γνωστές οι δυσκολίες που φέρει, όπως είναι για παράδειγμα η ύπαρξη πληροφορίας σε πολλαπλά χρονικά εύρη, ή όπως είναι η υποκειμενικότητα σε ό,τι αφορά ετικέτες δεδομένων. Σε αυτό το κεφάλαιο μελετώνται δύο προβλήματα: η αξιολόγηση συστημάτων για την αυτόματη σύνθεση μουσικής, και η αναγνώριση είδους

μουσικής.

Η αξιολόγηση συστημάτων που συνθέτουν μουσική είναι εξαιρετικά δύσκολο πρόβλημα, γεγονός που πηγάζει κυρίως από την υποκειμενικότητα της μουσικής, αλλά και από τις τόσες διαφορετικές πτυχές που μπορεί να έχει ένα κομμάτι μουσικής. Για παράδειγμα μπορεί η μουσική ενός συστήματος να αρέσει περισσότερο σε κάποια ομάδα ανθρώπων σε σχέση με άλλους, και μπορεί μια μετρική αξιολόγησης να είναι καλύτερη για κάποιο μοντέλο, ενώ άλλη μετρική χειρότερη. Χρησιμοποιώντας την κατάλληλη αναπαράσταση δεδομένων και το κατάλληλο επίπεδο αφαίρεσης, μπορούμε να αξιολογήσουμε τέτοια συστήματα ώς προς συγκεκριμένες πτυχές τους. Συγκεκριμένα στη διατριβή, δουλέυοντας με αναπαράσταση της μουσικής ως αλληλουχία συνόλων από νότες, ορίζουμε έναν μεθοδικό τρόπο με τον οποίο μπορεί κανείς να ορίσει μετρικές αξιολόγησης, ο οποίος βασίζεται σε μουσικοθεωρητικές ιδέες όπως είναι το Tonnetz, και σε απλές πράξεις συνόλων. Στο πλαίσιο αυτό ορίζουμε τέσσερα τέτοια ευριστικά κριτήρια αξιολόγησης, και με αυτά συγκρίνουμε διαφορετικά μοντέλα σύνθεσης μουσικής. Έχοντας ως βάση την αξιολόγηση από ανθρώπους (μουσικούς και μη), φαίνεται πως τέτοια κριτήρια μπορούν να χρησιμοποιηθούν για την προσέγγιση του προβλήματος της αξιολόγησης. Παρ'όλα αυτά, για να χρησιμοποιηθούν ως αυστηρά και έμπιστα κριτήρια αξιολόγησης, θα πρέπει να βασιστούν πιο στιβαρά στη θεωρία της μουσικής καθώς τα κριτήρια που εμείς ορίσαμε βασίζονται σε υποθέσεις και παρατηρήσεις. Για να το πετύχουμε αυτό θα πρέπει να αναπτύξουμε μεθόδους για την κωδικοποίηση της θεωρίας της μουσικής σε γράφους γνώσης, το οποίο είναι ενεργό ερευνητικό πεδίο.

Το δεύτερο πρόβλημα που μελετάμε σε συμβολικές αναπαραστάσεις μουσικής είναι η αναγνώριση είδους μουσικής. Αυτό το πρόβλημα έχει ενδιαφέρουσες δυσκολίες, όπως είναι η υποκειμενικότητα στα είδη μουσικής, η ποικιλία και ανισσοροπία των δεδομένων, η μεταβολή των χαρακτηριστικών των ειδών μουσικής ανά τα χρόνια, καθώς και η συμβολική αναπαράσταση η ίδια, στην οποία δεν είναι κωδικοποιημένη πληροφορία του ηχοχρώματος και της ερμηνείας, η οποία για κάποια είδη είναι καθοριστική. Ένα ακόμα ενδιαφέρον αυτού του προβλήματος είναι πως κάποιες από τις αποδοτικότερες προσεγγίσεις στη βιβλιογραφία είναι εξηγίσιμες. Συγκεκριμένα, με αλγορίθμους αναγνώρισης προτύπων, εξάγουν μουσικά θέματα από την παρτιτούρα, και στη συνέχεια χρησιμοποιώντας την ύπαρξη μουσικών θεμάτων σαν χαρακτηριστικά εκπαιδεύουν απλούς και εξηγήσιμους ταξινομητές για το πρόβλημα της αναγνώρισης είδους. Αυτό μας κινητοποιεί πρώτων να προσεγγίσουμε το πρόβλημα χρησιμοποιώντας μονοδιάστατα συνελικτικά δίκτυα, τα οποία ενδεχομένως να μπορούν να μάθουν μουσικά θέματα όπως οι αλγόριθμοι αναγνώρισης προτύπων, και δεύτερον να μελετήσουμε την εξηγισιμότητα στο πρόβλημα αυτό, συγκρίνοντας διαφορετικές μεθόδους, και μελετώντας ποιοτικά τη χρησιμότητα της ορολογίας που χρησιμοποιούνται στις εξηγήσεις, στα πλαίσια της γενικότερης ιδέας που συζητάμε στη διατριβή.

Για να μελετήσουμε τη χρησιμότητα των μονοδιάστατων συνελικτικών δικτύων ως μοντέλο για συμβολικές αναπαραστάσεις μουσικής, σχεδιάσαμε και εκτελέσαμε ένα σύνολο πειραμάτων στο οποίο κρατούσαμε σταθερά το πλήθος εκπαιδεύσιμων παραμέτρων, και το δεκτικό πεδίο των δικτύων, ενώ μεταβλητά ήταν το βάθος δικτύων και το πλάτος των συνελικτικών πυρήνων. Επιπρόσθετα πειραματιστήκαμε με μία δική μας κενοφανή ιδέα, στην οποία αξιοποιούμε την αρχική αλληλου- χία που αναπαριστά τη μουσική, σε πολλαπλές χρονικές αναλύσεις (MuSeRe - Multiple Sequence Resolution). Η ιδέα αυτή σκοπεύει τη "διευκόλυνση" των δικτύων να μάθουν μουσικά θέματα που επαναλαμβάνονται μετά από μεγάλο χρονικό διάστημα, καθώς και πληροφορία για τη δομή της μουσικής που δέχονται στην είσοδο. Τα αποτελέσματα αυτών των πειραμάτων θεωρούμε πως είναι πολύ ενδιαφέροντα από διάφορες απόψεις. Πρώτων, πολλές από τις εκδοχές των συνελικτικών δικτύων που σχεδιάσαμε, ξεπεράσανε σε απόδοση τα καλύτερα μοντέλα της βιβλιογραφίας, και μάλιστα σε κάποιες περιπτώσεις κατά πολύ (15% στη μετρική F1 score). Αυτό αναμενόταν σε κάποιο βαθμό, αν κοιτάξει κανείς πως έχουν υπερτερήσει τα βαθιά νευρωνικά δίκτυα σε άλλα πεδία δεδομένων πέραν των συμβολικών αναπαραστάσεων μουσικής. Δεύτερον, στα πειράματά μας φάνηκε τα "πλατιά" δίκτυα να είναι συστηματικά καλύτερα από τα "βαθιά" ως προς της επίδοσή τους. Αυτό είναι ένα ενδιαφέρον συμπέρασμα για την σχεδίαση και την αρχιτεκτονική μονοδιάστατων συνελικτικών νευρωνικών δικτύων ευρύτερα, το οποίο ενώ είναι εκτός πεδίου της

παρούσας διατριβής, είναι κάτι που αξίζει να μελετηθεί πιο διεξοδικά. Τέλος από τα πειράματά μας εξάγουμε κάποια επιπλέον συμπεράσματα που αφορούν την προσέγγιση της αδύναμης επίβλεψης, και τη σημαντικότητα του μήκους του μουσικού αποσπάσματος που βλέπει στην είσοδό του το νευρωνικό δίκτυο κάθε στιγμή, και πως αυτές επηρρεάζονται από την ανισσοροπία, και την ασάφεια που υπάρχει στις ετικέτες που καλείται να μάθει.

Βέβαια, όπως προαναφέρθηκε, πολλές από τις υπάρχουσες προσεγγίσεις της βιβλιογραφίας είναι εν γενώς εξηγήσιμες, κάτι το οποίο είναι δύσκολο να ποσοτικοποιηθεί. Στην πραγματικότητα, θεωρούμε πως υπάρχουν περιπτώσεις εφαρμογών, στις οποίες δεν αξίζει να θυσιαστεί η εν γενώς ερμηνευσιμότητα χάρην της βελτίωσης της επίδοσης. Ένα τέτοιο παράδειγμα εφαρμογής θα ήταν ένας μουσικός ο οποίος έχει γράψει μια παρτιτούρα, και θέλει να σκεφτεί ιδέες για την ενορχήστρωση. Για εκείνον θα ήταν πολύ πιο χρήσιμο, πέρα από τα είδη μουσικής, να υπάρχει δικαιολόγηση βασισμένη σε μουσικά θέματα που εντοπίστηκαν στην παρτιτούρα ως σημαντικά για το κάθε είδος. Με αυτό το σκεπτικό, μελετήσαμε *post hoc* μεθόδους εξήγησης για προσπαθήσουμε να ανακτήσουμε την ερμηνευσιμότητα που θυσιάσαμε με την αδιαφανή προσέγγιση των βαθιών συνελικτικών δικτύων. Η πρώτη οικογένεια μεθόδων που εφαρμόσαμε ήταν οι μέθοδοι σημαντικότητας χαρακτηριστικών (feature importance). Αυτές αμέσως φάνηκαν πως δεν είναι χρήσιμες για το πρόβλημα, και μπορεί να είναι εν δυνάμει παραπλανιτικές, καθώς ο χώρος των χαρακτηριστικών (pianorolls - πίνακες με διάσταση $128 \times t$, όπου t η διάρκεια, σε ανάλυση συνήθως 24 δείγματα ανά τέταρτο) είναι πολύ σύνθετος. Για παράδειγμα μπορεί σε ένα κομμάτι μουσικής να εμφανιστεί σαν σημαντικό χαρακτηριστικό μια νότα που παίζεται μια συγκεκριμένη στιγμή, αλλά δεν γνωρίζουμε από το ευρύτερο κομμάτι γιατί η νότα θεωρήθηκε σημαντική, οδηγόντας μας να κάνουμε υποθέσεις αναλύ- οντας την εξήγηση, οι οποίες μπορεί να είναι εσφαλμένες. Ένα άλλο παράδειγμα είναι πως πολλές φορές εμφανιζόταν σαν σημαντικό χαρακτηριστικό η έλλειψη νοτών σε ένα συγκεκριμένο εύρος και μια χρονική στιγμή, το οποίο πάλι μπορεί να οδηγήσει κάποιον χρήστη στο να παραπλανηθεί. Βασισμένοι στην ιδέα μας για εξηγήσεις που χρησιμοποιούν συγκεκριμένη ορολογία και αναφέρονται σε διαφορετικά επίπεδα αφαίρεσης, τροποποιήσαμε κάποιους αλγορίθμους σημαντικότητας χαρακτηριστικών. Συγκεκριμένα, υπάρχει μια οικογένεια μεθόδων οι οποίες ψάχνουν να βρουν σημαντικά χαρακτηριστικά σε μια γκαουσιανή γειτονιά γύρω από ένα δεδομένο, επιχειρώντας να μιμηθούν τον ταξινομητή με εν γενώς εξηγήσιμα μοντέλα. Εμείς τροποποιήσαμε αυτές τις μεθόδους με δύο τρόπους. Αρχικά προσθέσαμε μια συνάρτηση με πεδίο ορισμού τον χώρο των χαρακτηριστικών και πεδίο τιμών την πληροφορία που θέλουμε να εμφανίζεται στην εξήγηση. Συγκεκριμένα, το μοντέλο που καλείται να μιμηθεί τον ταξινομη- τή δέχεται στην είσοδό του τη συχνότητα εμφάνισης διαστημάτων νοτών στη μουσική, αντί για ολόκληρο το κομμάτι σε αναπαράσταση pianoroll. Επιπρόσθετα, ορίσαμε τη γειτονιά βασισμένοι στην επιθυμητή ορολογία (διαστήματα νοτών), αντί για γκαουσιανή στο χώρο των χαρακτηριστικών, ορίζοντας την απόσταση στο χώρο της συχνότητας διαστημάτων νοτών. Τέλος, μελετήσαμε και άλλους τύπους εξηγήσεων, όπως είναι αυτές των χαρακτηριστικών παραδειγμάτων και κριτικής (prototypes and criticisms), οι οποίες ενώ δεν ήταν τόσο ικανοποιητικές σαν εξηγήσεις των μοντέλων, ανέδειξαν κάποια χρήσιμη πληροφορία, όπως είναι κάποια δεδομένα τα οποία έχουν μάλλον λάθος ετικέτες.

Συνοψίζοντας, στο κεφάλαιο αυτό μελετάμε οριζόντια το πεδίο των συμβολικών αναπαραστάσεων μουσικής, και αναδείξαμε τη σημαντικότητα της ορολογίας σε ό,τι αφορά εξηγισιμότητα και αξιολόγηση, και πώς για τον σκοπό αυτό μπορεί να συνεισφέρει τυπικά αναπαραστημένη γνώση μουσικής θεωρίας. Σχετικά με την αξιολόγηση, όπως για τα συστήματα μηχανικής μετάφρασης χρησιμοποιούνται σύνθετες προσεγγίσεις για την αξιολόγηση διαφορετικών πτυχών των μεταφράσεων (πχ. ακρίβεια, συντακτική ορθότητα, έλεγχος για επιβλαβή στερεότυμα), έτσι και για την αυτόματη σύνθεση μουσικής χρειάζονται διαφορετικά μέτρα που να λαμβάνουν υπ'όψη διαφορετικές πτυχές της μουσικής. Σε αυτό το πλαίσιο, προτείναμε μια μέθοδο για την ποσοτικοποίηση απλών ιδεών που σχετίζονται με σύνολα νοτών και με την αρμονία τους. Ένα τέτοιο σύστημα πρέπει να είναι γερά βασισμένο σε ένα μουσικοθεωρητικό πλαίσιο το οποίο μας κινητοποιεί για την τυπική αναπαράσταση πιο σύνθετων εννοιών από τη μουσική θεωρία. Επιπλέον, εξηγώντας τα συστήματα για την αναγνώριση είδους μουσικής, ακόμα και με πολύ απλές μουσικοθεωρητικές

έννοιες, όπως είναι η εμφάνιση μουσικών διαστημάτων στη μουσική, παράξαμε εξηγήσεις οι οποίες ήταν ακριβείς, και πολύ πιο απλές στην κατανόηση για κάποιον χρήστη από αυτές της σημαντικότητας χαρακτηριστικών. Αναδείξαμε επίσης τη σημαντικότητα της εξηγησιμότητας ακόμα και για προβλήματα στα οποία δεν φέρει μεγάλη προτεραιότητα, καθώς δεν βασίζονται σε αυτά κρίσιμες αποφάσεις, αν και η αξία της είναι δύσκολο να ποσοτικοποιηθεί. Από πλευράς αλληλεπίδρασης ανθρώπου-υπολογιστή για παράδειγμα, δεν έχει τόση αξία 15% ψηλότερη τιμή της μετρικής F1-score όσο έχει η αιτιολόγηση της εξήγησης με μουσικά θέματα. Τέλος, στα περισσότερα πεδία υπάρχει η αβεβαιότητα, η ασάφεια και η υποκειμενικότητα στις ετικέτες δεδομένων, τα οποία είναι ενισχυμένα στα είδη μουσικής. Έτσι, μελετώντας αυτό το πρόβλημα, αναδεικνύεται η εξηγισιμότητα με χρήση κατάλληλης ορολογίας ως χρήσιμο εργαλείο για να κατανοήσουμε τα μοντέλα, και τα σύνολα δεδομένων στα οποία τα εκπαιδεύουμε.

Κεφάλαιο 4: Εξηγισιμότητα για την αναγνώριση COVID-19 από αρχεία ήχου

Στο τέταρτο και τελευταίο κεφάλαιο των περιεχομένων της διατριβής, μελετάμε το πρόβλημα της αναγνώρισης COVID-19 από ηχογραφήσεις βήχα, αναπνοής και ομιλίας. Στην αρχή της πανδημίας η προσοχή της ερευνητικής κοινότητας στράφηκε στο να αναπτύξει προτότυπες λύσεις στα διάφορα κενοφανή προβλήματα που είχαν προκύψει λόγω του ιού, όπως η καταμέτρηση περιστατικών, η συλλογή δημογραφικών στοιχείων, η πρόβλεψη περιστατικών και η διάγνωση. Μία από αυτές τις εν δυνάμει λύσεις ήταν η αναγνώριση του ιού από ηχητικά αρχεία που έχουν ηχογραφηθεί από κινητό. Από τους πρώτους μήνες κιόλας ξεκίνησαν να αναπτύσσονται σύνολα δεδομένων από ηχητικά αρχεία, επισημειωμένα ως προς αν τα υποκείμενα φέρουν τη νόσο, καθώς και μοντέλα που αξιοποιούν αυτά τα σύνολα δεδομένων για την πρόβλεψη ύπαρξης του ιού. Σε αυτό το πλαίσιο περιγράφεται η μεθοδολογία που αναπτύχθηκε για την επίλυση αυτού του προβλήματος, η οποία κατέκτησε την πρώτη θέση στο διαγωνισμό "Sensory Informatics Challenge" που διεξήχθει από την IEEE στο συνέδριο IEEE Healthcare Summit 2021. Αξιοσημείωτο είναι πως ένα από τα κριτήρια αξιολόγησης στο διαγωνισμό ήταν η ερμηνευσιμότητα. Στη συνέχεια περιγράφεται η ανάπτυξη του συνόλου δεδομένων και κυρίως της βάσης γνώσης smarty4covid, το οποίο χρησιμοποιήθηκε πειραματικά ως σύνολο δεδομένων εξηγήσεων, όπως αυτό είναι ορισμένο στο θεωρητικό πλαίσιο του πρώτου κεφαλαίου. Πάνω στο σύνολο δεδομένων αυτό μελετάμε διάφορες προσεγγίσεις για την επίλυση του προβλήματος της αναγνώρισης της νόσου με εξηγήσιμο τρόπο, και όμοια με τη μελέτη του τρίτου κεφαλαίου που αφορούσε τις συμβολικές αναπαραστάσεις μουσικής, καταλήγουμε σε διάφορα ενδιαφέροντα συμπεράσματα.

Στο διαγωνισμό της IEEE, όπου όπως αναφέρθηκε ένα από τα κριτήρια ήταν η ερμηνευσιμότητα, μας δινόταν ένα σύνολο δεδομένων από ηχητικά αρχεία βήχα, αναπνοής και ομιλίας από 965 ασθενείς, με το οποίο αναπτύξαμε και εκπαιδεύσαμε τα μοντέλα μας, τα οποία ύστερα αξιολογήθηκαν σε ένα κρυφό σύνολο δεδομένων. Η δική μας προσέγγιση ήταν να εκπαιδεύσουμε απλά συνελικτικά δίκτυα στα φασματογραφήματα των ηχητικών αρχείων, με ασθενή επίβλεψη, όπου το κάθε συνελικτικό δίκτυο εκπαιδεύεται σε πολύ μικρής διάρκειας σημεία των ηχητικών αρχείων (< 1 δευτερόλεπτο), αλλά παίρνωντας ως ετικέτα αυτήν ολόκληρου του αρχείου. Οι κύριοι λόγοι που επιλέξαμε αυτήν την προσέγγιση είναι: α) το μικρό πλήθος δεδομένων, όπου πιο σύνθετα δίκτυα ενδεχομένως να δυσκολεύονταν να γενικεύσουν β) η διαίσθησή μας πως αν υπάρχουν χαρακτηριστικά του ήχου χρήσιμα για την πρόβλεψη του ιού, αυτά κατά πάσα πιθανότητα θα αφορούν τη στιγμιαία αντίληψη του ήχου, και όχι τόσο τα μεγάλα χρονικά εύρη. Για παράδειγμα, σε ένα ηχητικό αρχείο 30 δευτερολέπτων ενός ανθρώπου να βήχει, αν υπάρχει πληροφορία σχετική με τον ιό, αυτή μάλλον θα υπάρχει σε κάθε βήχα ξεχωριστά, επομένως το δίκτυό μας δεν χρειάζεται να έχει μεγάλο δεκτικό πεδίο. Η ασθενής επίβλεψη προκύπτει από το γεγονός πως ενώ το δίκτυο εκπαιδεύεται σε μικρά σημεία του κάθε ηχητικού αρχείου, επιλεγμένα τυχαία (για παράδειγμα μπορεί τυχαία να επιλεχθεί σημείο σιγής στο οποίο προφανώς δεν υπάρχει πληροφορία σχετική με τον ιό), σαν ετικέτα λαμβάνει την ετικέτα όλου του αρχείου. Έχοντας εκπαιδεύσει τα δίκτυα με

αυτον τον τρόπο, μετά η τελική πρόβλεψη για ένα νεό δεδομένο προκύπτει παίρνωντας το μέσο όρο των προβλέψεων των δικτύων, σε κάθε χρονική στιγμή. Η απλή προσέγγιση αυτή μας επιτρέπει αμέσως να εντοπίσουμε ποια σημεία του ηχητικού αρχείου ήταν σημαντικά για την πρόβλεψη, αφού γνωρίζουμε πως η πρόβλεψη είναι αποτέλεσμα της μέσης τιμής των εξόδων των δικτύων σε κάθε σημείο του ηχητικού σήματος. Ύστερα, κάθε ένα από αυτά τα σημεία που εντοπίστηκαν ως σημαντικά μπορεί να αναλυθεί περεταίρω χρησιμοποιώντας *post hoc* μεθόδους εξήγησης από τη βιβλιογραφία.

Έχοντας αναπτύξει το θεωρητικό πλαίσιο του πρώτου κεφαλαίου, αναδεικνύοντας τη χρησιμότητά τού στο να παράγει σημασιολογικές εξηγήσεις για μάυρα κουτιά με *post hoc* προσέγγιση, όταν προέκυψε το ερευνητικό ενδιαφέρον για την αναγνώριση COVID-19 από ήχο προέκυψε μαζί και η ανάγκη για την ανάπτυξη ενός συνόλου δεδομένων εξήγησης, όπως αυτό είναι ορισμένο στο πρώτο κεφάλαιο. Έτσι, στα πλαίσια του έργου smarty4covid, αναπτύξαμε ένα σύνολο δεδομένων το οποίο συνοδεύεται από μια πλούσια βάση γνώσης η οποία κωδικοποιεί πληροφορία σχετική με δημογραφικά στοιχεία, υποκείμενα νοσήματα, συμπτώματα, χαρακτηρισμούς ειδικών κ.α. χρησιμοποιώντας ιατρική ορολογία οργανωμένη ιεραρχικά, βασισμένη στη γνώση SNOMED-CT. Έπειτα αυτή η βάση γνώσης μπορεί να αξιοποιηθεί για την *post hoc* επεξήγηση μοντέλων-μαύρων κουτιών για την πρόβλεψη COVID-19. Με αυτό το σκεπτικό, αρχικά αξιοποιήσαμε τις ετικέτες του smarty4-covid για να αξιολογήσουμε το προανεφερθέν μοντέλο του διαγωνισμού της IEEE. Προς έκπληξή μας, το μοντέλο αυτό είχε επίδοση χειρότερη από τυχαίο ταξινομητή, πετυχαίνωντας 0.49 στη μετρική Area Under the Receiver Operator Characteristic Curve - AUC. Υποθέτουμε πως για την απόκλιση αυτή σε επίδοση μεταξύ των συνόλων δεδομένων ενδεχομένως να οφείλονται παράγοντες όπως είναι οι διαφορετικές μεταλλάξεις του ιού και τα διαφορετικά συμπτώματα που εκείνες φέρουν, τα αυξημένα ποσοστά εμβολιασμού στον πληθυσμό, καθώς και οι κατανομή των καπνιστών στο δείγμα. Για να μελετήσουμε τα αίτια της χαμηλής επίδοσης, αξιοποιήσαμε το σύνολο δεδομένων και τη βάση γνώσης του smarty4covid για να παράξουμε αντιπαραθετικές εξηγήσεις για το μοντέλο του διαγωνισμού. Από τα αποτελέσματα αμέσως έγινε ξεκάθαρο πως το μοντέλο ήταν πολωμένο ως προς το φύλλο, και ως προς τις ηλικιακές ομάδες των χρηστών, το οποίο εξακριβώθηκε ύστερα από στατιστική ανάλυση στο σύνολο δεδομένων εκπαίδευσης του μοντέλου. Έτσι αναδεικνύεται για μια ακόμη φορά η χρησιμότητα του προτεινόμενου θεωρητικού πλαισίου, και τονίζεται η σημαντικότητα της ορολογίας και των σημασιολογικών εξηγήσεων για την κατανόηση, αξιολόγηση και ερμηνεία των αδιαφανών μοντέλων.

Ένα επιπλέον σύνολο πειραμάτων που πραγματοποιήθηκαν σε αυτόν τον τομέα περιελάμβανε την εξαγωγή χαρακτηριστικών από όλα τα αρχεία ήχου χρησιμοποιώντας μεθόδους ψηφιακής επεξεργασίας σήματος (DSP), και ύστερα συνδυάζοντας αυτά με τα αυτοαναφερόμενα χαρακτηριστικά από την πλατφόρμα smarty4covid, δημιουργώντας έτσι μια έκδοση του συνόλου δεδομένων σε μορφή πίνακα. Στη συνέχεια εκπαιδεύτηκαν σε αυτό το σύνολο απλούς, εγγενώς ερμηνεύσιμους ταξινομητές, όπως είναι τα δέντρα αποφάσεων, η λογιστική παλινδρόμηση και ο Naive Bayes. Ο κύριος λόγος που μας κινητοποίησε για να εκτελέσουμε αυτά τα πειράματα είναι η θέση πολλών ερευνητών της εξηγήσιμης τεχνητής νοημοσύνης που αναφέρεται στην τρίτη παράγραφο της εκτεταμένης αυτής περίληψης, πως για προβλήματα κρίσιμης σημασίας πρέπει να αναπτύσσουμε μεθοδολογίες οι οποίες με *ante hoc* λογική να είναι εξηγήσιμες. Τα αποτελέσματα αυτών των πειραμάτων είναι ενδιαφέροντα καθώς οι απλοί αυτοί ταξινομητές πετύχαιναν συστηματικά την καλύτερη επίδοση από όσους δοκιμάστηκαν, με τη βέλτιστη να προκύπτει από τον ταξινομητή XG-Boost. Σε ό,τι αφορά όμως την εγγενώς ερμηνευσιμότητα, αναδεικνύονται μέσω των πειρμάτων μας κάποια ζητήματα τα οποία δεν έχουν συζητηθεί εκτενώς από την ερευνητική κοινότητα. Το πρώτο είναι πως ακόμα και να είναι πλήρως διαφανείς οι ταξινομητές, αν τα χαρακτηριστικά τα ίδια, και η ορολογία με την οποία αυτά είναι εκφρασμένα δεν είναι κατανοητά από ανθρώπους, τότε ουτε οι εξηγήσεις θα είναι. Αυτό έγινε εμφανές στη δική μας περίπτωση, κυρίως σε ό,τι αφορά τα χαρακτηριστικά από την επεξεργασία σήματος, τα οποία για να τα κατανοήσει κανείς θα πρέπει να έχει γνώση του πως αυτά εξάχθηκαν. Το δεύτερο συμπέρασμα που συζητάμε είναι πως η εγγενώς ερμηνευσιμότητα των μοντέλων πολλές φορές παρουσιάζεται με πολύ απλοποιημένη μορφή, η

οποία ενδεχομένως να είναι παραπλανητική για κάποιον χρήστη, εάν εκείνος δεν έχει καλή γνώση του πως λειτουργεί το υποκείμενο μοντέλο. Χαρακτηριστικό παράδειγμα αυτού είναι οι εξηγήσεις σημαντικότητας χαρακτηριστικών (feature importance) που προκύπτουν από τον αλγόριθμο XGBoost. Συγκεκριμένα, η βιβλιοθήκη που έχουν αναπτύξει οι ερευνητές που κατασκεύασαν τον αλγόριθμο, προσφέρει διάφορους τύπους εξηγήσεων σημαντικότητας χαρακτηριστικών. Τα αποτελέσματα αυτών των τύπων (για το ίδιο μοντέλο) είναι διαφορετικά μεταξύ τους, επομένως δεν θα ήταν σωστό να δώσουμε ένα σύνολο χαρακτηριστικών ως απλά "σημαντικά" σε ένα χρήστη, αλλά θα πρέπει να είμαστε ξεκάθαροι για το πως αυτά υπολογίστηκαν, και ο χρήστης από τη μεριά του θα πρέπει να έχει αρκετά βαθιά γνώση του συστήματος ώστε να κατανοήσει τις εξηγήσεις.

Συνοψίζοντας, σε αυτό το κεφάλαιο μελετάμε το πρόβλημα της αναγνώρισης COVID-19 από ηχητικά αρχεία βήχα, ομιλίας και αναπνοής, δίνοντας έμφαση στην ερμηνευσιμότητα. Παρουσιάζουμε μια απλή αλλά καινοτόμα για το πρόβλμα προσέγγιση, η οποία περιλαμβάνει συνελικτικά δίκτυα μικρού βάθους και με μικρό δεκτικό πεδίο, τα οποία εκπαιδεύονται με ασθενή επίβλεψη. Η προσέγγιση αυτή κέρδισε ένα σχετικό διαγωνισμό στον οποίο συμμετείχαν διάφοροι ερευνητές, πετυχαίνοντας τα καλύτερα αποτελέσματα, και όντας σε κάποιο βαθμό εγγενώς εξηγήσιμη, χάρη κυρίως στην απλότητα της προσέγγισης. Στη συνέχεια, περιγράφουμε τη διαδικασία κατασκευής του συνόλου δεδομένων εξήγησης smarty4covid, δίνοντας έμφαση στον ορισμό της ορολογίας και το σχεδιασμό και την κατσκευή της βάσης γνώσης. Αξιοποιώντας το σύνολο αυτό, μέσω του θεωρητικού πλαισίου που παρουσιάζεται στο πρώτο κεφάλαιο της διατριβής, παράγουμε σημασιολογικές εξηγήσεις, οι οποίες μας φανερώνουν κάποιες επιβλαβείς πολώσεις που είχαν υιοθετήσει τα συνελικτικά δίκτυα του διαγωνισμού, κυρίως σε ό,τι αφορά τις ηλικιακές ομάδες και το φύλο. Τέλος, μελετάμε και την προσέγγιση των εγγενώς ερμηνεύσιμων απλών ταξινομητών που ακολουθούν μια διαδικασία εξαγωγής χαρακτηριστικών. Αυτή η προσέγγιση, ενώ οδήγησε στα καλύτερα αποτελέσματα ως προς την επίδοση ταξινόμησης, φανέρωσε κάποια ζητήματα της εγγενώς εξηγησιμότητας τα οποία δεν έχουν συζητηθεί εκτενώς από την ερευνητική κοινότητα, όπως είναι η σημαντικότητα της ορολογίας που χρησιμοποιείται για την κατανόηση των εξηγήσεων, καθώς και η καλή γνώση του υποκείμενου μοντέλου και της μεθοδολογίας για τον υπολογισμό εξηγήσεων, ώστε ο χρήστης να μην παραπλανείται από υπεραπλουστεύσεις όπως είναι η "σημαντικότητα χαρακτηριστικών".

Συμπεράσματα

Στην παρούσα διατριβή μελετάμε τα προβλήματα της αξιολόγησης, της ερμηνείας, της εξηγησιμότητας και της εποπτείας των αδιαφανών συστημάτων μηχανικής μάθησης. Η προσέγγισή μας για την επίλυση αυτών των προβλημάτων βασίζεται στην αξιοποίηση πλούσιας ορολογίας η οποία είναι αναπαραστημένη σε γράφους γνώσης. Συγκεκριμένα, στο πρώτο κεφάλαιο της διατριβής παρουσιάζεται ένα καινοτόμο θεωρητικό πλαίσιο, βασισμένο στο φορμαλισμό των περιγραφικών λογικών, στο οποίο ορίζονται εξηγήσεις σε μορφή κανόνων και σε μορφή αντιπαραθέσεων, οι οποίες είναι εκφρασμένες με συγκεκριμένη ορολογία, η σημασιολογία της οποίας είναι ορισμένη με μαθηματική αυστηρότητα σε μια υποκείμενη βάση γνώσης περιγραφικών λογικών. Το θεωρητικό πλαίσιο αυτό το εφαρμόζουμε εκτενώς για να εξηγήσουμε και να αξιολογήσουμε ταξινομητές εικόνας, εκτελώντας διεξοδικά ποιοτικές και ποσοτικές αξιολογήσεις, καθώς και συγκρίσεις με άλλες μεθόδους από τη βιβλιογραφία. Υπό το πρίσμα των βασικών ιδεών που βρίσκονται στον πυρήνα της διατριβής, που αφορούν τη σημαντικότητα της ορολογίας και της σημασιολογίας σε ό,τι αφορά την εξηγισιμότα, μελετάμε επίσης το πεδίο των συμβολικών αναπαραστάσεων μουσικής. Συγκεκριμένα αναπτύσσουμε μια μεθοδολογία για την αξιολόγηση συστημάτων αυτόματης σύνθεσης, βασισμένη σε ιδέες από τη θεωρία της μουσικής, όπως είναι ο κύκλος των πεμπτών, και αναδεικνύουμε τη χρησιμότητα ανάπτυξης μεθόδων αξιολόγησης βασισμένες σε τυπικά ορισμένη θεωρία. Επιπρόσθετα, μελετάμε το πρόβλημα της αναγνώρισης είδους μουσικής από συμβολικές αναπαραστάσεις, αναπτύσσοντας μια προσέγγιση η οποία ξεπέρασε σε επίδοση τις υπάρχουσες της βιβλιογραφίας. Μιας και η δική μας προσέγγιση δεν είναι εγγενώς εξηγήσιμη, όπως είναι

αυτές της βιβλιογραφίας, γεγονός το οποίο είναι δύσκολο να ποσοτικοποιηθεί, μελετήσαμε τις *post hoc* μεθόδους για την ερμηνευσιμότητα της προσέγγισής μας. Από αυτές τις δοκιμές φάνηκε πως δεν είναι τόσο εύκολο να ανακτηθεί η εξηγισιμότητα, ειδικά σε ένα τέτοιο πεδίο όπως είναι αυτό της συμβολικής μουσικής, για την οποία δεν έχουν αναπτυχθεί ή προσαρμωστεί συγκεκριμένες μέθοδοι. Το κύριο ζήτημα που οδηγεί στην αποτυχία των αλγορίθμων εξήγησης είναι η αναπαράσταση των δεδομένων (πίνακες pianoroll) και η ορολογία που χρησιμοποιείται, η οποία δεν είναι κατανοητή από ανθρώπους. Σε αυτά τα πλαίσια, εφαρμόσαμε τις ιδέες του θεωρητικού πλαισίου του πρώτου κεφαλαίου και καταφέραμε να παράξουμε εξηγήσεις οι οποίες σαν ορολογία χρησιμοποιούσαν μουσικά διαστήματα, αντί για στοιχεία του πίνακα pianoroll. Τέλος μελετάμε το πρόβλημα της πρόβλεψης COVID-19 από ηχητικά αρχεία, υπό το ίδιο πρίσμα, το οποίο περετέρω στηρίζει τα επιχειρήματά μας που αφορούν την ορολογία στις εξηγήσεις, για παράδειγμα μέσω των σημασιολογικών εξηγήσεων του ταξινομητή του διαγωνισμού, οι οποίες ανέδειξαν πως ο ταξινομητής ήταν πολωμένος ως προς το φύλο και τις ηλικιακές ομάδες. Επιπρόσθετα, για το πρόβλμα αυτό συγκρίνουμε ποιοτικά και διάφορες εγγενώς εξηγίσιμες προσεγγίσεις μηχανικής μάθησης, βασισμένες κυρίως σε εξαγωγή χαρακτηριστικών και τη χρήση απλών ταξινομητών για την επίλυση του προβλήματος. Με αυτόν τον τρόπο αναδείξαμε περετέρω θέματα της εξηγησιμότητας, όπως είναι πολλές φορές η ανάγκη ο χρήστης να έχει γνώση του πως λειτουργεί το υποκείμενο μοντέλο, πως έχουν εξαχθεί τα χαρακτηριστικά, και το πως είναι ορισμένες οι ίδιες οι εξηγήσεις.

Το κυριότερο συμπέρασμα της διατριβής είναι η ζωτικής σημασίας σημαντικότητα της ορολογίας σε ό,τι αφορά την εξηγισιμότητα. Μέσα από τα πειράματά μας δείχνουμε πόσο εύκολο είναι κανείς να παραπλανηθεί από τις πληροφορίες που περιέχονται μέσα σε "εξηγήσεις" και τονίζουμε πως αν αυτές οι πληροφορίες είναι εκφρασμένες με καλά ορισμένη ορολογία, "γειωμένες" σε μια βάση γνώσης, τότε ξεπερνώνται κάποια από τα προβλήματα και τις δυσκολίες της περιοχής. Επιπρόσθετα, παρόμοια συμπεράσματα εξάγουμε και για τις μεθόδους αξιολόγησης, κυρίως εκείνες για τη στοχευμένη αξιολόγηση ως προς συγκεκριμένα χαρακτηριστικά των συστημάτων (όπως για παράδειγμα ο εντοπισμός αν αυτά έχουν υιοθετήσει επιβλαβή στερεότυπα), στις οποίες μπορεί να είναι πολύ χρήσιμο να περιγράφει κανείς τα δεδομένα του σε διαφορετικό επίπεδο αφαίρεσης από εκείνο που λαμβάνει στην είσοδό του το μαύρο κουτί. Δεδομένων των συμπερασμάτων αυτών, αναδεικνύεται επίσης η ανάγκη για την ανάπτυξη συνόλων δεδομένων εξήγησης, όπως αυτά είναι ορισμένα στο πρώτο κεφάλαιο, διαδικασία η οποία μπορεί σε περιπτώσεις να είναι ακριβή σε πόρους, όπως η διαδικασία που ακολουθήσαμε για την κατασκευή της βάσης γνώσης smarty4covid. Τέλος, καθώς η ερευνητική περιοχή της ερμηνεύσιμης τεχνητής νοημοσύνης είναι ακόμα ρευστή, και τα δεδομένα αλλάζουν συνεχώς, θα πρέπει να να είμαστε κριτικοί, αλλά και ανοιχτόμυαλοι σε ό,τι αφορά την έρευνα στην εξηγήσιμη τεχνητή νοημοσύνη, και νέες μεθόδους που προκύπτουν, και αυτό συμπεριλαμβάνει και την παρούσα διατριβή.

Μελλοντικές επεκτάσεις

Η προτεινόμενη προσέγγιση στην παρούσα διατριβή είναι αρκετά γενική ώστε να υπάρχουν πολλές κατευθύνσεις προς την οποία θα μπορούσε να επεκταθεί. Αρχικά, σε ό,τι αφορά το θεωρητικό πλαίσιο, αυτό μπορεί να εμπλουτιστεί περετέρω με άλλες μορφές εξηγήσεων πέραν των κανόνων και των αντιπαραθέσεων, όπως είναι οι εξηγήσεις σε μορφή χαρακτηριστικών παραδειγμάτων. Επιπρόσθετα, στα πλαίσια της διατριβής δεν εμβαθύναμε στη βελτιστοποίηση των αλγορίθμων, και καθώς τα προβλήματα είναι πολύ ακριβά υπολογιστικά, χρειάζεται εκτενής μελέτη για την ανάπτυξη νέων αλγορίθμων για τον υπολογισμό σημασιολογικών εξηγήσεων, καθώς και οι επέκτασή τους για πιο εκφραστικές γνώσεις, οι οποίες περιλαμβάνουν για παράδειγμα συνεχείς αριθμητικές τιμές.

Πέρα από το θεωρητικό κομμάτι, για να εφαρμοστούν οι προτεινόμενες μέθοδοι στην πράξη, είναι απαραίτητη η ανάπτυξη συνόλων δεδομένων εξήγησης, περιλαμβάνοντας αυστηρό ορισμό της ορολογίας, και τυπική αναπαράσταση της γνώσης. Έχοντας εμβαθύνει στο πεδίο των συμβολικών αναπαραστάσεων μουσικής, ένα κομμάτι της μελλοντικής μας έρευνας αφορά την κωδικοποίηση εννοιών από τη μουσική θεωρία, τον χαρακτηρισμό μουσικής με αυτές, και την μετ'έπειτα

ανάπτυξη συνόλων δεδομένων εξήγησης. Ένα παράδειγμα ενός τέτοιου συνόλου θα ήταν κομμάτια μουσικής, τα οποία συνοδεύονται από τις αλληλουχίες συγχορδιών τους, όπου η σημασιολογία των συγχορδιών είναι ορισμένη στην υπάρχουσα οντολογία λειτουργικής αρμονίας. Στο κομμάτι της μουσικής μελετάμε επίσης και άλλα ζητήματα, όπως είναι η εξηγήσιμη σύνθεση μουσικής, και η δημιουργική εφαρμογή των μεθόδων που αναπτύξαμε (για παράδειγμα αντιπαραθέσεις για τη στοχευμένη τροποποίηση ενός κομματιού μουσικής).

Σε ό,τι αφορά την έρευνά μας για την αναγνώριση COVID-19 από ηχητικά αρχεία, μελετάμε τρόπους για την καλύτερη αξιοποίηση της ετερογενούς πληροφορίας που φέρει ο γράφος γνώσης. Για παράδειγμα θα μπορούσαμε στις εξηγήσεις να δίνουμε μεγαλύτερη έμφαση σε πληροφορία η οποία έχει προκύψει από χαρακτηρισμούς ειδικών, παρά από αυτοαναφορές των χρηστών. Επίσης υπάρχουν αντίστοιχα σύνολα δεδομέ- νων για άλλες ασθένειες, και με διαφορετικούς τύπους δεδομένων, όπως είναι οι εικόνες ακτινογραφιών, στις οποίες θα μπορούσαμε να εφαρμόσουμε τις ιδέες μας για την εξηγησιμότητα και την αξιολόγηση.

Chapter 1

Introduction

The main focus of our research has been the utilization of knowledge graphs for explaining and evaluating opaque Artificial Intelligence (AI). We developed a framework for computing rule-based, and counterfactual explanations of machine learning classifiers using knowledge graphs (chapter 3), allowing for explanations at different levels of abstractions, using terminology tailored to specific use-cases. We applied this framework on multiple domains, including image classifiers (chapter 4), where high level conceptual information is used instead of pixels to generate human-understandable explanations. Furthermore, motivated by our research in the domain of music (chapter 5), in which evaluation is often a difficult task due to subjectivity, while explainability has the potential to broaden the applicability of machine learning systems, we utilized our ideas for evaluating symbolic music generation systems in addition to developing an approach for genre classification, comparing with existing approaches, and focusing on explainability. Finally, we applied our ideas on a real-world, decision-critical application, by utilizing the proposed explainability framework for explaining audio classifiers that automatically diagnose COVID-19 given cough, speech and breath audio of a user (chapter 6), and comparing it with other explainable AI approaches.

In recent years, AI has progressed explosively, with thousands of papers being published daily and new applications and use-cases being constantly proposed. This progress is mainly fueled by the advancements in the field of Machine Learning (ML), and in particular that of Deep Learning [113, 68]. Despite their apparent success in solving problems across domains, deep learning models suffer drawbacks that make it difficult for them to be applied in various real-world settings. At the core of these drawbacks is the inherent opacity of deep learning models, stemming from their structural complexity, it is very difficult to *explain* their output in a given setting. These drawbacks have given rise to ethical and legal concerns [69] regarding the use of opaque AI in everyday applications, and have led to the emergence of the eXplainable AI (XAI) field of research [192, 77]. The target of XAI is the end-user who depends on the decisions of AI models. In this context, researchers are exploring a wide range of questions, such as: What is an explanation? How can we develop models that are inherently transparent without sacrificing performance? How can we convincingly explain a black-box model's predictions? How can we make sure that explanations are accurate, useful and not misleading? As research progresses, we are getting closer to answering such questions [143, 75], however most remain without a convincing answer.

In this rapidly growing research area, new issues have emerged, some leading to important arguments against using *post hoc* XAI for explaining black-boxes, instead of using models that are interpretable in the first place [172]. However, black-box models are essentially the only solution for a multitude of tasks ranging from image classification, to chat bots. In addition, even transparent models used in industry are obfuscated as being proprietary, and can only be treated as black-boxes. Thus the problem of *post hoc* explainability remains an important one, and researchers should take under consideration the criticism when proposing new ideas. A particular set of XAI approaches which has emerged as "promising", and avoids many of the pitfalls of *post hoc* XAI, involves the utilization of knowledge graphs (KG) [94, 191]. These provide a structured representation of information which is based on the way humans perceive the world, that is typically both machine-readable and human-understandable. There is also a vast body of theoretical work concerning knowledge graphs, such as the framework of description logics, that provides theoretical guarantees and rea-

soning capabilities for logical entailment, both useful tools in the context of XAI. Even using simple knowledge, such as for instance hierarchies of concepts, can boost transparency of a system and in some cases even performance, as shown in our work in [43], or more informative and transparent evaluation of AI models, as shown in our work in [129].

1.1 Thesis overview

The main matter of the thesis is split into four chapters. Chapter 3 introduces the theoretical framework and the explanation generation algorithms we developed, while the following chapters showcase the application of the framework in different domains. In particular, in Chapter 4 we experiment explaining image classifiers by utilizing existing tools and available knowledge, in Chapter 5 we broaden our scope and apply our ideas for explaining and evaluating systems in the domain of symbolic music, while in Chapter 6 we develop all necessary resources for explaining and evaluating audio classifiers for COVID-19 detection.

1.1.1 Explanations in Terms of Knowledge

The developement of our framework for *post hoc* explainability, was introduced in [42], and was motivated by the need to provide explanations at a level of abstraction that is meaningful to humans, and using terminology that is understandable. As AI is now applicable in most domains and scientific areas, there is a greater need than ever for interdisciplinary work. When it comes to explainability, there is a vast body of work from philosophy, cognitive and social sciences that attempts to formalize, define, and study what is a "good" explanation for humans, on which computer scientists, and in particular XAI researchers can build on [138, 21]. A motivating example for our work, inspired by [84] is the following. Consider an explanation for a car accident. For a car mechanic such an explanation might be "the car accident happened because the breaks were not in working order". However the same explanation for the same car accident would not be meaningful to a civil engineer, who might expect something along the lines of "The car accident happened because the stop sign was not clearly visible from the road". It is clear that different users expect explanations at different levels of abstractions, and they understand different terminology. Our proposed framework attempts to tackle this exact issue.

The only possible interaction with a black-box model is to feed it data and to then observe its output, and most *post hoc* explanation pipelines follow the illustration shown in figure 1.1, where the *explainer* operates in the feature domain of the classifier. The issue here is that low-level feature representations, such pixels or audio signals, are not necessarily understandable and meaningful to humans. To mitigate this problem, in our framework [42, 59, 121, 41], we propose using data for which we have available information in two representations: one suited for the black-box, and one suited for explanations. A typical *post hoc* XAI pipeline within this framework would follow the illustration shown in figure 1.2. By using information at different levels of abstraction we can have more control over the generated explanations, their form, the terminology they use and the information they provide, and we are not constrained by what the black-box expects at its input. For example, consider a classifier that provides a risk assessment, or a diagnosis given audio of a person's cough [181]. Traditional XAI methods would provide explanations in terms of the input of the classifier (typically spectral representations of the audio). If, however, we had available additional information about the data, such as for example the age, or the sex of the person in the recording, then we could provide explanations in terms of this information. This way, it would be easy to uncover specific biases of the black-box classifier, which would be obscured if the only information we had available were the spectral representations. Taking this idea further, consider we have available a dataset of audio of coughs which has been extensively annotated by medical professionals, using standardized medical terminology. Then, explanations can be provided, that use as a vocabulary standardized medical terminology, and a XAI procedure could make use of the

relationships between medical terms, as they are defined in knowledge graphs such as SNOMED-CT [50] and ICD-10 [92]. This can be useful in many ways, including improving understandability, especially when the black-box input representation contains low level sub-symbolic information (such as pixels or samples of a digital audio signal).

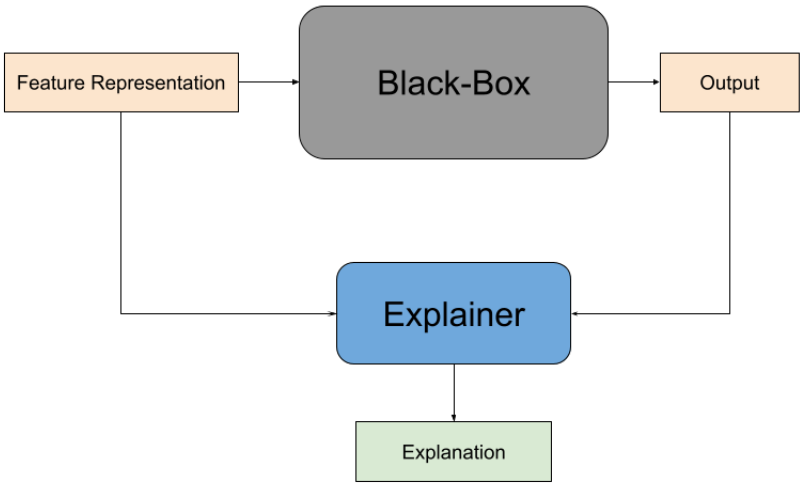


Figure 1.1: Typical *post hoc* XAI pipeline

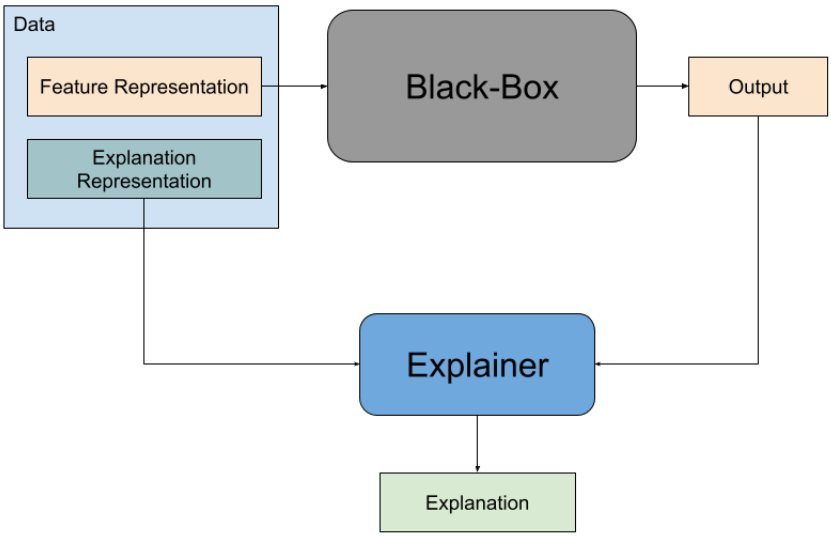


Figure 1.2: Proposed *post hoc* XAI pipeline

Given the idea of using different representations of a data samples for explanations, than what the black-box expects at its input, the question arises of *what representation should be used*? There are many different options, depending on the type of explanation required, the domain of data and the application. For example in [40] we used a vector representation of musical intervals for explaining a MIDI genre recognition neural network, after modifying the GPX [57] algorithm to be compatible with this framework. In another example [59] we used sets of objects depicted in images and in terms of these provided counterfactual explanations. In the formal description of the

framework introduced in [42], the additional representation is provided in the form of a *semantic description*. Specifically provided a mapping of each item to an individual name from a given vocabulary representing an individual from a given description logics knowledge base, we can provide explanations in terms of the vocabulary of the knowledge, by leveraging theoretical results from the area of semantic query answering.

Choosing a different representations of data for providing explanations, also effects the applicability of different algorithms, and their scalability. For example, using the mapping to individual names as a representation suited for explanations, the problem of finding rules which mimic the behaviour of the classifier, which is a form of explanations, is reduced to a semantic query reverse engineering problem which is known to be very difficult to solve [6, 155, 46, 28]. To overcome this difficulty, we can either use simpler representations, such as sets (similarly to our work in [59]) or vectors (similarly to our modification of GPX in [40]), or to *approximate* the solution, for which we developed heuristic algorithms [122, 121]. Thus the choice of explanation representation, and explanation algorithm heavily depends on the application, and available data, and different approaches can uncover different aspects of the black-box under investigation.

1.1.2 Semantic Explanations of Image Classifiers

Image classification is one of the most intuitive tasks for humans, and one of the first to be taken over by deep learning, by use of deep Convolutional Neural Networks (CNN) [110]. The process of human visual perception is intricate and encompasses the binding of spatial, structural, and semantic information [189]. On the other hand, machine vision encounters challenges in capturing high-level conceptual and semantic information with the same ease as humans. Unlike the human visual system, which effortlessly grasps complex scenes and recognizes objects based on their context, machines primarily rely on low-level information such as pixels, and automatic pattern recognition, extrapolating to higher level concepts by means of training on large datasets. This is an issue for explainability, as there is a missing link between semantic understanding, the ability to infer meaning and context from images, and the complex inner workings of systems that extrapolate meaning from pixel values.

A classic example of pixel-based explanations are saliency maps [182]. The result of such methods is typically a set of values assigned to each pixel, representing the importance of each pixel for a specific prediction, or for a specific class. When it comes to explainability, such methods have been shown to often be misleading [2]. A good example from [172] is shown in figure 1.3. If a human was shown the evidence for the image being classified as a Siberian Husky, they might be convinced, as the highlighted pixels make sense as being indicative of the particular dog breed. However, the explanation stops making sense when we observe the second saliency map, showing evidence for the image depicting a transverse flute. Such explainability methods do actually show what pixels are important for a prediction, but they do not show *why*. For instance a classifier might always consider the pixels in the center of the image as being important, regardless of what they depict. In these cases humans might tend to fill in the gaps, and end up being misled pertaining to why the classifier actually made its prediction.

These issues have long been recognized by the computer vision and XAI communities, and other approaches have been proposed for explaining image classification. One such approach involves *counterfactual explanations*, where instead of pixel importances, explanations have the form of pixel edits. For instance, in [72], the explanations indicate a specific region of an image, and how it should change (based on a region from a different image) in order for the classification to change. As such counterfactual methods can often end up with infeasible, or nonesense explanations, further work has attempted to enforce constraints, such that the explanations are semantically consistent [196]. The importance of semantics in explainability of image classifiers is also highlighted in concept attribution methods [62, 206]. These, especially useful in a *global* setting, extract human-defined concepts, and measure their importance for the classifier under investigation making them more understandable and less prone to be misleading. In the husky example (figure 1.3), such an explana-

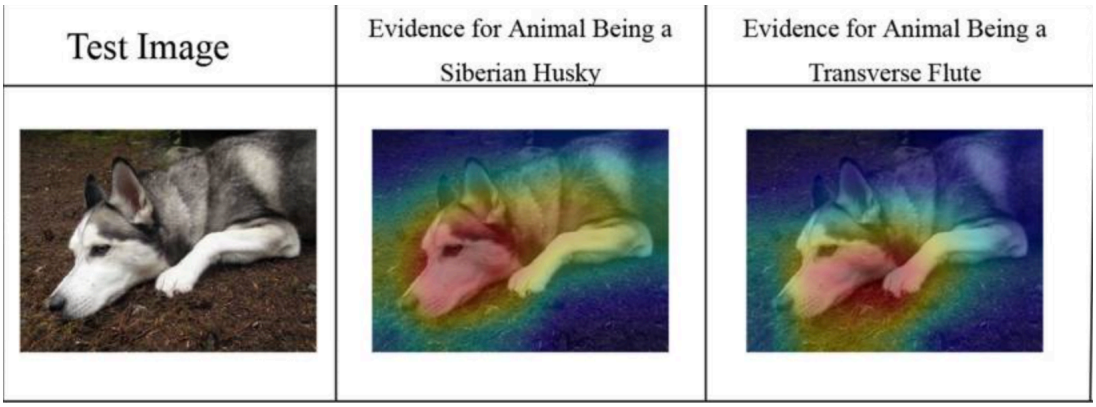


Figure 1.3: Example of misleading pixel importance explanation. Taken from [172]

.

tion might include concepts such as the muzzle of the dog, fur, color etc. However these methods often suffer from other drawbacks, such as their requirement for white-box access to the classifier, which is often not possible, as is the case of proprietary models, where one could argue that explainability is especially important.

In our work, we applied the proposed knowledge-based explainability framework for generating multiple types of explanations, including rule-based, counterfactual, local and global. Importantly, without requiring white-box access to the model under investigation, we utilize semantic descriptions of images in order to better understand what the classifier under investigation is doing, by only observing sets of input-output pairs. In particular, we show that within our framework we can generate explanations that are more *semantically consistent* to humans than the related work [41], and we argue that by using well defined terminology the explanations are more understandable. We also showcase how the proposed framework can be used for detecting biases, and debugging black-box classifiers.

1.1.3 Explainability and Evaluation of AI in the Domain of Symbolic Music

The second domain we tackled with the idea of utilizing knowledge and semantics for explanainability and evaluation was the domain of symbolic music. Specifically, while exploring applications involving symbolic representations of music, with emphasis on automatic music composition, we quickly realized that the bottleneck obstructing progress in this area is the lack of trustworthy and objective evaluation procedures, thus making it impossible to compare different approaches to automatic music generation. In our work in [38] we proposed a framework which facilitates the development of evaluation metrics for music, based on music-theoretical concepts such as the circle of fifths and the Tonnetz [194]. We showed that metrics defined within this framework can be used heuristically to determine if music represented symbolically was likely composed by a human or a machine. This is an example of how knowledge graphs (such as the circle of fifths), that encode domain knowledge (such as music theory), can be used for evaluating AI, in addition to explaining it.

Furthermore, in our work in [39] we were researching the merits of specialized neural architectures for symbolic music, and achieved promising results, surpassing the state-of-the-art for genre recognition based on F1 score and the area under the receiver operating characteristic curve. However, we realized that these metrics, even though objective, do not tell the whole story, and we hypothesize that a musician or another end-user would probably prefer to use the genre recognition approach presented by Ferraro and Lemstrom in [55], since it is based on extracted musical patterns and is thus more transparent than our deep learning based approach, even though our approach achieves better performance based on the metrics. For this task specifically, explainability is especially important, as there are known issues of ground truth reliability, and ill defined genres.

For example the notion of "Pop" music has drastically changed over the past decades, and given a black-box classifier that classifies a song as being "Pop", we should be able to understand what the black-box considers to be "Pop", in order to use it in a real-world setting. Motivated by this, we then extended our work and studied *post hoc* explainability techniques which could be applied to explain the predictions of our neural networks, for our deep learning based approach to be compared with the inherently transparent one presented by Ferraro and Lemstrom, with mixed results [40]. Specifically, by experimenting with a multitude of different explainability methods, we determined that in many cases the explanations were not understandable, not consistent, and could be misleading. We also applied our framework, in which explanations are provided across different levels of abstraction (for example musical intervals instead of musical notes), and show how some of the afforementioned pitfalls can sometimes be avoided.

Our hypothesis about the importance of transparency of AI in the domain of music was validated to an extent by professional musicians in our work in [58], in which we developed a methodology to automate musical effects via brain-waves of a performer in a live setting. Especially in this context of performer-computer interaction, we saw that musicians want to know, to an extent, how the AI will react to their actions. This research motivated us to further explore *post hoc* explainability that utilizes different levels of abstraction, in which underlying AI models are treated as black-boxes. To this end, as the proposed framework can make use of complex semantics defined in description logics knowledge bases, we also developed the musical harmony ontology [99] which represents the theory of tonal and modal harmony as a knowledge graph, which is a more rich representation than what had been previously used, and we are in the process of extending this knowledge with more music-theoretical notions (such as for example the structure of music in the music part ontology). Our ongoing research involves utilizing this represented knowledge for the purpose of explainability, similarly to how we utilized higher level information to explain genre classification [40], or how we utilized more complex knowledge graphs for explaining image classifiers.

1.1.4 Explainability for COVID-19 audio classification

In the final chapter of the main matter of this dissertation, we sought to validate our proposed explainability framework in a real world, decision critical application. Specifically, motivated by the COVID-19 pandemic and the challenges it brought forth, an area of research showing promising results was explored by the community, in which assessments of COVID-19 infection are computed given audio of people's coughing, breathing or speech [209, 148, 20, 25, 29, 181]. We took part in the efforts to explore this approach, by our winning entry [1] at the IEEE COVID-19 sensor informatics challenge hackathon [2], in which one of the evaluation criteria was explainability. Using methods from related literature we were able to explain (to an extent) the predictions of our classifier. However as these "traditional" explainability methods operate in the feature space of the classifier, they suffer from aforementioned pitfalls that similar methods do, such as for example being expressed in terms of a spectral representation of the audio, that is not necessarily understandable to humans.

The limiting factor for applying our knowledge-based explainability framework is its reliance on the existence of well-defined structured knowledge. For this reason, in the context of the smarty4covid project [3] in which we developed a crowd-sourced dataset containing audio of coughs, speech and breathing, in addition to information such as demographics, symptoms, preexisting conditions, and expert annotations we invested in developing an accompanying knowledge base [219], in which all information gathered by crowd-sourcing and labeling procedures was represented using the web ontology language (OWL) [4], and the formalism of Description Logics (DLs) [9]. Such a knowledge base is required for applying the proposed framework, and even though its development is a resource intensive procedure, requiring significant human effort, we argue that in some cases it is

[1] https://ieee-dataport.org/analysis/ntuautn-ieee-covid-19-sensor-informatics-challenge

[2] https://healthcaresummit.ieee.org/data-hackathon/

[3] https://www.smarty4covid.org/

[4] https://www.w3.org/OWL/

worth it.

To show this, we utilized the smarty4covid knowledge base, alongside deep learning models, for attempting to solve the task of COVID-19 assessment from audio. From our experiments, we realized that many available models, including our winning entry from the hackathon, do not generalize well, and their performance can vary significantly between datasets. Using traditional explainability methods, the reasons for such discrepancies were not clear. However, by using the terminology defined in the smarty4covid knowledge base, and by applying our general idea of explaining classifiers at a different level of abstraction than the feature space, we were able to gain insights about the problem, that would otherwise be difficult, or even impossible to detect using traditional methods.

Chapter 2

Background Material

In this chapter we introduce background material relating to explainable AI, knowledge representation, music, and deep learning, focusing on higher level concepts that are important throughout the dissertation. Where necessary, additional background material is provided at the beginning of each chapter.

2.1 Explainable AI

Explainable AI (XAI) is not a new area of research [178], but has gained popularity in recent years due to the problem of opacity of deep learning models. As there are constantly a multitude of advances and new propositions from the research community, XAI is constantly evolving, and in our opinion has not yet reached maturity. Nonetheless, there have been attempts at surveying the area, and defining terminologies, taxonomies, methodological approaches, and challenges [74, 112, 212, 78, 8]. In this section we discuss how this dissertation fits in the broader context of XAI.

Interpretability and Explainability An example that indicates the immaturity of the field, is that there is not an agreed upon distinction between the terms "interpretability" and "explainability". In this work we often use the words interchangeably. Their difference in our view is that "interpretability" refers to the action of the user, who given information, attempts interpret it, while "explainability" refers to the action of the system, which is tasked with providing an explanation. For example, if a user, by viewing the coefficients of a linear model, is able to exhume feature importances, then they would be interpreting the model, and we would consider this to be "interpretability". Contrarily, if a system were to sort the coefficients by absolute value, and show the top-3 to a user as being important, then we would consider this to be "explainability", but these definitions are used loosely.

Post hoc* and *ante hoc A first distinction of explainability approaches is whether the explanations are generated after the model under investigation has been developed and trained (*post hoc*), or if the model was designed in a way that it is interpretable (*ante hoc*). The latter are often referred to as "inherently interpretable", or "transparent" models, and include methods such as decision trees, linear models, and even some specialized neural networks, while the former can be applied on any "opaque" model, such as a deep neural network. Many researchers argue that *post hoc* explainability is a flawed approach, and that we should focus on inherently interpretable solutions [172], instead of using deep learning for everything and attempting to solve the explainability problem after the fact. However, the undeniable performance supremacy of opaque models in specific domains (for example Natural Language Processing, Computer Vision), motivates research into *post hoc* explainability, which is a main focus of this dissertation. Our proposed framework in chapter 3, outlines some current issues with *post hoc* XAI, and how we attempt to tackle them.

Global and Local A second distinction of XAI methods, is whether they produce "Global" or "Local" explanations. The first provide information for explaining the general behaviour of a model to a user, while the second explain why a specific prediction was made on a specific data sample. Both

global and local explanations are useful, and each provides different insights into the model under investigation. For example, if an AI supported decision was accompanied by a *local* explanation, it would allow a domain expert to evaluate the decision, potentially uncovering new information, and ultimately making the decision themselves, which would be an ideal scenario for the medical domain. On the other hand, reliable *global* explanations would be imperative for regulation of AI in industry, potentially uncovering unwanted biases, and serving as a way of evaluating models across more dimensions besides typical performance metrics. Our proposed framework facilitates both global and local explanations, and throughout our experiments we often discuss and compare both approaches.

Black box and White box A third distinction of XAI approaches concern whether the explainer has access to all components of model under investigation, such as the weights of a neural network, or if the only access involves probing the model with inputs and observing outputs. The former, also referred to as "white box" or "model-specific" explanations, extract information from the inner mechanisms of a model in order to produce an explanation, and are able to provide meaningful information even for very complex models, such as concept based explanations for large transformers [97], or counterfactual explanations for image classifiers [196]. These white-box approaches, despite their effectiveness, are much less appealing than pure black-box approaches, as the latter can be applied to a wider variety of use-cases. Especially when the model is hidden, either as being proprietary, or even being maliciously obfuscated, the need for explainability is higher. It is however a much harder problem to solve. Throughout our work, the focus has been black-box explainability, where we only have access to input-output pairs for generating an explanation.

Features and Concepts A distinction that is not often discussed separately, concerns the *terminology* used in the explanations, and specifically if they use the features that the model uses, or if they use different terminology. This distinction is sometimes grouped with the broader classification of different forms of explanations, where approaches that use higher-level terminology are referred to as concept-based XAI. In our work, this distinction is crucial, and throughout the thesis we highlight the benefits of using higher level concepts in the explanation (such as objects depicted in an image), as opposed to features (such as pixels). The main issue with concept-based explanations is the analogue of the grounding problem [85], where we cannot know if the semantics of symbols (such as concepts) are the same for the model and for the user's mental model. For example the concept of a "Dog" is different in each person's mind, and it will also be different for an AI model. In our work, we attempt to *ground* our explanations on rigorously defined knowledge bases using the formalism of description logics, introduced in the next section.

Forms of explanations There is a large variety of forms of explanations in the context of XAI, ranging from a simple bar chart of feature importance, to natural language, and visual explanations. The form of explanations is crucial for understandability and informativeness, and there is not an agreed upon ideal form of an explanation. Some forms have been argued to be misleading, such as feature importance, as they do not explain *why* a feature is important, and the end-user might end up inferring incorrect information, while others have been argued to be aligned with the way humans explain things to one another, such as rule-based and counterfactual explanations. In our work, we experiment with multiple forms of explanations, but the proposed framework is tailored to generate rule-based and counterfactual ones. Finding the ideal form of explanations should be a result of interdisciplinary work with cognitive scientists, which is currently lacking in XAI [21], but we do believe that rule-based and counterfactual explanations provide a good foundation for human-understandability when compared to other forms.

2.2 Knowledge Representation

Representing knowledge in a way that is structured, easily accessible to humans and to machines, accurate, and unambiguous is very difficult in practice. One data structure that has prevailed for knowledge representation is the *knowledge graph* [94], where by appropriately labeling nodes and edges of a directed graph, complex information can be efficiently encoded in a structured way, including entities, relations, attributes, and semantics. Knowledge graphs are scalable, can be interlinked, and can be efficiently queried, thus they are useful resources for explainability [191].

Description Logics Description Logics (DLs) [9] provide a formal foundation for representing, and reasoning on knowledge. DLs define a set of *languages*, that include a vocabulary and constructors, that are used to define *knowledge bases*, that consist of axioms and assertions. Within this framework we can describe a *world* using the artefacts of a DL language to assert facts and define axioms, and then via reasoning algorithms we can infer new facts about the world, we can check if a piece of information is true, false or unknown, and we can perform *semantic query answering*. The framework of DLs allows for very expressive languages, that in turn allow for encoding, and reasoning on, complex ideas. However, in practice, utilizing very expressive knowledge is not feasible, as the complexity of reasoning algorithms becomes exponential or even undecidable. In this work, even though we do not make full use of the expressive power and reasoning capabilities of DLs, they are used as a way of future-proofing the proposed explainability and evaluation framework, which could be extended in the future for more expressive knowledge than what is presented here. Specifically, we make certain assumptions about the structure of DL knowledge bases, that mainly allow the information to be represented in finite directed labeled graphs. We give a short introduction to DLs as they are used in this work, while a more detailed description of the notation used within our proposed framework and algorithms is provided in section 3.2.

Given a vocabulary $\mathcal{V} = \langle \mathsf{CN}, \mathsf{RN}, \mathsf{IN} \rangle$ where $\mathsf{CN}, \mathsf{RN}, \mathsf{IN}$ are mutually disjoint finite sets of concept, role and individual names, we consider $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ to be a knowledge base, where the ABox $\mathcal{A}$ is a set of assertions of the form $C(a)$ and $r(a, b)$ where $C \in \mathsf{CN}$, $r \in \mathsf{RN}$ and $a, b \in \mathsf{IN}$, and the TBox $\mathcal{T}$ is a set of terminological axioms of the form $C \sqsubseteq D$ where $C, D \in \mathsf{CN}$ or $r \sqsubseteq s$ where $r, s \in \mathsf{RN}$. The symbol '$\sqsubseteq$ ' denotes inclusion or subsumption. For example, a concept name (in CN) could be Dog, an individual name (in IN) could be the (unique) name of a specific dog, for example $\mathsf{snoopy\_42}$, and a role name (in RN) could be a relation, such as "eating". Then an ABox could contain the assertion $\mathsf{Dog}(\mathsf{snoopy\_42})$, indicating that $\mathsf{snoopy\_42}$ is a Dog, and a TBox could contain the axiom $\mathsf{Dog} \sqsubseteq \mathsf{Animal}$, representing the fact that all dogs are animals (where Animal is also a concept name in CN). In such a knowledge base, both the ABox and the TBox can be represented as labeled graphs. An ABox $\mathcal{A}$ can be represented as the graph $\langle V, E, \ell_V, \ell_E \rangle$ (an *ABox graph*), where $V = \mathsf{IN}$ is the set of nodes, $E = \{\langle a, b \rangle \mid r(a, b) \in \mathcal{A}\} \subseteq \mathsf{IN} \times \mathsf{IN}$ is the set of labeled edges, $\ell_V : V \to 2^{\mathsf{CN}}$ with $\ell_V(a) = \{C \mid C(a) \in \mathcal{A}\}$ is the node labeling function, and $\ell_E : E \to 2^{\mathsf{RN}}$ with $\ell_E(a, b) = \{r \mid r(a, b) \in \mathcal{A}\}$ is the edge labeling function. A TBox $\mathcal{T}$ that only contains hierarchies of concepts and roles, can be represented as a directed graph $\langle V, E \rangle$ (a TBox graph) where $V = \mathsf{CN} \cup \mathsf{RN} \cup \{\top\}$ the set of nodes. The set of edges E contains an edge for each axiom in the TBox, in addition to edges from atoms appearing only on the right side of subsumption axioms, and atoms that don't appear in the TBox, to the $\top$ node: $E = \{\langle a, b \rangle \mid a \sqsubseteq b \in \mathcal{T}\} \cup \{\langle a, \top \rangle \mid c \sqsubseteq a \in \mathcal{T} \wedge a \sqsubseteq d \notin \mathcal{T} \wedge c, d \in \mathsf{CN} \cup \mathsf{RN}\} \cup \{\langle a, \top \rangle \mid a \notin sig(\mathcal{T})\}$. This is abusive notation, in that the symbol $\top$ is overloaded and symbolizes both the universal concept and the universal role.

Knowledge graphs in XAI On a high level knowledge graphs seem perfect for XAI applications, as the desiderata for good explanations, such as human-understandability, and semantic clarity, are inherent features of well-constructed knowledge graphs. Furthermore, the strengths and weaknesses of knowledge representation techniques seem to be complementary to those of machine learning, thus making hybrid approaches potentially the best of both worlds. Specifically, sub-symbolic in-

formation is prevalent in machine learning, which is data-driven, while knowledge-driven methods work by manipulating well-defined symbolic information. In addition, machine learning systems are often accompanied by uncertainty and fuzziness, as opposed to knowledge graphs, which cannot easily handle uncertainty, especially in cases that introduce inconsistencies, which have the potential to "break" the knowledge. Crucially, sub-symbolic machine learning is opaque, while symbolic knowledge-driven AI is transparent.

However, utilization of knowledge graphs for XAI is not straight-forward in practice, especially in the case of *post hoc* black box explainability. Most knowledge based XAI approaches are either *ante hoc*, where the models are designed to utilize knowledge during training (such as Deep Knowledge Aware Networks [201]), or white box, where the knowledge is extracted from inner layers of a neural network (such as concept based explanations for transformers [97]) [191]. Contrarily, a black box approach, having access only to input-output pairs, attempts to discover knowledge that adequately describes the model in a given context. For example, in [4], the authors use reasoning on geospatial knowledge for explaining the errors of a satellite image segmentation, resulting in explanations that use high level concepts, such as "Park", "Manmade Structure" or "Shadowy area", that are defined in an ontology, and end up describing errors of the image segmentation system. Our proposed framework, as a black box, *post hoc* approach, works among similar lines. The input-output pairs are mapped to a knowledge graph, where each is semantically described using appropriate terminology, as individuals in a DL knowledge base. Then, the explanations are expressed in terms of the knowledge, resulting for example in rules that explain the behaviour of the black box.

2.3 Music

Chapter 5 covers multiple different aspects of AI in the domain of symbolic music, from classification, generation, evaluation, and explainability. In this section we introduce some basic notions from music theory that are used throughout.

Music Theories There are numerous theories about different aspects of music, across different cultures, historical periods, genres, and musical instruments. All of them are descriptive, meaning that they are used to describe, *a posteriori*, what sounds good to humans, and are not meant to be used for creating music. This makes the utilization of music theoretical knowledge for explainability and evaluation an interesting problem, because we can explore how much specific aspects of music effect a model's behaviour, since a single piece of music can be described in many different ways.

Symbolic Music Representation In the context of 12-tone equal tempered western music, which is the most prevalent system in moden music, the most common digital symbolic representation is the MIDI protocol. For the purpose of this chapter, this representation is equivalent to the *pianoroll*, an example of which is shown in figure 2.1. This is an two-dimensional array, where the horizontal axis represents time, and the vertical axis represents pitch. Time is typically quantized in subdivisions of beats, meaning that the duration in seconds depends on the tempo of the music. Pitch on the other hand is quantized in semitones comprising of 12 pitch classes that repeat every octave (indicating a doubling in frequency). Different values in the array (indicated as different colors in figure 2.1) represent velocity - how loud the note is played. Such representations are abundantly available and much easier to handle than audio recordings of music, and they are perfectly suited for linking with knowledge representations of music theoretical notions.

Intervals and Harmony The music-theoretical notions we utilize mainly relate to musical harmony. These describe the effect that groups of notes have on the music when played simultaneously, or in succession. An interval is the distance between two notes in semitones, and there are music theoretical descriptions of intervals in a large variety of contexts. For example, there are intervals that themselves are considered dissonant, such as the semitone (interval 1) and the tritone (interval

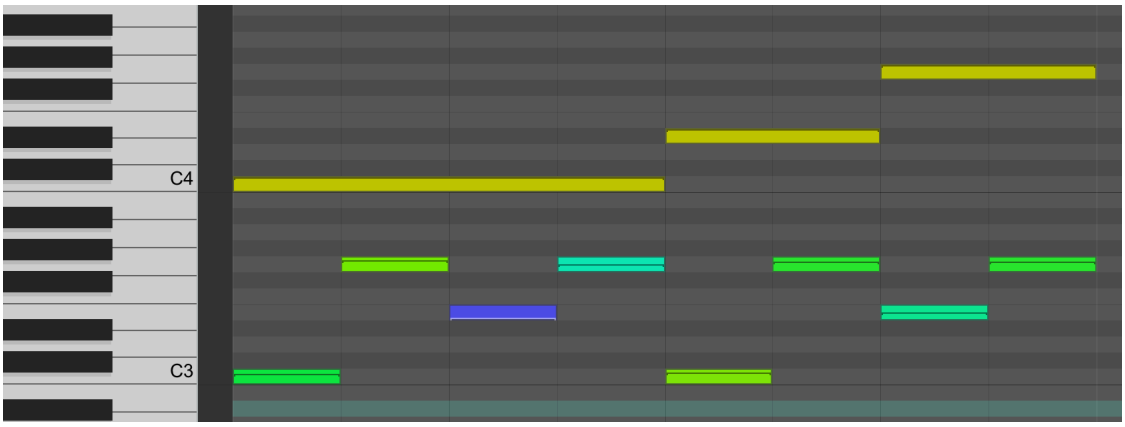


Figure 2.1: An example of a pianoroll.

6), consonant such as the perfect fifth (interval 7), joyous, such as the major third (interval 4), and sad, such as the minor third (interval 3). These descriptions become more complicated when considering groups of more than two notes, the broader context of what notes were playing before and what notes are to follow, and rhythmic characteristics, such as note duration. A crucial component of most ideas that describe groups of notes is the idea of a tonic note. The way humans perceive tonal sound, i.e. sound where a specific frequency is prevalent, is by isolating the prevalent frequency as the pitch, and perceiving all other frequencies in relation to the prevalent pitch, typically as timbre. Similarly, when listening to music, we perceive pitch in relation to multiple broader contexts, each of which defines a tonic note. For example, when listening to the group of notes G,B,D, if the note G is perceived as the tonic (for example by being the lowest frequency note), then we perceive a "joyous" G major chord. If in the broader context the perceived tonic is the note C (for example the piece of music being written in the key of C major), then we perceive the G major chord as building tension which we expect to be resolved, contrarily to if the context implied a G major key.

2.4 One-dimensional Convolutional Neural Networks

Part of our research involved developing custom convolutional neural networks (Sections 5.3.2, 6.2), and our motivation when developing these focused on aspects such as receptive field, and trainable parameters. In this section we give a brief introduction to one-dimensional CNNs. For further reading we refer to [68].

A one-dimensional convolution with kernels of size k is an operation, which acts on a sequence of T vectors X of size N to produce a sequence of vectors Y, where at timestep i and channel j:

$$Y_{i,j} = \sigma(\sum_{n=1}^{N}\sum_{m=1}^{k} W_{j,n,m} X_{i+m-1,n}) \tag{2.1}$$

The matrix W consists of the convolution's trainable parameters, while the function σ enforces non-linearity. Note that there is also a bias term that has been omitted. Each element of the output sequence Y only depends on k consecutive elements of the input sequence X, leading to the effectiveness of the convolutional operation for capturing local structures and patterns in the data. A deep neural network may then be constructed by stacking such operations in a depth-wise fashion. This results in the first convolutional operations capturing low-level features in the data, while deeper operations capture more complex high-level features.

Receptive Field and Trainable Parameters

An important attribute of such a network is its receptive field, which is defined as the number of elements in the input sequence that affect a single element of the output sequence. A single

convolution C_1 with kernels of size k_1 has a receptive field of k_1. A second convolution with kernels of size k_2 which is fed C_1's output will depend on k_2 consecutive elements of said output, leading to its dependence on $k_2 + k_1 - 1$ elements of the original input sequence. Another important attribute of a CNN to keep track of is the number of trainable parameters, equivalent to the size of all weight matrices W. The number of trainable parameters is the primary factor for the memory requirements of a network which is often a bottleneck for network design.

Efficiently increasing a network's receptive field is crucial for effectively capturing features across multiple time scales. By stacking convolutional layers, receptive field increases linearly with network depth and with kernel size k. However, increasing depth could give rise to difficulties during training such as the exploding and vanishing gradients problem (EVGP)[82][210] among others in addition to increasing the number of trainable parameters, while increasing k by dk leads to a $f_{in} \times f_{out} \times dk$ increase in trainable parameters, where f_{in} and f_{out} are the numbers of input features and output features (number of kernels in the layer). There exist many methods for increasing a network's receptive field more efficiently, for instance using dilated convolutions such as in [146], using strided convolutions, or the most common approach: pooling layers.

Pooling

A pooling layer of stride S and kernel size K acts on a sequence of vectors X of dimensions $T \times N$ and outputs a sequence Y of dimensions $(\frac{T-K}{S} + 1) \times N$. The stride parameter S determines how many input samples are skipped in between applications of the pooling kernel. A common pooling operation is max pooling with a stride equal to kernel size $K = S$. In this case for the output sequence Y:

$$Y_{i,j} = \mathsf{max}_{m<K} X_{(i*K+m),j} \tag{2.2}$$

Another common pooling operation is average pooling with $K = S$, for which case:

$$Y_{i,j} = \mathbb{E}_{m<K}(X_{(i*K+m),j}) \tag{2.3}$$

A pooling operation has a receptive field K and does not have any trainable parameters. Each of two consecutive samples in the pooling operation's output depends on K input samples, however, these samples are spaced apart by on average S samples in the input sequence. This means that feeding the pooling operation's output to another layer effectively increases receptive field by a factor of S without an increase in trainable parameters. Finally, pooling operations introduce invariance to local translations of an input sequence which could be useful for the learning process but they entail information loss which could be detrimental to learning.

Chapter 3

Explanations in Terms of Knowledge

3.1 Introduction

The first concern when thinking about explanations of machine learning systems is the end-user. They could be a system developer that seeks explanations of the system they are developing, to improve it in a further iteration, or to troubleshoot. They could be a judge using a recidivism risk assessment tool, that requires explanations to assess fairness, as such tools have been shown to be biased [5]. They could be a person that was declined a loan by a bank's AI, and requires explanations that offer recourse: what should they do for their loan to get approved. For each of the above use-cases, the ideal explanation would be different. For instance, the developer supposedly has knowledge of the underlying system they are trying to explain, and would want the most *informative* explanation. On the other hand, the judge has no technical knowledge of how the automatic risk-assessment tool works, so they want explanations to be *understandable*. Furthermore, as the decision they have to make based on the explanation is critical, they want explanations to have *guarantees*, be *accurate*, and clearly stated using legal *terminology*. Contrarily, the person who was declined a loan application, wants the explanation that offers the best recourse.

The varying and ill-defined desiderata for good explanations has led to the development of a plethora of XAI methods [75]. First, there is a distinction between *local* explanations that aim to explain a specific input-output pair, and *global* explanations that aim to explain the behaviour of the system in general. An example of the former are *counterfactual explanations* of classifiers that answer the question: 'What is the minimal change we have to make to a data sample for it to be classified to class A instead of class B?'. Such explanations can be useful in the bank loan use-case [69, 199]. There are still however important problems to be solved for producing good counterfactual explanations, such as how to encode feasibility and actionability in the explanations [157], how to make explainers more robust [184, 161, 36], and what is an appropriate definition of minimality for each domain of application. An example of the latter (global explanations) are *rule-based explanations* [214, 139], some of which also make use of logics and reasoning [117, 174], that provide explanations in the form of logic rules, or decision trees that mimic the behaviour of the black-box. There are also local rule-based explanations [76], in addition to global counterfactual explanations [59], and a multitude of other forms of explanations, such as feature importance, and example-based. In this chapter however we are mainly concerned with rule-based and counterfactual explanations.

Each of these XAI methods is suited to different use-cases, and might have different priorities or outcomes regarding the desiderata (understandability, informativeness, accuracy etc.). It is thus important that we develop theoretical frameworks [81] to unify explanation methods, and subsequently to formally define and quantify what is a good explanation, and while there have been attempts to formalise notions of interpretability and its evaluation [51], there is still no agreement on what constitutes a good explanation [126]. To this end, symbolic AI systems play a key role in the eXplainable AI (XAI) field of research [143, 7], and one promising approach for mitigating the pitfalls of XAI methods is to utilize knowledge graphs [191] as a complement or extension to machine learning systems [112]. An example are global explanations of image classifiers, which would be very difficult to express in terms of pixels while maintaining understandability and informativeness. Instead, global explanations for computer vision often have the form of concept attribution

or concept importance methods [62, 206], which extract important concepts (such as "black and white stripes") and present them as global explanations. This motivates us to use formal knowledge representation, that allows us to define a specific terminology (for example names of concepts), and relationships between them (for example a hierarchy of concepts), in addition to descriptions of the data using this terminology, and to utilize this formally represented information for producing explanations.

In this chapter, we define a novel theoretical framework, based on Description Logics (DL), that includes rule-based and counterfactual explanations, and ways to compute them, based on knowledge. Specifically, we first introduce the background and notation used throught this chapter in section 3.2. Then, we describe the core idea of an *explanation dataset* in section 3.3, that contains items that can be fed to the black-box and are simultaneously described in a DL knowledge base, and acts as a bridge between the black-box classifier and the knowledge. In sections 3.4 and 3.5 we define rule-based and counterfactual explanations respectively, in the context of an explanation dataset. Finally, in section 3.6 we discuss other usages of explanation datasets, in addition to how we could convincingly measure informativeness, understandability, and accuracy by using explanation datasets in future work, and in section 3.7 we connlude the chapter.

3.2 Background and Notation

Description Logics Let $\mathcal{V} = \langle \mathsf{CN}, \mathsf{RN}, \mathsf{IN} \rangle$ be a *vocabulary*, where CN, RN, IN are mutually disjoint finite sets of *concept*, *role* and *individual* names, respectively. Let also $\mathcal{T}$ and $\mathcal{A}$ be a terminology (TBox) and an assertional database (ABox), respectively, over $\mathcal{V}$ using a Description Logics (DL) dialect $\mathcal{L}$, i.e. a set of axioms and assertions that use elements of $\mathcal{V}$ and constructors of $\mathcal{L}$. The pair $\langle \mathcal{V}, \mathcal{L} \rangle$ is a *DL-language*, and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a *(DL) knowledge base* (KB) over this language. The semantics of KBs are defined the standard model theoretical way using interpretations. Given a non-empty domain Δ, an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ assigns a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each $C \in \mathsf{CN}$, a set $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each $r \in \mathsf{RN}$, and an $a^{\mathcal{I}} \in \Delta$ to each $a \in \mathsf{IN}$. $\mathcal{I}$ is a *model* of a KB $\mathcal{K}$ iff it satisfies all assertions in $\mathcal{A}$ and all axioms in $\mathcal{T}$.

When the ABox $\mathcal{A}$ is a set of assertions of the form $C(a)$ and $r(a, b)$ where $C \in \mathsf{CN}$, $r \in \mathsf{RN}$ and $a, b \in \mathsf{IN}$, it can be represented as the labeled graph $\langle V, E, \ell_V, \ell_E \rangle$ (an *ABox graph*), where $V = \mathsf{IN}$ is the set of nodes, $E = \{\langle a, b \rangle \mid r(a, b) \in \mathcal{A}\} \subseteq \mathsf{IN} \times \mathsf{IN}$ is the set of labeled edges, $\ell_V : V \to 2^{\mathsf{CN}}$ with $\ell_V(a) = \{C \mid C(a) \in \mathcal{A}\}$ is the node labeling function, and $\ell_E : E \to 2^{\mathsf{RN}}$ with $\ell_E(a, b) = \{r \mid r(a, b) \in \mathcal{A}\}$ is the edge labeling function.

When the TBox $\mathcal{T}$ is a set of terminological axioms of the form $C \sqsubseteq D$ where $C, D \in \mathsf{CN}$ or $r \sqsubseteq s$ where $r, s \in \mathsf{RN}$, ie a hierarchy of concepts and roles, then it can be represented as a directed graph $\langle V, E \rangle$ (a TBox graph) where $V = \mathsf{CN} \cup \mathsf{RN} \cup \{\top\}$ the set of nodes. The set of edges E contains an edge for each axiom in the TBox, in addition to edges from atoms appearing only on the right side of subsumption axioms, and atoms that don't appear in the TBox, to the $\top$ node: $E = \{\langle a, b \rangle \mid a \sqsubseteq b \in \mathcal{T}\} \cup \{\langle a, \top \rangle \mid c \sqsubseteq a \in \mathcal{T} \wedge a \sqsubseteq d \notin \mathcal{T} \wedge c, d \in \mathsf{CN} \cup \mathsf{RN}\} \cup \{\langle a, \top \rangle \mid a \notin sig(\mathcal{T})\}$. This is abusive notation, in that the symbol $\top$ is overloaded and symbolizes both the universal concept and the universal role.

Conjunctive queries A *conjunctive query* (simply, a *query*) q over a vocabulary $\mathcal{V}$ is an expression $\{ \langle x_1, \ldots x_k \rangle \mid \exists y_1 \ldots \exists y_l . (c_1 \wedge \ldots \wedge c_n) \}$, where $k, l \geq 0$, $n \geq 1$, x_i, y_i are variables, each c_i is an atom $C(u)$ or $r(u, v)$, where $C \in \mathsf{CN}$, $r \in \mathsf{RN}$, u, v are some x_i, y_i or in IN, and all x_i, y_i appear in at least one atom. The vector $\langle x_1, \ldots x_k \rangle$ is the *head* of q, its elements are the *answer variables*, and $\{c_1, \ldots, c_n\}$ is the *body* of q. For simplicity, we write queries as $q \doteq \{c_1, \ldots, c_n\}_{x_1, \ldots, x_k}$. $\mathsf{vars}(q)$ is the set of all variables appearing in q. A query q can also be viewed as a graph, with a node v for each element in $\mathsf{vars}(q)$ and an edge (u, v) if there is an atom $r(u, v)$ in q, and labeling nodes and edges by the respective atom predicates. A query is *connected* if its graph is connected. In this paper we focus on connected queries having *one* answer variable in which all arguments of all c_is are variables, which we call *instance queries*. A query q_2 *subsumes* a query q_1 (we write $q_1 \leq_S q_2$) iff there is a

substitution θ s.t. $q_2\theta \subseteq q_1$. If q_1, q_2 are mutually subsumed, they are *syntactically equivalent*. Let be q a query and $q' \subseteq q$. If q' is a minimal subset of q s.t. $q' \leq_S q$, then q' is a *condensation* of q (cond(q)). Given a KB $\mathcal{K}$, an instance query q and an interpretation $\mathcal{I}$ of $\mathcal{K}$, a *match* for q is a mapping $\pi : \text{vars}(q) \to \Delta^{\mathcal{I}}$ such that $\pi(u) \in C^{\mathcal{I}}$ for all $C(u) \in q$, and $(\pi(u), \pi(v)) \in r^{\mathcal{I}}$ for all $r(u, v) \in q$. Then, a is a *(certain) answer* for q over $\mathcal{K}$ if in every model $\mathcal{I}$ of $\mathcal{K}$ there is a match π for q such that $\pi(x) = a^{\mathcal{I}}$. We denote the set of certain answers (*answer set*) to q by cert$(q, \mathcal{K})$. Because computing cert$(q, \mathcal{K})$ involves reasoning on $\mathcal{K}$, queries on top of DL KBs are characterized as *semantic queries*.

Rules A (definite Horn) *rule* is a First Order Logic (FOL) expression of the form $\forall x_1 ... \forall x_n \ (c_1, ..., c_n \Rightarrow c_0)$, usually written as $c_1, ..., c_n \to c_0$, where the c_is are atoms and x_i all appearing variables. In a rule over a vocabulary $\mathcal{V}$, each c_i is either $C(u)$ or $r(u, v)$, where $C \in \mathsf{CN}$, $r \in \mathsf{RN}$. The body of a rule can be represented as a graph similarly to the queries. In this paper we assume that all rules are *connected*, i.e. the graph of the body extended with the head variables is connected.

Classifiers A classifier is viewed as a function $F : \mathcal{D} \to \mathcal{C}$, where $\mathcal{D}$ is a domain of item feature data (e.g. images, audio, text), and $\mathcal{C}$ a set of classes (e.g. Dog, Cat).

3.3 Explanation Dataset

The first step for attempting to understand a black box is to choose what data to feed it. In this work we explore the merits of feeding it data for which there is available information in a knowledge base. This data comes in the form of what we call *exemplars*, that are described as individuals in the underlying knowledge, and can be mapped to the feature domain of the classifier. Such semantic information that describes exemplars can be acquired from knowledge graphs available on the web (for example wordnet [137]), or conceptnet [187]), it can be extracted using knowledge extraction methods (such as scene graph generation), or, ideally, it can be provided by domain experts, with the purpose of explaining opaque models. A motivating example would be a set of X-rays that have been thoroughly described by medical professionals, and using standardized medical terminology their characterizations have been encoded in a description logics knowledge base. Having such a set of exemplars allows us to provide explanations in terms of the underlying knowledge instead of being constrained by the features of the classifier.

Definition 1 (Explanation Dataset). *Let $\mathcal{D}$ be a domain of item feature data, $\mathcal{C}$ a set of classes, and $\mathcal{V} = \langle \mathsf{IN}, \mathsf{CN}, \mathsf{RN} \rangle$ a vocabulary such that $\mathcal{C} \cup \{\mathsf{Exemplar}\} \subseteq \mathsf{CN}$. Let also $\mathsf{EN} \subseteq \mathsf{IN}$ be a set of* exemplars. *An* explanation dataset *$\mathcal{E}$ in terms of $\mathcal{D}$, $\mathcal{C}$, $\mathcal{V}$ is a tuple $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$, where $\mathcal{M} : \mathsf{EN} \to \mathcal{D}$ is a mapping from the exemplars to the item feature data, and $\mathcal{S} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a DL KB over $\mathcal{V}$ such that $\mathsf{Exemplar}(a) \in \mathcal{A}$ iff $a \in \mathsf{EN}$, the elements of $\mathcal{C}$ do not appear in $\mathcal{S}$, and* Exemplar *and the elements of* EN *do not appear in $\mathcal{T}$.*

Intuitively, $\mathcal{D}$ contains items that can be fed to a classifier. Each such item is represented in the associated semantic data description by an individual (exemplar) $a \in \mathsf{EN}$, which is mapped to the respective feature data by $\mathcal{M}$. The knowledge base $\mathcal{S}$ contains the semantic data descriptions about all individuals in EN. The concept Exemplar is used solely to identify the exemplars within $\mathcal{A}$ (since other individual may exist) and should not appear elsewhere. The classes $\mathcal{C}$ should not appear in $\mathcal{S}$ so as not to take part in any reasoning process. The explanation dataset thus provides items with which we can probe the black-box classifier to explain it, by making use of the semantic descriptions of the items, in the context of the underlying knowledge.

An example of an explanation dataset for producing explanations is illustrated in figure 3.1. Consider an image classifier that operates on the pixel-level. In order to provide explanations for the classifier on a higher level of abstraction, we have available a set of images and their semantic descriptions, in the form of scene graphs (ABox). Furthermore, we have available terminological axioms involving the vocabulary used in the ABox, defining relationships between terms (TBox). Now consider that we notice that every image that depicts a cat or a dog is classified to class A,

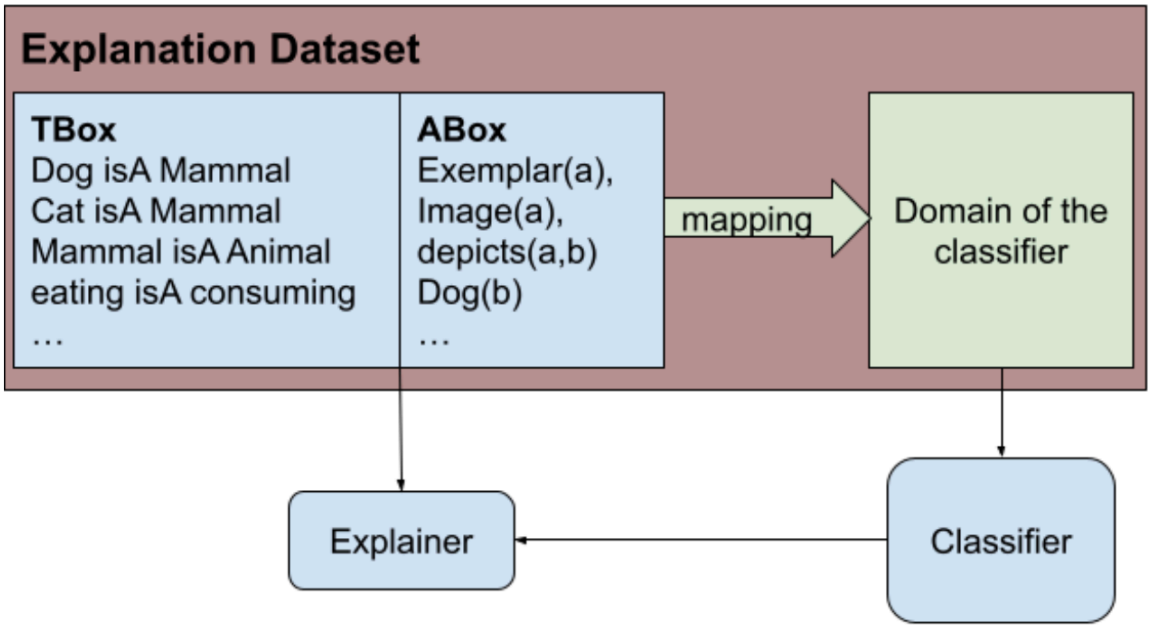


Figure 3.1: Overview of explanation dataset usage

while images that depict fish or dolphins are classified to class B. Using the knowledge to extrapolate, we conclude that the classifier classifies images depicting domestic animals to class A, and aquatic animals to class B.

3.4 Rule-based Explanations

The first type of explanations we will explore utilizing explanation datasets are rule-based explanations. Specifically, given an explanation dataset, an unknown classifier, and a class C, the aim of the rule-based explainer is to detect the semantic properties and relations of the exemplar data items that are classified by the unknown classifier to class C, and represent them in a human-understandable form, as rules utilizing the terminology of the knowledge.

Definition 2 (Explanation Rule). *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier , $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$ an explanation dataset in terms of $\mathcal{D}$, $\mathcal{C}$ and an appropriate vocabulary $\mathcal{V} = \langle \mathsf{CN}, \mathsf{RN}, \mathsf{IN} \rangle$. Given a concept $C \in \mathcal{C}$, the rule*

$$\mathsf{Exemplar}(x), c_1, c_2, \ldots, c_n \to C(x)$$

where c_i is an atom $D(u)$ or $r(u, v)$, where $D \in \mathsf{CN}$, $r \in \mathsf{RN}$, and u, v are variables, is an explanation rule *of F for class C over $\mathcal{E}$. We denote the rule by $\rho(F, \mathcal{E}, C)$, or simply by ρ whenever the context is clear. We may also omit $\mathsf{Exemplar}(x)$ from the body, since it is a conjunct of any explanation rule.*

Explanation rules describe *sufficient* conditions for an item to be classified in class C by a classifier. E.g., if the classifier classified images depicting wild animals in a zoo class, an explanation rule could be $\mathsf{Exemplar}(x), \mathsf{Image}(x), \mathsf{depicts}(x, y), \mathsf{WildAnimal}(y) \to \mathsf{Zoo}(x)$, assuming that $\mathsf{Image}, \mathsf{WildAnimal} \in \mathsf{CN}$, $\mathsf{depicts} \in \mathsf{RN}$, and $\mathsf{Zoo} \in \mathcal{C}$. It is important that explanation rules refer only to individuals $a \in \mathsf{EN}$ that correspond to items $\mathcal{M}(a) \in \mathcal{D}$; this is guaranteed by the conjunct $\mathsf{Exemplar}(x)$ in the explanation rule body. Indeed, since the classifier under explanation is unknown, the only guaranteed information is the classification of the exemplars.

Given a classifier $F : \mathcal{D} \to \mathcal{C}$ and a set of individuals $\mathcal{I} \subseteq \mathsf{EN}$, the positive set (pos-set) of F on $\mathcal{I}$ for class $C \in \mathcal{C}$ is $\mathsf{pos}(F, \mathcal{I}, C) = \{a \in \mathcal{I} : F(\mathcal{M}(a)) = C\}$, ie the pos-set for a class is the set of individuals that are mapped to items that are classified positively to the specific class.

Definition 3 (Explanation Rule Correctness). *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier, $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$ an explanation dataset in terms of $\mathcal{D}$, $\mathcal{C}$ and an appropriate vocabulary $\mathcal{V}$, and $\rho(F, \mathcal{E}, C)$ an explanation rule. The rule ρ is* correct *over F and $\mathcal{E}$ if and only if*

$$\begin{aligned}\mathsf{fol}(\mathcal{S} \cup \{&\mathsf{Exemplar} \sqsubseteq \{a \mid a \in \mathsf{EN}\}\} \\ &\cup \{C(a) \mid a \in \mathsf{pos}(F, \mathsf{EN}, C)\}) \models \rho\end{aligned}$$

where $\mathsf{fol}(\mathcal{K})$ *is the first-order logic translation of DL KB $\mathcal{K}$.*

The intended meaning of a correct explanation rule is that for every $a \in \mathsf{EN}$, if the body of the rule holds, then the classifier classifies $\mathcal{M}(a)$ to the class indicated in the head of the rule. Intuitively, an explanation rule is correct if it is a logical consequence of the underlying knowledge extended by the axiom $\mathsf{Exemplar} \sqsubseteq \{a \mid a \in \mathsf{EN}\}$, which forces $\mathsf{Exemplar}(x)$ to be true in an interpretation $\mathcal{I}$ only for $x = a^{\mathcal{I}}$ with $a \in \mathsf{EN}$. For instance, the rule of the previous example $\mathsf{Exemplar}(x), \mathsf{Image}(x), \mathsf{depicts}(x, y), \mathsf{WildAnimal}(y) \to \mathsf{ZooClass}(x)$ would be correct for the KB $\mathcal{S}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$, where $\mathcal{A}_1 = \{\mathsf{Image(a)}, \mathsf{depicts(a, b)}, \mathsf{Wolf(b)}\}$ and $\mathcal{T}_1 = \{\mathsf{Wolf} \sqsubseteq \mathsf{WildAnimal}\}$ if $\mathsf{a} \in \mathsf{pos}(F, \mathsf{EN}, \mathsf{ZooClass})$, while it would not be correct for the KB $\mathcal{S}_2 = \langle \emptyset, \mathcal{A}_1 \rangle$, nor would it be correct for $\mathcal{S}_1$ if $\mathsf{a} \notin \mathsf{pos}(F, \mathsf{EN}, \mathsf{ZooClass})$. Checking whether a rule is correct is a reasoning problem which can be solved by using standard DL reasoners. On the other hand, finding rules which are correct is an inverse problem which is much harder to solve.

As mentioned in section 3.2, an instance query has the form $\{c_1, \dots, c_n\}_x$, which resembles the body of an explanation rule with head some $C(x)$. Thus, by representing the bodies of explanation rules as queries, the computation of explanations can be treated as a query reverse engineering problem.

Definition 4 (Explanation Rule Query). *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier, $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$ an explanation dataset in terms of $\mathcal{D}$, $\mathcal{C}$ and an appropriate vocabulary $\mathcal{V}$, and $\rho(F, \mathcal{E}, C)$: $\mathsf{Exemplar}(x), c_1, c_2, \dots, c_n \to C(x)$ an explanation rule. The instance query*

$$q_\rho \doteq \{\mathsf{Exemplar}(x), c_1, c_2, \dots, c_n\}_x$$

is the explanation rule query *of explanation rule ρ.*

This definition establishes a 1-1 relation (up to variable renaming) between ρ and q_ρ. To compute queries corresponding to explanation rules that are guaranteed to be correct, we prove Theorem 1

Theorem 1. *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier, $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$ an explanation dataset in terms of $\mathcal{D}$, $\mathcal{C}$ and an appropriate vocabulary $\mathcal{V}$, $\rho(F, \mathcal{E}, C)$: $\mathsf{Exemplar}(x), c_1, c_2, \dots, c_n \to C(x)$ an explanation rule, and q_ρ the explanation rule query of ρ. The explanation rule ρ is correct if and only if*

$$\mathsf{cert}(q_\rho, \mathcal{S}) \subseteq \mathsf{pos}(F, \mathsf{EN}, C)$$

Proof. Let $\mathcal{S} = \langle \mathcal{T}, \mathcal{A} \rangle$. Because by definition $\mathsf{Exemplar}(a) \in \mathcal{A}$ iff $a \in \mathsf{EN}$ and Exemplar does not appear anywhere in $\mathcal{T}$, we have

$$\begin{aligned}\mathsf{cert}(q_\rho, \mathcal{S}) &= \mathsf{cert}(\{\mathsf{Exemplar}, c_1, \dots, c_n\}, \langle \mathcal{T}, \mathcal{A} \rangle) \\ &= \mathsf{EN} \cap \mathsf{cert}(\{c_1, \dots, c_n\}, \langle \mathcal{T}, \mathcal{A} \rangle) \\ &= \mathsf{cert}\left(\{\mathsf{Exemplar}\}, \left\langle \left\{ \mathsf{Exemplar} \sqsubseteq \bigsqcup_{a \in \mathsf{EN}} \{a\} \right\}, \mathcal{A} \right\rangle\right) \cap \mathsf{cert}(\{c_1, \dots, c_n\}, \langle \mathcal{T}, \mathcal{A} \rangle) \\ &= \mathsf{cert}\left(\{\mathsf{Exemplar}, c_1, \dots, c_n\}, \left\langle \mathcal{T} \cup \left\{ \mathsf{Exemplar} \sqsubseteq \bigsqcup_{a \in \mathsf{EN}} \{a\} \right\}, \mathcal{A} \right\rangle\right) \\ &= \mathsf{cert}\left(q_\rho, \left\langle \mathcal{T} \cup \left\{ \mathsf{Exemplar} \sqsubseteq \bigsqcup_{a \in \mathsf{EN}} \{a\} \right\}, \mathcal{A} \right\rangle\right)\end{aligned}$$

Because by definition C does not appear anywhere in $\mathcal{S}$), we have also that $\mathsf{cert}(q_\rho, \mathcal{S}) = \mathsf{cert}(q_\rho, \mathcal{S}')$, where $\mathcal{S}' = \mathcal{S} \cup \{\mathsf{Exemplar} \sqsubseteq \bigsqcup_{a \in \mathsf{EN}} \{a\}\} \cup \{C(a) \mid a \in \mathsf{pos}(F, \mathsf{EN}, C)\}\}$, since the assertions $C(a)$ are not involved neither in the query nor in $\mathcal{S}$ and hence have no effect.

By definition of a certain answer, $e \in \mathsf{cert}(q, \mathcal{K})$ iff for every model $\mathcal{I}$ of $\mathcal{K}$ there is a match π s.t. $\pi(x) = e^{\mathcal{I}}$ and $\pi(u) \in D^{\mathcal{I}}$ for all $D(u) \in q$ and $(\pi(u), \pi(v)) \in r^{\mathcal{I}}$ for all $r(u, v) \in q$.

Assume that ρ is correct and let $e \in \mathsf{cert}(q_\rho, \mathcal{S})$. We have proved that also $e \in \mathsf{cert}(q_\rho, \mathcal{S}')$. Because ρ is correct, by Def. 3 it follows that every model $\mathcal{I}$ of $\mathcal{S}'$ is also a model of ρ. Because the body of q_ρ is the same as the body of ρ, π makes true both the body of ρ and the head of ρ, which is $C(x)$, hence $e^{\mathcal{I} \in C^{\mathcal{I}}}$. It follows that $C(e)$ is true in $\mathcal{I}$. But the only assertions of the form $C(e)$ in $\mathcal{S}'$ are the assertions $\{C(a) \mid a \in \mathsf{pos}(F, \mathsf{EN}, C)\}$, thus $e \in \mathsf{pos}(F, \mathsf{EN}, C)$.

For the inverse, assume that $\mathsf{cert}(q_\rho, \mathcal{S}) \subseteq \mathsf{pos}(F, \mathsf{EN}, C)$, equivalently $\mathsf{cert}(q_\rho, \mathcal{S}') \subseteq \mathsf{pos}(F, \mathsf{EN}, C)$. Thus if $e \in \mathsf{cert}(q_\rho, \mathcal{S})$ then $e^{\mathcal{I} \in C^{\mathcal{I}}}$. Since this holds for every model $\mathcal{I}$ of $\mathcal{S}'$ and the body of q_ρ is the same as the body of ρ, it follows that $\mathcal{I}$ is also a model of ρ, i.e. ρ is correct.

□

Theorem 1 allows us to compute guaranteed correct rules, by finding a query q for which $\mathsf{cert}(q, \mathcal{S}) \subseteq \mathsf{pos}(F, \mathsf{EN}, C)$. Intuitively, an explanation rule query is correct for class C, if all of its certain answers are mapped by $\mathcal{M}$ to feature data which is classified in class C. It follows that a query with one certain answer which is an element of the pos-set is a correct rule query, as is a query q for which $\mathsf{cert}(q, \mathcal{S}) = \mathsf{pos}(F, \mathsf{EN}, C)$. Thus, it is useful to define a *recall* metric for explanation rule queries by comparing the set of certain answers with the pos-set of a class C, as shown in equation 3.1.

Furthermore, an explanation rule query might not be correct due to the existence of individuals in the set of certain answers which are not in the pos-set. By viewing these individuals as exceptions to a rule, we are able to provide as an explanation a rule that is not correct, along with the exceptions which would make it correct if they were omitted from the explanation dataset; the exceptions could provide useful information to an end-user about the classifier under investigation. Thus, we extend the framework by introducing correct explanation rules with exceptions, as follows.

Definition 5 (Explanation rule with exceptions). *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier, $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$ an explanation dataset in terms of $\mathcal{D}$,$\mathcal{C}$ where $\mathcal{S}$ is a knowledge base $\mathcal{S} = \langle \mathcal{A}, \mathcal{T} \rangle$, EN the set of exemplars of $\mathcal{E}$, and let EX be a subset of EN. An explanation rule $\rho(F, \mathcal{E}, C)$ is correct with exceptions EX for class C if the rule $\rho(F, \mathcal{E}', C)$ is correct for class C, where $\mathcal{E}' = \langle \mathcal{M}, \mathcal{S}' \rangle$, and $\mathcal{S}'$ is the knowledge $\mathcal{S}' = \langle \mathcal{A}', \mathcal{T} \rangle$, and $\mathcal{A}' = \mathcal{A} \ \ \{\mathsf{Exemplar}(a) \mid a \in \mathsf{EX}\}$.*

Since we allow exceptions to explanation rules, it is useful to define a measure of precision of the corresponding explanation rule queries as shown in equation 3.2. Obviously, if the precision of a rule query is 1, then it represents a correct rule, otherwise it is correct with exceptions. Furthermore, we can use the Jaccard similarity between the set of certain answers of the explanation rule query and the pos-set, as a generic measure which combines recall and precision to compare the two sets of interest as what we call the degree of an explanation rule query, as shown in equation 3.3.

Metrics

$$\mathsf{recall}(q, \mathcal{E}, C) = \frac{|\mathsf{cert}(q, \mathcal{S}) \cap \mathsf{pos}(F, \mathsf{EN}, C)|}{|\mathsf{pos}(F, \mathsf{EN}, C)|}, \tag{3.1}$$

$$\mathsf{precision}(q, \mathcal{E}, C) = \frac{|\mathsf{cert}(q, \mathcal{S}) \cap \mathsf{pos}(F, \mathsf{EN}, C)|}{|\mathsf{cert}(q, \mathcal{S})|}, \tag{3.2}$$

$$\mathsf{degree}(q, \mathcal{E}, C) = \frac{|\mathsf{cert}(q, \mathcal{S}) \cap \mathsf{pos}(F, \mathsf{EN}, C)|}{|\mathsf{cert}(q, \mathcal{S}) \cup \mathsf{pos}(F, \mathsf{EN}, C)|}. \tag{3.3}$$

3.4.1 Computing Explanation Rules

As we have already mentioned, the computation of explanation rules in the context of this framework can be reduced to a query reverse engineering problem. This query reverse engineering problem follows the *Query by Example* paradigm or QbE. This term refers to reverse engineering queries that contain some positive examples in their answer set but not any negative examples and is widely used in the related literature [224, 131, 79, 149, 98, 32]. It has been shown to be a difficult problem to solve for conjunctive queries (coNEXPTIME-Complete [11]), thus any algorithms developed will either be extremely resource intensive, and not scalable, or they will solve a relaxed version of the problem, and provide approximate solutions. In this thesis we use two algorithms for computing explanation rules. The first is an adaptation of the method developed by Chortaras et al. in [28] for exploring the entire space of semantic queries in DL knowledge bases, which we call KGRules, and the second developed by Liartis et al. in [122] that searches the space of semantic queries using heuristic criteria.

KGRules

The computation of arbitrary candidate explanation rule queries for the KB $\mathcal{S}$ of an explanation dataset is in general hard since it involves exploring the query space $\mathcal{Q}$ of all queries that can be constructed using the underlying vocabulary $\mathcal{V}$ and getting their certain answers for $\mathcal{S}$. Difficulties arise even in simple cases, since the query space is in general infinite. However, the set of all possible distinct answer sets is finite and in most cases it is expected to be much smaller than its upper limit, the powerset 2^{IN}.

Alg. 1 explores a useful finite subset of $\mathcal{Q}$, namely the tree-shaped queries of a maximum depth k [64]. It constructs all possible such queries (that include $\mathsf{Exemplar}(x)$ in the body), obtains their answers, and arranges them in a directed acyclic graph (the *query space DAG*) using the subset relation on the answer sets. The queries are constructed in the for loop, and then the while loop replaces queries having the same answer set by their intersection. The *intersection* $q_1 \sqcap q_2$ of two instance queries q_1, q_2 with answer variable x is the query $\mathsf{cond}(q_1 \cup q_2\theta)$, where θ renames each variable appearing in q_2 apart from x to a variable not appearing in q_1. Thus, from all possible queries with the same answers, the algorithm keeps only the *most specific* query q of all such queries. Intuitively, this is the most detailed query. Finally, the queries are arranged in a DAG. By construction, each node of the DAG is a query representing a distinct answer set.

Theorem 2. *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier, $\mathcal{E} = \langle \mathcal{M}, \mathcal{S} \rangle$ an explanation dataset in terms of $\mathcal{D}$, $\mathcal{C}$ and an appropriate vocabulary $\mathcal{V}$, and $\rho(F, \mathcal{E}, C)$ a correct tree-shaped explanation rule of maximum depth k. The DAG constructed by Alg. 1 contains a query $q_{\rho'}$ corresponding to a correct explanation rule $\rho'(F, \mathcal{E}, C)$ with the same metrics as ρ, s.t. $q_{\rho'} \leq_S q_\rho$.*

Given Theorem 2 a node corresponding to a correct rule for some $\mathsf{pos}(F, \mathsf{EN}, C)$ can be reached by traversing the graph starting from the root and finding the first node whose answer set equals $\mathsf{pos}(F, \mathsf{EN}, C)$. The descendants of that node provide all queries corresponding to correct explanation rules. The DAG has a unique root because answer sets are subsets of $\mathsf{cert}(\{\mathsf{Exemplar}(x)\}_x, \mathcal{S})$.

An unavoidable difficulty in using Alg. 1 is its complexity. The sizes of $\mathcal{B}$ and $\mathcal{F}$ are at the orders of $2^{|\mathsf{CN}|}$ and $4^{|\mathsf{RN}|}$ respectively, and the number of tree-shaped queries with k variables is at the order of $2^{k|\mathsf{CN}|} \cdot 4^{(k-1)|\mathsf{RN}|}$. However, in practice the query space is much smaller since most queries have zero answers and can be ignored. To get answer sets, Alg. 1 assumes a function that returns $\mathsf{cert}(q, \mathcal{K})$ for any query q.

If $\mathcal{K}$ is fully materialized, i.e. if no reasoning is needed to answer queries, it is easy to implement the function for $\mathsf{cert}(q, \mathcal{K})$. The sets $\mathcal{B}$ and $\mathcal{F}$ can be computed to contain only queries with at least one answer, and queries can be constructed incrementally; once a query with no answers is reached, no queries with additional conjuncts are considered.

If $\mathcal{K}$ is not materialized, or impossible to materialize, the incremental query construction process should by coupled with the necessary reasoning to get the query answers. For the DL-Lite$_{\mathcal{R}}^{-}$ dialect,

Algorithm 1: QuerySpaceDAG

Data: Vocabulary $\mathcal{V}$, KB $\mathcal{K}$, a maximum query depth $k \geq 0$
Result: Query space DAG $\mathcal{G}$

Compute the set $\mathcal{B}$ of all non-syntactically equivalent queries $\{C_1(x), \dots, C_n(x)\}_x$, where $C_i \in \mathsf{CN}\ \{\mathsf{Exemplar}\}$, $n \geq 1$.;
Compute the set $\mathcal{F}$ of all non-syntactically equivalent queries $\{r_1(u_1, v_1), \dots, r_n(u_n, v_n)\}_{x,y}$, where $r_i \in \mathsf{RN}$, $n \geq 1$, each u_i, v_i is either x or y and $u_i \neq v_i$.;
Initialize an empty set of queries $\mathcal{Q}$.;
for $i = 0 \dots k$ **do**
    Compute the set $\mathcal{T}_i$ of all trees of depth i.;
    foreach $t \in \mathcal{T}_i$ **do**
        Assign to each node v of t a distinct variable $\mathsf{var}(v)$. Assign x to the root of t.;
        Construct all non-syntactically equivalent queries q obtained from t by adding to the body of q: i) for each node v of t, the body of an element of $\mathcal{B} \cup \{\emptyset\}$ after renaming x to $\mathsf{var}(v)$, ii) for each edge (v_1, v_2) of t, the body of an element of $\mathcal{F}$ after renaming x to $\mathsf{var}(v_1)$ and y to $\mathsf{var}(v_2)$, and iii) $\mathsf{Exemplar}(x)$.;
        Condense all qs and add them to $\mathcal{Q}$.;
    end
end
while there are $q_1, q_2 \in \mathcal{Q}$ s.t. $\mathsf{cert}(q_1, \mathcal{K}) = \mathsf{cert}(q_2, \mathcal{K})$ **do**
    remove q_1, q_2 from $\mathcal{Q}$ and add $q_1 \sqcap q_2$ to $\mathcal{Q}$.;
end
Arrange the elements of $\mathcal{Q}$ in a DAG $\mathcal{G}$, making q_1 a child of q_2 iff $\mathsf{cert}(q_1, \mathcal{K}) \subset \mathsf{cert}(q_2, \mathcal{K})$.;
return the transitive reduction of $\mathcal{G}$

a more efficient alternative to Alg. 1 is proposed in [28]. DL-Lite$_{\mathcal{R}}^{-}$ allows only axioms of the form $C \sqsubseteq D$ or $r \sqsubseteq s$, where C, D are concepts, and r, s are atomic roles. D can be either atomic or of the form $\exists r^{(-)}.\top$, and C can be of the form $\exists r^{(-)}.A$, where A is atomic. The authors exploit the fact that query answering in DL-Lite$_{\mathcal{R}}^{-}$ can be done in steps by rewriting a query to a set of queries, the union of whose answer sets are the answers to the original query, to incrementally compute the tree-shaped queries of a maximum depth with at least one answer.

Further simplifications to reduce the practical complexity of Alg. 1, that may affect its theoretical properties, include not condensing queries, keeping an arbitrary query for each answer set instead of the most specific one, and setting a minimum answer set size threshold for a query to be considered.

KGRules-H

As the KGRules algorithm is not always practical, due to its complexity, Liartis et al. developed KGRules-H, that provides approximate solutions by using heuristic criteria. The pipeline of the algorithm is illustrated in figure 3.2, and is outlined in algorithm 2.

First of all, KGrules-H can only be applied for knowledge bases in which the TBox can be eliminated. In particular, *TBox elimination* is the process of expanding a given ABox by applying the axioms of a given TBox, so that the TBox in eventually not needed, in the sense that all certain answers of the original knowledge can now be obtained only from the expanded ABox. Because TBox elimination is not always possible, this poses certain restrictions on the applicability of Alg. 2, namely that if the original knowledge of the explanation dataset is indeed of the form $\langle \mathcal{T}, \mathcal{A} \rangle$ with $\mathcal{T} \neq \emptyset$, it should be possible, as outlined above, to be transformed through TBox elimination to an equivalent w.r.t. query answering finite assertional-only knowledge base $\mathcal{A}'$, such that $\mathsf{cert}(q, \langle \mathcal{T}, \mathcal{A} \rangle) = \mathsf{cert}(q, \langle \emptyset, \mathcal{A}' \rangle)$ for any q. For knowledge bases that this is possible, the standard approach for TBox elimination is *materialization* and is typically performed by first encoding the

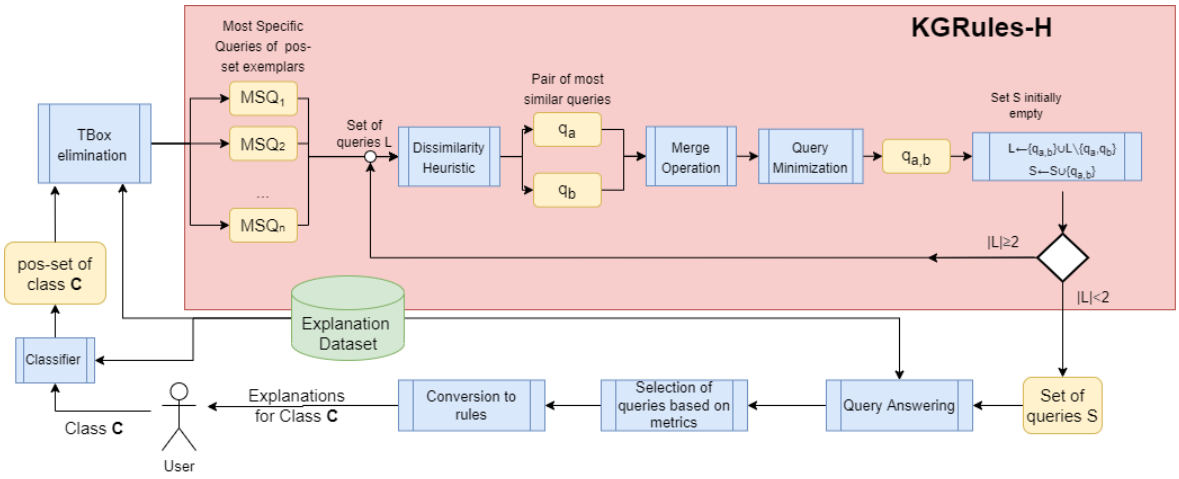


Figure 3.2: Visualization of how KGrules-H is integrated into our framework.

Algorithm 2: KGRULES-H

Input: An atomic ABox $\mathcal{A}$ and a set of individual names I.
Output: A list of queries S.

$S \leftarrow [\,]$
$L \leftarrow \{\mathsf{MSQ}(a, \mathcal{A}) \mid a \in I\}$
while $|L| \geq 2$ **do**
  $q_A, q_B \leftarrow \arg\min_{q,q' \in L,\, q \neq q'} \mathsf{QueryDissimilarity}(q, q')$
  $q \leftarrow \mathsf{Merge}(q_A, q_B)$
  $L \leftarrow (L \quad \{q_A, q_B\}) \cup \{q\}$
  append q to S
end
return S

axioms in $\mathcal{T}$ as a set of inference rules generating ABox assertions, and then iteratively applying them on the knowledge base until no more assertions can be generated. Materialization is possible, e.g. for Description Logic Programs and the Horn-$\mathcal{SHIQ}$ DL dialect. Finite materialization may not be possible even for low expressivity DL dialects, such as DL-Lite. [106, 65].

Having eliminated the TBox, the algorithm starts with the Most Spcific Query (MSQ) of each exemplar in the pos-set, and iteratively merges the most similar queries, with the goal of producing more general descriptions, eventually describing the entire pos-set. Different criteria can be used for computing similarity of queries, and different methods can be used for merging queries, and these are hyperparameters of algorithm 2.

Specifically, for a cheap approximation of query dissimilarity, Liartis et al. define it as follows. Given two queries q_1, q_2 with respective graph representations $G_1 = (V_1, E_1, \ell_{V_1}, \ell_{E_1})$ and $G_2 = (V_2, E_2, \ell_{V_2}, \ell_{E_2})$, we define the *query dissimilarity* heuristic between q_1 and q_2 as follows:

$$\mathsf{QueryDissimilarity}(q_1, q_2) = \sum_{v_1 \in V_1} \min_{v_2 \in V_2} \mathsf{diss}_{q_1 q_2}(v_1, v_2) + \sum_{v_2 \in V_2} \min_{v_1 \in V_1} \mathsf{diss}_{q_2 q_1}(v_2, v_1)$$

where

$$\mathsf{diss}_{q_1 q_2}(v_1, v_2) = |L_1(v_1) \quad L_2(v_2)|$$
$$+ \sum_{r \in R} \{\max(\mathsf{indegree}^r_{G_1}(v_1) - \mathsf{indegree}^r_{G_2}(v_2), 0) + \max(\mathsf{outdegree}^r_{G_1}(v_1) - \mathsf{outdegree}^r_{G_2}(v_2), 0)\},$$

R is the set of all role names appearing in the edge labels of the two graphs, and $\mathsf{indegree}^r_G(v)$ ($\mathsf{outdegree}^r_G(v)$) is the number of incoming (outcoming) edges e in node v of graph G with $r \in \ell(e)$. The intuition behind this dissimilarity measure is that the graphs of queries which are dissimilar consist of nodes with dissimilar labels connected in dissimilar ways. Intuitively, we expect such queries to have dissimilar sets of certain answers, although there is no guarantee that this will always be the case.

For merging queries into more general ones, Liartis et al. use two different procedures. The first is the computation of the Query Least Common Subsumer (QLCS) of the two queries, which is defined as the most specific generalization of two queries, and is computed as the Kroenecker product of the two queries. The second, outlined in algorithm 3, greedily merges queries based on their common conjuncts. Since the Kroenecker product of two graphs of $|V_1|$ and $|V_2|$ vertices, will have $|V_1| \times |V_2|$ vertices, which in the context of graph representations of queries correspond to variables, the merged queries will have many redundant conjuncts. As these queries will be iteratively merged, and the end-result will be shown to a human, after each merge, Liartis et al perform query minimization by removing redundant conjuncts. Unfortunately, the problem of minimizing a query, also known as condensation is coNP-complete [71], so they propose an approximation to query minimization described in algorithm 4.

Algorithm 3: GreedyCommonConjuncts

Input: Two queries $q_1,\ q_2$, with query graphs
$G_{q_1} = \langle V_1, E_1, \ell_{V_1}, \ell_{E_1} \rangle, G_{q_2} = \langle V_2, E_2, \ell_{V_2}, \ell_{E_2} \rangle$.
Output: A query consisting of common conjuncts of q_1 and q_2.

if $|\mathsf{var}(q_1)| < |\mathsf{var}(q_2)|$ **then**
  Swap q_1, q_2
end
$\theta \leftarrow \{\ \}$
$U \leftarrow \mathsf{var}(q_1) \quad \{x\}$
$V \leftarrow \mathsf{var}(q_2) \quad \{x\}$
while true do
  $S \leftarrow \{z \mapsto y \mid z \in V,\ y \in U,\ \langle z, z' \rangle \in E_1,\ \langle y, y' \rangle \in E_2,\ \ell_{E_1}(\langle z, z' \rangle) \cap \ell_{E_2}(\langle y, y' \rangle) \neq \emptyset,\ z' \mapsto y' \in \theta\}$
    $\cup \{z \mapsto y \mid z \in V,\ y \in U,\ \langle z', z \rangle \in E_1,\ \langle y', y \rangle \in E_2,\ \ell_{E_1}(\langle z', z \rangle) \cap \ell_{E_2}(\langle y', y \rangle) \neq \emptyset,\ z' \mapsto y' \in \theta\}$
  if $S \neq \emptyset$ **then**
    $\hat{z} \mapsto \hat{y} \leftarrow \arg\max_{z \mapsto y \in S} \{|q_1 \cap q_2(\theta \cup \{z \mapsto y\})| - |q_1 \cap q_2\theta|\}$
    $\theta \leftarrow \theta \cup \{\hat{z} \mapsto \hat{y}\}$
    $V \leftarrow V \quad \{\hat{z}\}$
    $U \leftarrow U \quad \{\hat{y}\}$
  else
    break
  end
end
$q \leftarrow q_1 \cap q_2\theta$
return q

While the details of how to approximate solutions to the query reverse engineering problem, and the KGrules-H algorithm are out of the scope of this dissertation, we have mentioned some of the details, to show i) why the problem is theoretically difficult (merging, minimizing, comparing), and ii) How with specific relaxations, approximate explanation rules can be computed in a scalable way. For further details on KGrules-H, we refer to the work of Liartis et al.

Algorithm 4: ApproxQueryMinimize

Input: A query represented as a graph $q = \langle V, E, \ell_V, \ell_E \rangle$.
Output: An approximately minimized query q', also represented as a graph.

do
  $q' \leftarrow q$
  foreach pair (v, v'), $v, v' \in V$, $v \neq v'$ **do**
    if $\ell_V(v') \subseteq \ell_V(v)$ **and**
      $\langle v', v'' \rangle \in E \Rightarrow (\langle v, v'' \rangle \in E,\ \ell_E(v', v'') \subseteq \ell_E(v, v'')),\ v'' \neq v'$ **and**
      $\langle v'', v' \rangle \in E \Rightarrow (\langle v'', v \rangle \in E,\ \ell_E(v'', v') \subseteq \ell_E(v'', v)),\ v'' \neq v'$ **and**
      $\langle v', v' \rangle \in E \Rightarrow (\langle v, v \rangle \in E,\ \ell_E(v', v') \subseteq \ell_E(v, v))$ **then**
      Delete variable v' from q.
    end
  end
while $q' \neq q$
return q'

3.5 Counterfactual Explanations

The second type of explanations we explore utilizing explanation datasets are counterfactual explanations. As these are supposed to answer the question "What has to change for a data sample to be classified to class B instead of class A", they often have the form of *input edits*. ie small changes of the features of the data sample that lead to a different classification by the black-box. In our approach, counterfactual explanations have the form of *semantic edits* that are applied on an ABox corresponding to an explanation dataset, instead of the features of the data sample, while the notion of "minimality" (small changes), is defined based on the underlying knowledge. Specifically, given an exemplar and a desired class, we are searching for a set of edits that when applied on the ABox lead to the exemplar being *indistinguishable* from any exemplar that is classified to the desired class, where two exemplars are indistinguishable if their connected components on the ABox graph are equal.

Definition 6 (Counterfactual Explanation). *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier and $\langle \mathcal{M}, \mathcal{K} \rangle$ an explanation dataset where $\mathcal{M} : \mathsf{EN} \to \mathcal{D}$ is a mapping function, EN is a set of exemplars and $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ is a knowledge base. A* counterfactual explanation *for an exemplar $a \in \mathsf{EN}$ and class $C \in \mathcal{C}$ is a tuple $\langle c, E \rangle$ where $c \in \mathsf{EN}$ and $F(\mathcal{M}(c)) = C$, and E is a set of* edit operations *that when applied on the connected component of a on the ABox graph make it equal to the connected component of c. An edit operation on an ABox can be any of:*

- *Replacement of assertion $D(a)$ with $E(a)$, symbolized $e_{D \to E}$*
- *Replacement of $r(a, b)$ with $s(a, b)$, symbolized $e_{r \to s}$*
- *Deletion of $D(a)$ or $r(a, b)$, symbolized $e_{D \to \top}$ or $e_{r \to \top}$*
- *Insertion of $D(a)$ or $r(a, b)$, symbolized $e_{\top \to D}$ or $e_{\top \to r}$*

where $D, E \in \mathsf{CN}$ and $r, s \in \mathsf{RN}$.

For example, consider an image classifier F that classifies to the classes

$$\mathcal{C} = \{\mathsf{WildAnimal}, \mathsf{DomesticAnimal}\}$$

and two exemplars e_1, e_2 each classified to a different class: $F(e_1) = \mathsf{WildAnimal}$ and $F(e_2) = \mathsf{DomesticAnimal}$. The connected components of each exemplar in the ABox graph might be:

$$\mathcal{A}_{e_1} = \{\mathsf{Exemplar}(e_1), \mathsf{depicts}(e_1, a), \mathsf{depicts}(e_1, b), \mathsf{isIn}(a, b), \mathsf{Animal}(a), \mathsf{Forest}(b)\}$$

$$\mathcal{A}_{e_2} = \{\mathsf{Exemplar}(e_2), \mathsf{depicts}(e_2, c), \mathsf{depicts}(e_2, d), \\ \mathsf{isIn}(c, d), \mathsf{Animal}(c), \mathsf{Bedroom}(d)\}$$

Then an explanation for exemplar e_1 and class DomesticAnimal would be the replacement of assertion $\mathsf{Forest}(b)$ with $\mathsf{Bedroom}(b)$, which would be symbolized $\langle e_2, \{e_{\mathsf{Forest}\to\mathsf{Bedroom}}\}\rangle$ and it would be interpreted by a user as "If image e_1 depicted animal a in a Bedroom instead of a Forest, then the image would be classified as a DomesticAnimal". Of course there is no way to know if the image e_1 with the Forest replaced with a Bedroom would be classified to the target class, because we do not have a way to edit the pixels of the image and feed it to the classifier. The explanation however provides useful information to the user and can potentially aid in the detection of biases of the classifier. For example, after viewing this explanation, the user might choose to feed the classifier images depicting wild animals in bedrooms to see whether or not they are misclassified as domestic animals.

To provide more information to the end user, we can accumulate counterfactual explanations for multiple exemplars and the desired class and provide statistics about what changes tend to flip the prediction of the classifier, as a form of a "global" explanation. For example, one could ask "What are the most common semantic edits that when applied on exemplars depicting bedrooms lead to them to be classified as wild animals?". To do this, we first compute the multiset $\mathcal{G}$ of all counterfactual explanations from each exemplar in the source subset to the target class, and then we show the end-user the *importance* of each atom for changing the prediction on the source exemplars to the target class, where

$$\mathsf{Importance}(y) = \frac{|\{e_{x\to y} \in \mathcal{G}\}| - |\{e_{y\to x} \in \mathcal{G}\}|}{|\mathcal{G}|} \tag{3.4}$$

where , $x, y \in \mathsf{CN}$, or $x, y \in \mathsf{RN}$.

Intuitively, the importance of an atom shows how often it is introduced (either via replacement or via insertion) as part of the semantic edits of a set of counterfactual explanations. A negative importance would indicate that the atom tends to be removed (either via replacement or via deletion of assertions). For example, one could gather all exemplars that are classified as WildAnimal, along with their counterfactual explanations for target class DomesticAnimal and compute how important the presence (or absence) of a concept or a role is for distinguishing between the two classes.

3.5.1 Computing Counterfactual Explanations

Given an explanation dataset $\langle \mathsf{EN} \to \mathcal{D}, \langle \mathcal{A}, \mathcal{T}\rangle\rangle$, the first step for computing useful counterfactual explanations is to determine the edit operations on the ABox that transform the description of every exemplar to every other exemplar, thus this is a computation that has to be done $O(|\mathsf{EN}|^2)$ times, but it only has to be done once for an explanation dataset. Ideally, each set of edit operations will be minimal as they are intended to be shown to users as explanations, which means that the problem to be solved is the exact graph edit distance problem [173].

Edit Distance Between Exemplars

Unfortunately, computing the graph edit distance is NP-Hard [220], and even though there are optimized algorithms for its computation [1], it will not be feasible for explanation datasets with a large number of exemplars. One way to overcome the complexity is to simplify the problem, and to work with *sets* instead of graphs, which will allow us to use an algorithm similar to the one presented in [59] for the computation of explanations. Of course converting a graph into a set without losing information is not generally possible. In this work, we convert the connected components of exemplars on the ABox graph into **sets of sets** of concepts, by rolling up the roles into concepts. Specifically, we add information about *outgoing edges* to the label of each node in the ABox graph, by defining new concepts $\exists r.C$ for each pair of role name r and concept name C, and then adding

$\exists r.C$ to the label of a node a if $r(a,b), C(b) \in \mathcal{A}$ for any $b \in \mathsf{IN}$. Then every exemplar of the explanation dataset is represented as the set of labels of nodes that are part of the connected component of the exemplar on the ABox. For instance, an exemplar e with a connected component:

$$\mathcal{A}_e = \{\mathsf{Exemplar}(e), \mathsf{depicts}(e,a), \mathsf{depicts}(e,b), \mathsf{depicts}(e,c), \\ \mathsf{Cat}(a), \mathsf{eating}(a,b), \mathsf{Fish}(b), \mathsf{in}(b,c), \mathsf{Water}(c)\}$$

would be represented as the set of labels (ignoring the Exemplar node): $\{\{\mathsf{Cat}, \exists \mathsf{eating}.\mathsf{Fish}\}, \{\mathsf{Fish}, \exists \mathsf{in}.\mathsf{Water}\}, \{\mathsf{Water}\}\}$.

Now, to compute counterfactual explanations, we have to solve a *set edit distance* problem between *concept set descriptions* of exemplars.

Cost of Edits

Before solving the edit distance problem we first have to determine how much each edit costs. Intuitively, we want counterfactual explanations to be semantically similar exemplars, thus the cost of an edit should reflect how much the exemplar changes semantically after applying the edit. Furthermore, the edits should be as transparent as possible in order to provide the user with comprehensive explanations. For instance, if the distance among concepts equals the distance between their embedded representations provided by a word embedding system or a graph neural network, we would not know why these concepts are close or distant. Thus it is imperative to use a transparent method for this calculation. To do this, we utilize the information that is present in the TBox. For the first type of ABox edits, that involves replacing concept assertions ($e_{A \to B}$), we assign a cost to the replacement of concept A with concept B equal to their distance on the TBox graph, ignoring the direction of the edges. For example, given a TBox: $\mathcal{T} = \{\mathsf{Cat} \sqsubseteq \mathsf{Mammal}, \mathsf{Dog} \sqsubseteq \mathsf{Mammal}, \mathsf{Ant} \sqsubseteq \mathsf{Insect}, \mathsf{Mammal} \sqsubseteq \mathsf{Animal}, \mathsf{Insect} \sqsubseteq \mathsf{Animal}\}$ the cost of replacing a $\mathsf{Cat}(a)$ assertion with $\mathsf{Mammal}(a)$ would be 1, the cost of replacing $\mathsf{Cat}(a)$ with $\mathsf{Dog}(a)$ would be 2 and the cost of replacing $\mathsf{Cat}(a)$ with $\mathsf{Ant}(a)$ would be 4. Similarly, the cost of replacing a role assertion $r(a,b)$ with $s(a,b)$ (symbolized $e_{r \to s}$) is assigned to be the distance of the shortest path on the undirected TBox graph from r to s. It is worth mentioning that this is not necessarily the optimal way to compute semantic similarities of concepts and roles, and other measures exist in the literature [33], which we plan to experiment with in future work.

For the insertion of concept or role assertions, as is apparent from the notation $e_{\top \to a}$, we assign a cost equal to the distance of the inserted atom (either a role or a concept) from the $\top$ node in the TBox graph. This means that it is more expensive to insert more specific atoms, than more general ones. Similarly for the deletion of atoms $e_{a \to \top}$, the cost is assigned to be the distance of the deleted concept or role from the $\top$ node on the undirected TBox graph meaning it is more expensive to delete more specific concepts and roles.

Based on the above, when dealing with *concept set descriptions* instead of graphs, where we have rolled up the roles into $\exists r.C$ concepts, the cost of inserting or deleting an $\exists r.C$ concept to/from a set is equal to the cost of inserting or deleting both a role assertion $r(a,b)$ and a concept assertion $C(b)$ so the cost would be the sum of the costs of $e_{\top \to r}$ and $e_{\top \to C}$. In this case, a concept C cannot be immediately replaced with a concept $\exists r.D$, instead it has to first be deleted $e_{C \to \top}$ and then the new concept inserted $e_{\top \to \exists r.D}$. On the other hand, replacing a concept $\exists r.C$ with $\exists s.D$ ($e_{\exists r.C \to \exists s.D}$) is equivalent to replacing a role assertion $r(a,b)$ with $s(a,b)$ and a concept assertion $C(b)$ with $D(b)$ so the cost of replacement would be the sum of the costs of $e_{r \to s}$ and $e_{C \to D}$.

Finally, we allow a user to manually assign cost to edits which could be useful in specific applications where some edits might not be feasible in the real world. For example, if we had exemplars representing people, and concepts representing their age ($\mathsf{Young}, \mathsf{Old}$) we might want to disallow the edit $e_{\mathsf{Old} \to \mathsf{Young}}$ as it would require time-travel in order to be implemented realistically, so we could assign an infinite cost to this edit.

Additional Criteria for Good Counterfactuals

In the context of this framework, the simplest counterfactual explanation for an exemplar e and a target class C would be the exemplar x (along with the edits) that is the closest with respect to edit distance to e while considering exemplars that are classified to C. If we have access to the output probabilities of the classifier for each class, then we can utilize this information and provide additional criteria to determine which counterfactual explanations to show to a user.

Target Significance The first additional criterion, defined as *significance* in [59], is to find the exemplar x that maximizes the fraction $\frac{P_C(x)}{\text{edit\_distance}(e,x)}$, where $P_C(x)$ is the probability for exemplar x to be classified to target class C. Intuitively, we are searching for a small set of low-cost edits (minimize edit_distance) that largely effect the output of the classifier for the desired class C (maximize $P_C(x)$).

Source-Target Significance Another option for a criterion would be to also take under consideration the prediction probability for the class that the original exemplar is classified to. Similarly to before, a counterfactual for exemplar e would be exemplar x (along with the edits) that maximizes the fraction $\frac{P_C(x)-P_D(x)}{\text{edit\_distance}(e,x)}$, where D is the class that e is classified to. Counterfactual explanations are supposed to answer the question "Why class **D** and not class **C**?", and while the previous criteria emphasize the "...and not class **C**" part of the question, intuitively source-target significance puts more weight on the "Why class **D**" part.

Entropy A final criterion we explore in this work is to consider the *confidence* of the classifier for classifying an exemplar to the target class C. As a measure of confidence we use the entropy at the output of the classifier, where a lower value indicates a more confident prediction. To do this, we find exemplar x that is classified to target class C and maximizes the fraction $\frac{\sum_{i\in\mathcal{C}} P_i(x)\log P_i(x)}{\text{edit\_distance}(e,x)}$, where $\mathcal{C}$ is the set of classes of the classifier.

Algorithm

In the general case, the algorithm for computing counterfactual explanations has two steps. The first step (preprocessing) is to compute the edit path between all pairs of exemplars in an explanation dataset and to acquire predictions of the classifier on all exemplars, including the prediction probabilities if they are available. The second step is, given an exemplar and a target class, to find the exemplar with the minimal edit distance that is classified to the target class that maximizes (or minimizes) the chosen criterion, out of those mentioned in section 3.5.1. Regarding complexity, when using the graph representation, as mentioned before graph edit distance computation is NP-Hard and it has to be done $|\mathsf{EN}|^2$ times. In our experiments we use the implementation provided by the python package networkx [1]. In our experiments we use a depth-first graph edit distance algorithm proposed in [1]. For the case of *concept set descriptions*, first we need to find the connected components of exemplars on the ABox graph. Then we need to add $\exists r.C$ concepts to the labels of nodes a for which $r(a,b)C(b)$ is in the ABox.

To compute the set edit distance between two labels of nodes ℓ_a, ℓ_b, each of which is a set of concepts (either atomic or of the form $\exists r.C$), we first construct a bipartite graph where each element of ℓ_a is connected to every element of ℓ_b and has a cost based on the TBox $\mathcal{T}$, as defined in section 3.5.1. On this bipartite graph we then compute the minimum weight full match using an implementation of Karp's algorithm [100] for the problem to get the optimal set of edits from one set of concepts to another. Finally, to compute the edit distance between two sets of labels L_1, L_2, each of which is **a set of sets** of concepts, we first compute the edit distance from each label in L_1

[1] https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.similarity.optimize_graph_edit_distance.html

Algorithm 5: Explanation Graph Construction

Data: A classifier F, an explanation dataset D, an undirected TBox Graph G_T
Result: Explanation Graph G_E

```
//the explanation graph will have a node for each element in the explanation dataset
Initialize Directed Graph G_E = (V_E = D, E_E = ∅);
foreach (x_i, C_i) ∈ D do
    foreach (x_j, C_j) ∈ D  {(x_i, C_i)} do
        Initialize Graph G_C = (V_C = C_i ∪ C_j, E_C = ∅);
        foreach k ∈ C_i do
            foreach l ∈ C_j do
                //Compute concept distance using TBox graph
                d_T(k, l) = |ShortestPath(G_T, k, l)|
                //Add an edge to G_C with weight d_T
                E_C = E_C ∪ {(k, l, d_T)}
            end
        end
        //Compute minimum weight full matching of the bipartite graph G_C
        {(c_m, c_n)}, w = MinFullMatch(G_C)
        //Concept Set Edit Distance
        D_T(C_i, C_j) = w
        //Compute criterion
        σ(i, j) = criterion(x_i, x_j)
        //Add an edge to the explanation graph G_E with weight 1/σ and as a label the edits
          corresponding to the minimum weight full match
        E_E = E_E ∪ {(v_i, v_j, 1/σ(i,j), {e_{c_m → c_n}})}
    end
end
return G_E
```

to every label in L_2 by using the procedure described in the previous paragraph for each pair of labels, meaning the set edit distance computation is performed $|L_1||L_2|$ times. Then to find the edit distance between L_1 and L_2 we use the same procedure as with sets of concepts (bipartite graph and full match), but this time the weights of the edges of the bipartite graph are assigned according to set the edit distance. Having preprocessed the explanation dataset and saved the edit paths, an explanation can be provided in $O(|\mathsf{EN}|)$. The result of the preprocessing stage is what we call an explanation graph, where the shortest paths indicate counterfactual explanations. The construction of this graph for concept set descriptions is outlined in algorithm 24.

Complexity

Regarding complexity, when using the graph representation, as mentioned before graph edit distance computation is NP-Hard and it has to be done $|\mathsf{EN}|^2$ times. For the case of *concept set descriptions*, first we need to find the connected components of exemplars on the ABox graph which requires time $O(|\mathcal{A}|)$. Then we need to add $\exists r.C$ concepts to the labels of nodes a for which $r(a,b)C(b)$ is in the ABox, which also requires time $O(|\mathcal{A}|)$. To compute the set edit distance between two labels of nodes ℓ_a, ℓ_b, each of which is a set of concepts (either atomic or of the form $\exists r.C$), we first construct a bipartite graph where each element of ℓ_a is connected to every element of ℓ_b and has a cost based on the TBox $\mathcal{T}$, as defined in section 3.5.1. The computation of the cost of a single edit can be done with Dijkstra's algorithm on the TBox graph, requiring time $O(v + |\mathcal{T}|\log(v))$, where $v = |\mathsf{CN}| + |\mathsf{RN}|$ thus the construction of this bipartite graph requires

time $O(|\ell_a||\ell_b|(v + |\mathcal{T}|\log(v)))$. On this bipartite graph we then compute the minimum weight full match using an implementation of Karp's algorithm [100] for the problem with time complexity $O(|\ell_a||\ell_b|\log(|\ell_b|))$ to get the optimal set of edits from one set of concepts to another. Finally, to compute the edit distance between two sets of labels L_1, L_2, each of which is **a set of sets** of concepts, we first compute the edit distance from each label in L_1 to every label in L_2 by using the procedure described in the previous paragraph for each pair of labels, meaning the set edit distance computation is performed $|L_1||L_2|$ times. Then to find the edit distance between L_1 and L_2 we use the same procedure as with sets of concepts (bipartite graph and full match), but this time the weights of the edges of the bipartite graph are assigned according to set the edit distance. Following from the above, for computing the edit distance between all pairs of exemplars, the total preprocessing time ends up being $O(|\mathsf{EN}|^2(L^2(l^2(v + T\log v + \log l)) + L^2\log L))$ where $|\mathsf{EN}|$ is the number of exemplars, L is the maximum number of nodes in a connected component of an exemplar, l is the maximum cardinality of a label of a node, T is the size of the TBox and v is the number of atomic concepts and roles. Having preprocessed the explanation dataset and saved the edit paths, an explanation can be provided in $O(|\mathsf{EN}|)$.

3.6 Discussion: Other Usage of Explanation Datasets

Explanation datasets can be useful for a multitude of different problems besides explainability. In this section we showcase two instances where explanation datasets were used for a task besides explainability: a) Improving performance of Information Retrieval (IR) systems, and b) Evaluating IR systems.

Regarding improving performance of IR systems, in our work in [43], we were tackling the problem of retrieving scientific documents given a text query. Specifically, in the context of the COVID-19 pandemic, new papers were being published daily, and key questions about the pandemic still did not have clear answers. Thus, it was imperative to develop tools to sift through the publications and get sources that answer a specific question, such as "What are the comorbidities of the virus?". A common approach for text-document retrieval is to use large language models to get vector representations of queries and corpus, and then using the cosine similarity of the vector representations, as a proxy to semantic similarity. Thus, given a query, the most semantically similar document is retrieved.

There are several pitfalls to this approach, and we focused on two of them. Firstly, especially since the documents were scientific papers, and the COVID-19 pandemic was at its early stages, the large language models will not have been trained on similar texts, there will be out-of-vocabulary words, and words appearing in a different context. Secondly, most language models are limited by the number of tokens they accept at their input, so getting a vector representation of an entire document is not trivial. The overall pipeline of the system is summarized in figure 3.3, however the details are out of scope of this dissertation. We see however, that the document relevance ranking procedure, takes into consideration, besides the document vector representation and the query vector representations, a semantically enriched representation for each. These "semantically enriched" representations, are sets of concepts, that appear in SNOMED-CT, a large knowledge graph of clinical terms.

Essentially, what we had was an explanation dataset, where we had individuals (exemplars), that were mapped to documents, and were labeled with a set of concepts (ABox), that were defined in SNOMED-CT (TBox). We hoped to use this knowledge to mitigate the first pitfall, of out-of-vocabulary words, since many words that a language model cannot understand (such as medical terminology) will appear in the semantic descriptions of the exemplars. Thus, our idea was to first filter documents based on the number of common concepts (including parents from the SNOMED hierarchy) with the query, before comparing vector representations. This had the added benefit of fewer cosine similarity comparisons to return an answer, as many documents would have been filtered out for not having common concepts with the query, in addition to not being constrained by

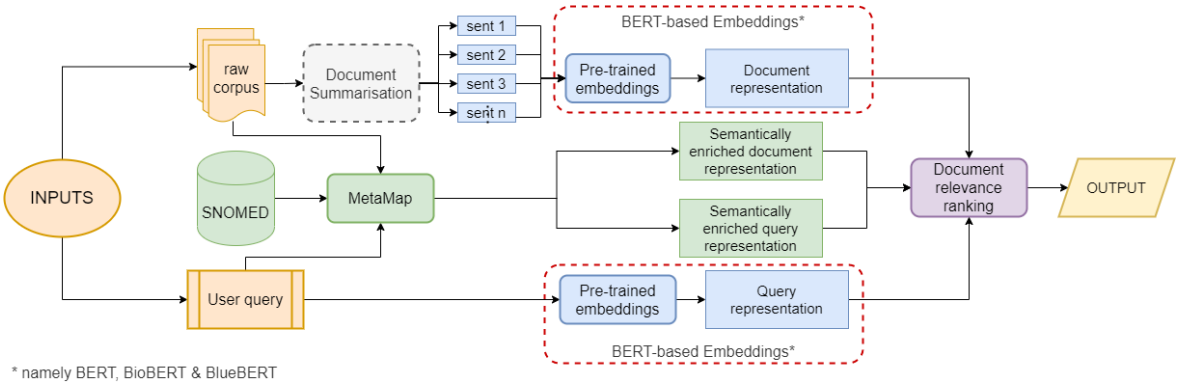


Figure 3.3: System architecture for COVID-19 document retrieval

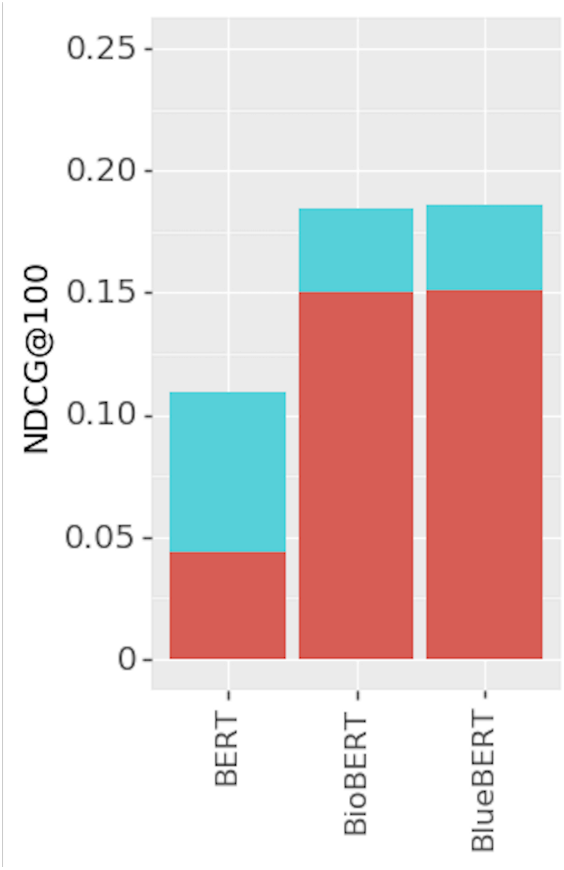


Figure 3.4: Performance improvement for NDCG@100 over initial BERT models with the addition of SNOMED-based filtering. The improvement is shown in light blue.

the length of the document, as concepts were extracted from the entire text. This led to a significant improvement in performance, especially for generic language models such as BERT [44] (as opposed to BioBERT [115] and BlueBERT [154] that have been fine-tuned on biomedical and clinical data respectively), as shown in figure 3.4.

The second task where an explanation datasets proved useful was evaluating IR systems, in our work in [129]. The typical metrics used for evaluating IR, involve a set of ground truth query-answer pairs, such as the Normalized Discounted Cumulative Gain (NDCG) metric shown in figure 3.4. Other evaluation metrics include precision at the n first documents, recall at n, mean reciprocal rank, median rank, etc. However, these numbers can be quite superficial, they do not explain why a particular system performs poorly or well, and their dependence on the ground-truth assumes that

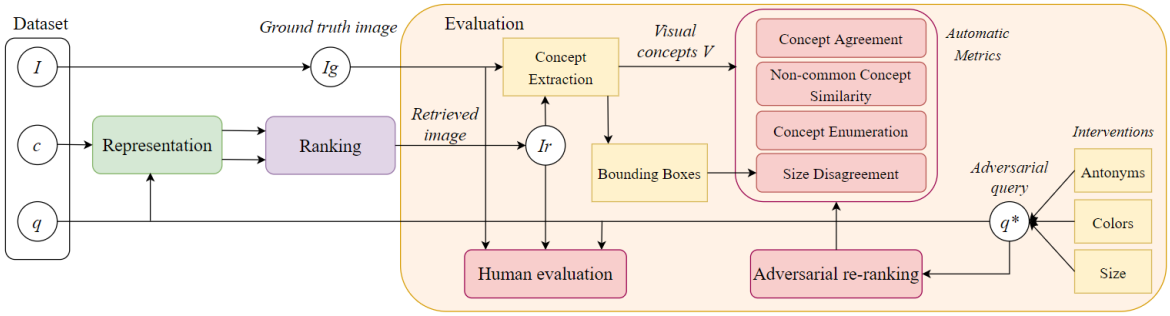


Figure 3.5: Evaluation pipeline for text-image retrieval

the ground-truth ranking is always reliable.

In our work we were concerned with the evaluation of text to image retrieval systems. Specifically, we aimed to develop metrics that quantify specific aspects of these systems, such as their ability to consider particular semantics, such as color, enumeration, size, etc. The pipeline of the system is shown in figure 3.5, the details of which our out of the scope of this dissertation. Firstly, again in this pipeline, we have essentially developed an explanation dataset. Specifically, we have available images annotated with concepts (such as COCO and the Visual Genome as shown in experiments in chapter 4). Then, given a text-image retrieval system, we can use the explanation dataset for comparing the retrieved image with the ground truth, based on the underlying knowledge. Based on this comparison (that is very similar to our counterfactual explanation computation of section 3.5.1), we can score the retrieval system, based on the conceptual similarities between retrieved and ground-truth images.

3.7 Conclusion

Using knowledge for explaining black-box systems makes intuitive sense. When the only option we have is to feed the black-box inputs, and observe its outputs, then having useful, and plentiful knowledge about the inputs, and the outputs, can lead to the extraction of useful, and plentiful information about the operation of the black-box. The main limitation of this approach, as it was presented in this chapter, seems to be computational complexity. Both the exponential KGrules and the NP-Hard graph edit distance version of the counterfactual generation algorithm, are prohibitively resource intensive. Thus, future efforts will be focused on developing optimizations, adaptations, and entirely different approaches for computing rule-based and counterfactual explanations. An example of such an approach could be to utilize knowledge embeddings [202] or graph embeddings [22] as components of algorithms such as algorithm 2, and 24, where for example instead of the graph edit distance, we could use a cosine similarity of vector representations of the graphs.

Furthermore, the most widely accepted method of evaluating explanation methods in the literature, are user studies. However, there is no consensus for the best way to conduct these studies, so part of our future endeavours involves conducting a human study to quantify the qualities of knowledge-based explanations, when compared to more traditional methods. In addition, we are exploring ways for utilizng explanation datasets, to detect specific aspects of other explanation systems, such as what semantics they can encode. For example, saliency maps highlight important pixels for a prediction. We might then be able to use an explanation dataset to determine why pixels might be highlighted as important. It could be their location in the image, it could be their color, their texture, or a combination.

We are also exploring ways to extend this framework to include more types of explanations besides rule-based and counterfactual. Specifically, of interest are prototype and criticism explanations [103], in which representative examples (prototypes) for each class are computed, along with “interesting” outliers (criticisms), where the examples could be chosen based on the defined

knowledge in the context of an explanation dataset. Finally, as we discuss in section 3.6, explanation datasets can be useful tools for a variety of tasks, thus, we should explore what constitutes a "good" explanation dataset, and methodologies for developing them by combining domain experts, existing resources, and knowledge extraction.

Chapter 4

Semantic Explanations of Image Classifiers

4.1 Introduction

In this chapter we apply the proposed framework for the purpose of explaining image classifiers. Starting from a controlled setting with a clear experimental objective (section 4.2), we progressively evolve the experimental setting, moving to the utilization of knowledge for explaining real-world state-of-the-art classifiers (section 4.3). We also explore the efficacy of automatic knowledge extraction for the purpose of explainability, and provide qualitative and quantitative results.

Image classification is one of the most popular machine learning tasks, with hundreds of benchmarks, thousands of papers, and a constantly evolving state-of-the-art. This task is broadly applicable, with applications such as self-driving cars [60], medical imaging [23], and the sciences, such as geology [123], meteorology [177], and astronomy [134]. It is also one of the first tasks for which claims of superhuman performance was reported [88]. There are known issues with most deep-learning based image classification methods, such as robustness to adversarial attacks [49], and since its applications can be decision critical (e.g. self-driving cars, medical imaging), explainability becomes crucial.

There is a large variety of explainability methods for image classification, including many domain-agnostic approaches such as LIME [165], Anchors [166], and example-based prototype/criticism explanations [103]. There are also domain specific methods, and even model-specific, such as Grad-CAM [179], and counterfactual visual explanations [72]. An important issue with explainability in the computer vision domain, is the vocabulary used in explanations i.e. pixel values. These are not really understandable to humans, whose mental model typically breaks down images into high level concepts, instead of low-level pixels, and as a result many such approaches have been criticized as being misleading [172, 141].

An important family of explainability methods for image classifiers, are concept attribution methods [104, 73]. These provide explanations in terms of human understandable concepts, instead of low-level pixels, mitigating many of the issues with other approaches, but they still suffer from their own pitfalls. An important one, which our work manages to overcome, is their reliance on white-box access to the classifier under investigation. This is an important caveat, because it means that these explainability methods cannot be applied to explain pure black-box models, such as those that are proprietary, which one might argue are the most important ones to be able to explain.

By applying our framework on multiple different classifiers, image classification tasks, and settings, we are able to generate semantic explanations, and we show their worth -both qualitatively and quantitatively.

4.2 CLEVR-Hans: Synthetic Images, Pre-determined Bias

In our first set of experiments, we employ CLEVR-Hans3 [188] which is a dataset of images with intentionally added biases in the train and validation set which are absent in the test set. Specifically, the dataset depicts scenes containing 3D geometric objects of different shapes, colours, sizes, materials and locations, and the images are split into three classes. The first class contains images

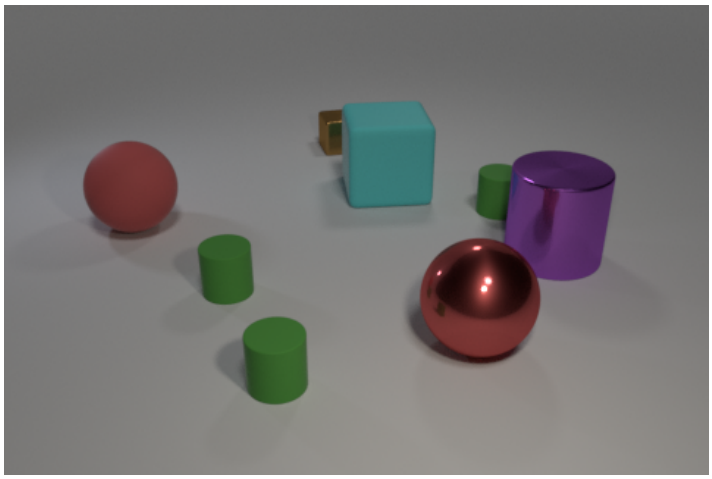


Figure 4.1: An image from the CLEVR-Hans3 dataset.

that depict a large cube and a large cylinder, but the large cube is always gray in the training and validation sets, while it has a random color in the test set (intentionally added bias - confounding factor). The second class contains images that depict a small metal cube and a small sphere, where the sphere is always metal in the training and validation sets, but random in the test set, while the third class contains images depicting a large blue sphere and a small yellow sphere, and has no confounding factors. The existence of these intentionally added, well defined biases makes the dataset ideal for the evaluation of XAI frameworks since it leads to classifiers with foreknown biases which we can attampt to detect.

4.2.1 Explanation Dataset

The first step for applying the proposed framework is creating the explanation dataset, which involves representing available information as a Description Logics Knowledge Base. To this end, we define an individual name for each image and for each object depicted therein, and a concept name for each color, size, shape and material of the objects. We also include a role name contains, to connect images to objects they depict. Then, in the ABox, we assert the characteristics of each object and link them to the appropriate images by using the role. For example, in Fig. 4.1 we can see a sample image from the CLEVR-Hans3 dataset with id i, which is described in the ABox of our explanation datasets with the assertions: {Exemplar(i), contains(i,o_1), Red(o_1), Sphere(o_1), Large(o_1), Rubber(o_1), contains(i,o_2), Green(o_2), Cylinder(o_2), Small(o_2), Rubber(o_2), ... }. In this case, the TBox of the corresponding knowledge base is empty. Using this representation, we created multiple explanation datasets: One from the training set, in order to compare our approach with methods that are meant to be applied on the training set (such as FACE [157], and several explanation datasets of varying sizes from the test set of CLEVR-Hans3. Throughout our experiments, we attempt to provide explanations for a ResNet34 model that was trained on the training set, and as is apparent from the confusion matrix shown in Table 4.1 the classifier likely learned the intentional biases, as shown from the poor performance on the two confounded classes (1 and 2), as oposed to the third class.

4.2.2 Rule-based Explanations

KGrules

Using the true labels of the data allows us to use the description of each class as ground truth explanations. Table 4.2 shows a condensed version of the explanation rules produced by KGrules (Alg. 1) for an ideal classifier (accuracy=100%) with explanation datasets of various sizes, along with

ground truth and the correct explanation rule with highest recall per class for a real classifier. The full explanations are obtained by adding to that condensed versions the conjuncts Exemplar(x), and contains(x,t), for all other appearing variables $t \neq x$, as well as the tail of the rule ($\rightarrow ClassX$) for the respective class X. All explanations on the ideal classifier achieved recall=100%. We can see that with explanation datasets with 600 or more exemplars we are able to predict the ground truth for all 3 classes. Even with 20 exemplars we are able to produce the ground truth explanation for one of the classes and with 40 or more exemplars we produce ground truth explanations for 2 out of the 3 classes and almost for the third class too (only one characteristic of one object missing). In order to produce accurate explanations it seems useful to have individuals close to the "semantic border" of the classes, i.e. individuals of different classes with similar descriptions. Intuitively, such individuals guide the algorithm to produce a more accurate explanation in a similar manner that near-border examples guide a machine learning algorithm to approximate better the separating function. Following this intuition, we experiment with two of the small explanation datasets that almost found the perfect explanations (size of 40 and 80). By strategically choosing individuals , we are able to obtain two small explanation datasets, one of size 43 and one of size 82, that when used by Alg. 1 produce the ground truth explanations for all 3 classes. This indicates the importance of the curation of the explanation dataset, which is not an easy task, and the selection of "good" individuals for the explanation dataset is not trivial.

After observing the explanations produced by our method for the ideal classifier, we were also able to detect the foreknown biases of the classifier due to the confounding factors of the dataset. To this end, we curated an explanation dataset of 100 individuals that accurately produces the ground truth rule. Then, running the KGrules algorithm we acquire the explanation rules shown at the bottom of table 4.2. For example, regarding the first class (all images contain a large cube and a large cylinder), the rule with the highest recall produced for the real classifier is: $\mathsf{contains}(x, y)$, $\mathsf{Gray}(y)$, $\mathsf{Large}(y)$, $\mathsf{contains}(x, z)$, $\mathsf{Cylinder}(z)$, $\mathsf{Large}(\mathsf{z}) \rightarrow \mathsf{Class}_1(x)$ showing the existence of a large cylinder, and detecting the potential color bias of another large object created by the intentional bias of the train and validation set (the large cube is always gray in the train and validation sets).

KGrules-H

Regarding KGrules-H, that also produces explanation rules with exceptions contrarily to KGrules, the best rule computed for each metric on the entire test set of CLEVR-Hans3, for the ResNet34-based model is shown in table 4.4. The algorithm found a correct rule (precision = 1) for each class, in addition to a rule query with recall = 1, whose certain answers are a superset of the positive set. The best degree was achieved for class 3, which lacks a confounding factor, meaning the classifier is not expected to be biased. Correct rule queries are of particular interest since they can be translated into guaranteed IF-THEN rules which the classifier follows on the particular dataset. For instance the highest recall correct rule query for class 1 is translated into the rule "If the image contains a Large Gray Cube, a Large Cylinder and a Large Metal Object then it is classified to class 1.". This rule clearly shows the bias of the classifier, since it is the description of the class with the added confounding factor (the Large Cube is Gray). Similarly the (not correct) rule query with recall = 1

Table 4.1: Performance of the ResNet34 model on CLEVR-Hans3.

	Test set metrics			Confusion matrix		
True label	Precision	Recall	F1-score	Class 1	Class 2	Class 3
Class 1	0.94	0.16	0.27	118	511	121
Class 2	0.59	0.98	0.54	5	736	9
Class 3	0.85	1.00	0.92	2	0	748

Table 4.2: Explanations on CLEVR-Hans3. The concepts in parentheses are the confounding factors in the ground truth row. The symbol (☒) indicates that the explanation is the same with the ground truth (without the confounding factors).

Nr. of images	Class 1	Class 2	Class 3
20	☒	Small(y), Metal(y), Cube(y)	Yellow(y), Small(y), Blue(z), Large(z), Sphere(z)
40 / 60	☒	☒	Yellow(y), Small(y), Blue(z), Large(z), Sphere(z)
80 / 100 / 200 / 400	☒	☒	Yellow(y), Small(y), Sphere(y), Blue(z), Sphere(z)
600 / 800 / 1000	☒	☒	☒
Ground Truth	(*Gray(y)*) , Large(y), Cube(y), Large(z), Cylinder(z)	Small(y), (*Metal(y)*), Sphere(y), Small(z), Metal(z), Cube(z)	Yellow(y), Small(y), Sphere(y), Blue(z), Large(z), Sphere(z)
Real Classifier	Large(y), Cube(y), Gray(z), Large(z), Large(w), Cylinder(w)	Small(y), Metal(y), Cube(y)	☒

for the same class can be translated into the rule "If the image does not contain a Large Cube then it is not classified to class 1", since the set of certain answers is a super set of the positive set. We observed that correct rule queries tend to be more specific than others, with the most general rules with exceptions being those with recall = 1. Other rules which were correct with exceptions, tended to lie somewhere in the middle with respect to how general or specific they are, but they were the ones which lead to the highest values of degree. By observing these results, we concluded that in practice, a set of rules, both correct and with exceptions, can give us a very clear picture of what the black-box classifier is doing. However, in order to not overwhelm an end-user with a large number of rules, we should develop a strategy to select which rules to show to the user, that is out of the scope of this dissertation, but is a priority for future work.

It is interesting to note that the rule query with recall = 1 produced for class 1 contained a Large Cube but not a Large Cylinder, which is also in the description of the class. This shows that in the training process the classifier learned to pay more attention to the presence of cubes rather than the presence of cylinders. The elements of the highest recall correct rule that differ from the true description of class 1 can be a great starting point for a closer inspection of the classifier. We expected the presence of a Gray Cube from the confounding factor introduced in the training and validation sets, but in a real world scenario similar insights can be reached by inspecting the queries. In our case, we further inquired the role that the Gray Cube and the Large Metal Object play in the correct rule by removing either of them from the query and examining its behavior. In Table 4.3 we can see that the gray color was essential for the correct rule while the Large Metal Object was not, and in fact its removal improved the rule and returned almost the entire class.

Another result that piqued our attention was the highest degree explanation for class 3 which is the actual rule that describes this class. This explanation was not a correct rule, since it had two exceptions, which we can also see in the confusion matrix of the classifier and we were interested to examine what sets these two individuals apart. We found that both of these individuals are answers to the query "y1 is Large, Gray, Cube". This showed us once again the great effect the confounding factor of class 1 had on the classifier.

Our overall results show that the classifier tended to emphasize low level information such as color and shape and ignored higher level information such as texture and the combined presence of multiple objects. This was the reason why the confounding factor of class 1 had an important effect to the way images were classified, while the confounding factor of class 2 seemed to have had a much smaller one. Furthermore, the added bias made the classifier reject class 1 images, which however had to be classified to one of the other two classes (no class was not an option). Therefore one of the other classes had to be "polluted" by samples which were not confidently classified to a class. This motivates us to expand the framework in the future to work with more informative sets than the pos-set, such as elements which were classified with high confidence, and false and true, negatives and positives.

Table 4.3: Two modified versions of the class 1 correct rule produced by removing conjuncts.

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

4.2.3 Counterfactual Explanations

Using the explanation dataset constructed from the test set of CLEVR-Hans3, we also used algorithm 24 to generate counterfactual explanations. In figure 4.2 the importance of concepts for for images which were classified in class 1, with the target class being class 0, is shown (as per equation 3.4). The bias of the classifier is immediately detected for the confounded class 0. As mentioned previously, the confounding factor for class 0 is that the Large Cube is always Grey in the train set. This is apparent from the first three bars of the plot on the left, where the most important insertions seem to be the concepts: (Gray, GrayLargeCube, GrayLarge). The reason for which GrayLargeCube has a larger importance than GrayLarge is because, for some local counterfactuals, GrayLarge objects (which are not necessarily Cube) might be removed, thus lowering the importance of this concept.

In fig.4.3 we show local counterfactual explanations generated for three randomly selected images (first column), which were classified in class B (Small Metal Cube and Small Sphere - where the Small Sphere is always Metal in the train set) and with the target class being class A (Large Cube, Large Cylinder, where the Large Cube is always Grey in the train set). The second column shows the suggestions of the FACE algorithm and the third column shows the suggestions of our

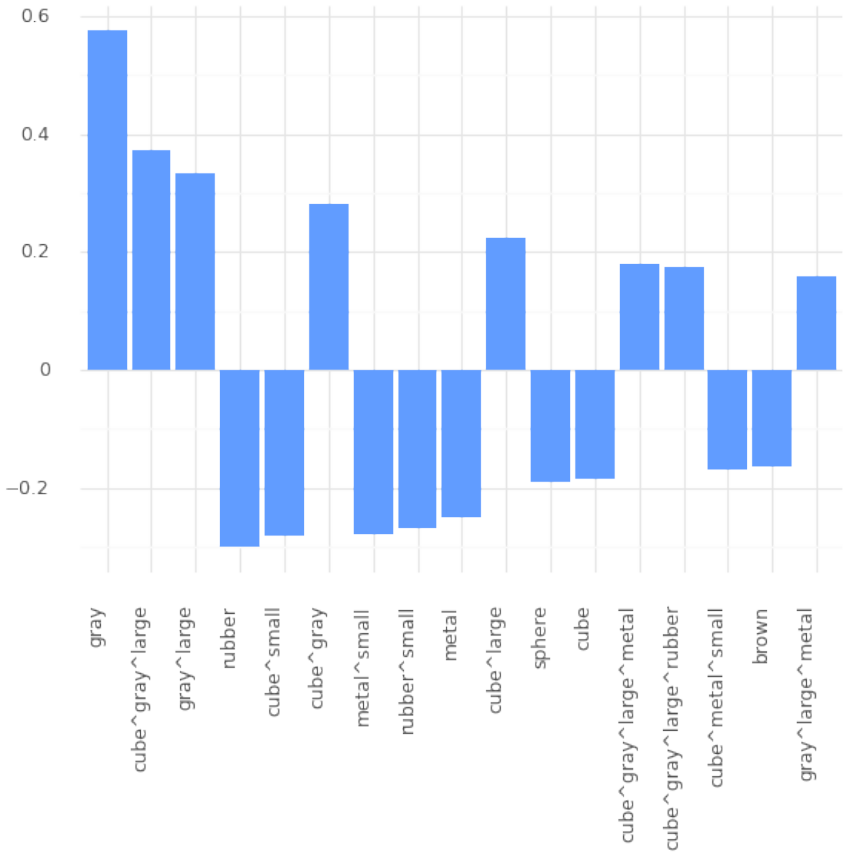


Figure 4.2: Global explanation for the subset of CLEVR-Hans3 which is classified in class B, with target class A

algorithm. At first glance, neither results are very intuitive, and we argue that the form of the explanations (sequence of samples from the training set) is the reason. Note that the counterfactual images generated within our framework would normally be accompanied by the edits themselves, not shown here for brevity. A more thorough observation reveals that our approach tends to keep the number of objects in an image constant, which is due to the high cost of adding and deleting concepts rather than replacing them, while FACE, which relies on the distribution of the dataset, operating on a pixel-level and having no knowledge of the objects depicted, tends to transition to images that contain a large number of objects.

This experiment demonstrates the usefulness of the proposed method for detecting biases, in addition to the efficacy of the overall framework, that allows us to produce explanations, without having to consider the complex space of pixels, but instead leverage human understandable, semantic information.

4.3 Real World Images, State-of-the-art classifiers

In this set of experiments we focus on explaining classifiers of real-world images, specifically those trained on the ImageNet [37, 110], and Places [223] datasets. Furthermore, we apply our ideas on more specialized settings, such as CUB [200] classification of bird species, and MNIST [114] for hand-written digits.

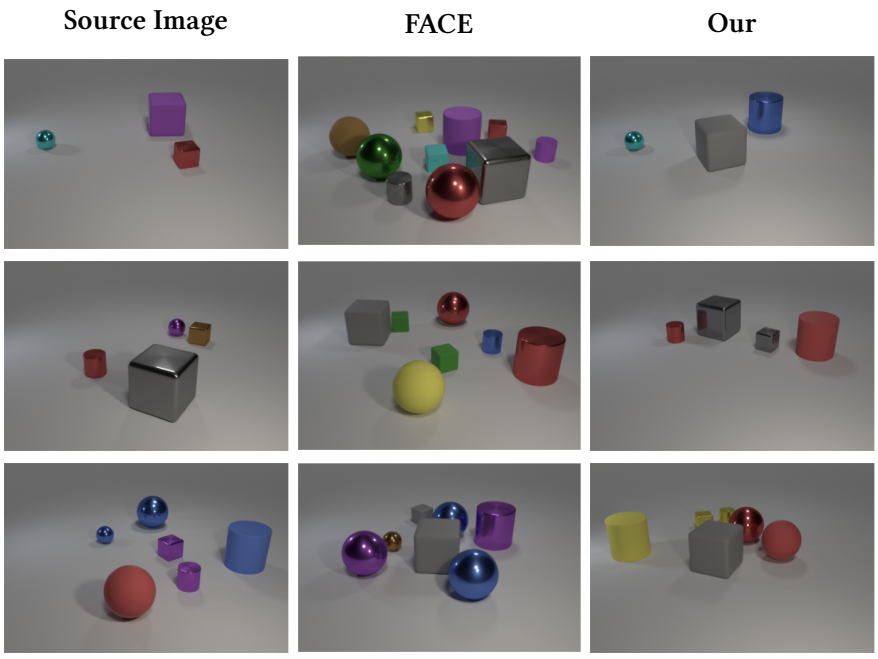


Figure 4.3: Counterfactuals for 3 images (first column) which classified in class B with target class A, using FACE (second column) and our proposed method (third column)

4.3.1 Explanation Datasets

For applying the proposed framework, the main requirement is the existence of an explanation dataset. We experimented with multiple different such datasets, and ways of acquiring them.

COCO, Visual Genome and WordNet

COCO The first explanation dataset we created from a subset of COCO [124], which contains real-world images, annotated with objects, which we can automatically link to external knowledge such as WordNet. This allows us to use the simpler, and more efficient *set* version of the counterfactual explanation algorithm. Specifically, we gathered images pertaining to two classes: "Restaurant" related and "Bedroom" related images. For the *restaurant-related* class we gathered all images from COCO that contained the concepts: 1. {dining table, person, pizza} (1000+ images) 2. {dining table, person, wine glass} (1200+ images). For the *bedroom-related* class we gathered all images that contained the label combinations of: 1. {bed, person} (1300+ images) 2. {bed, book} (800+ images) 3. {bed, teddy bear} (300+ images). On top of that, we wanted to make sure that we included some images that might be puzzling for the classifier. Those images were the ones including COCO label combinations of: 1. {bed, fork} (10 images) 2. {bed, spoon} (20 images) 3. {bed, wine glass} (20 images) 4. {bed, pizza} (10 images) 5. {dining table, bed} (170 images). For each image in COCO, a description of the objects present in that image is provided. To create the explanation dataset, we automatically linked these object descriptions with WordNet synsets by using the NLTK python package[1]. We used WordNet synsets as the set of concept names CN, and the hyponym-hypernym hierarchy as a TBox. The explanation dataset contains only one role ($|\mathsf{RN}| = 1$) that links images (exemplars) to objects they depict, for example:

$$\mathcal{A} = \{\mathsf{Exemplar(a)}, \mathsf{depicts}(a, b), \mathsf{Dog(b)}\}, \mathsf{Brown(b)} \ldots$$

$$\mathcal{T} = \{\mathsf{Dog} \sqsubseteq \mathsf{Canine}, \mathsf{Canine} \sqsubseteq \mathsf{Mammal} \ldots\}$$

[1] https://www.nltk.org/howto/wordnet.html

Visual Genome We also utilizez the Visual Genome dataset (VGD) [109] which contains richly annotated images, including descriptions of regions, attributes of depicted objects and relations between them, leading to a graph representation for the knowledge describing each image (opposed to a set representation for COCO). Specifically, similarly to the COCO case, we represent the available VGD annotations as a Description Logics Knowledge Base, where the ABox consists of the scene graphs for each image, in which each node and edge is labeled with a WordNet (WN) synset and the TBox consists of the WN hypernym-hyponym hierarchy. In the ABox we also include assertions about which objects are depicted by an image in order to connect the exemplar data with the scene graphs. For example, the image in Fig. 4.4 with id i is described in the ABox by the assertions: {Ex-

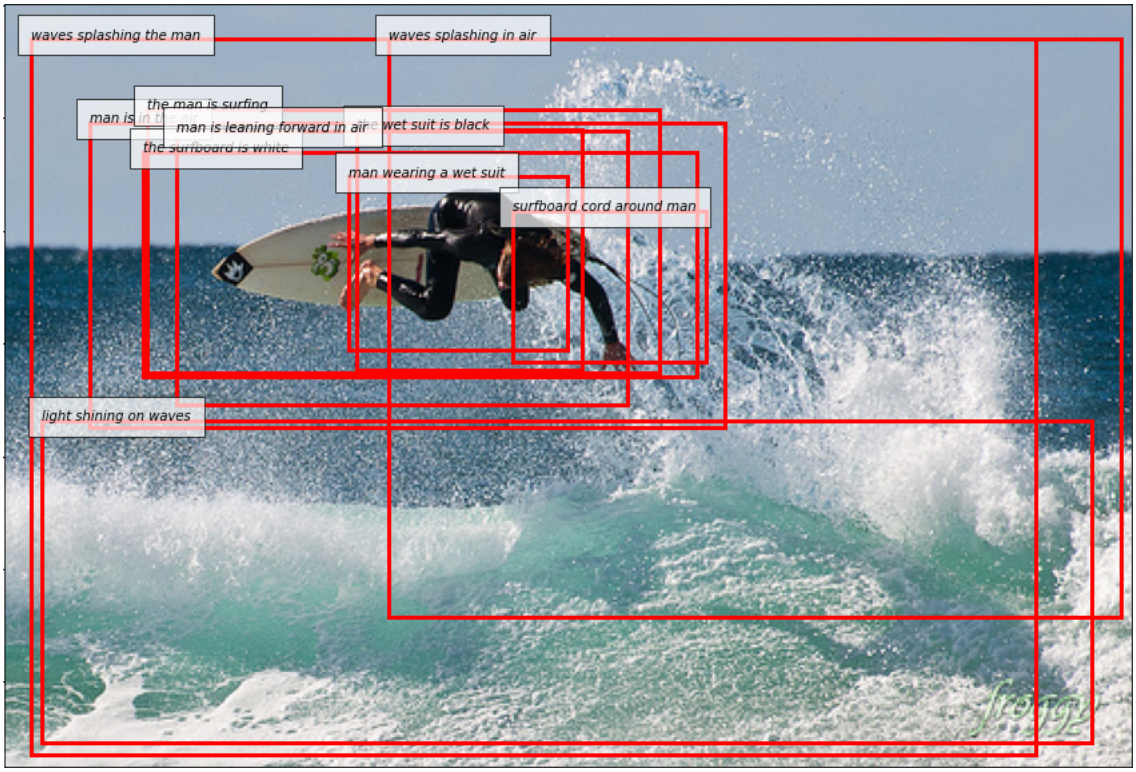


Figure 4.4: An image from the Visual Genome dataset.

emplar(i), contains(i,person$_1$), contains(i,sea$_1$), surfer.n.01(person$_1$), ocean.n.01(sea$_1$), blue.s.01(sea$_1$), travel.v.01(person$_1$,sea$_1$)} , and the TBox contains the axioms: {ocean.n.01 $\sqsubseteq$ body_of_water.n.01, surfer.n.01 $\sqsubseteq$ swimmer.n.02, ...}.

Since in the original VGD annotations are linked to wordnet automatically, there are errors, thus we chose to manually curate a subset of 100 images. This is closer to the intended use-case of our proposed method, in which experts would curate explanation datasets for specific domains. The relatively low number of images, when compared to the other explanation datasets is justified, as the intent was to run the high-complexity KGrules and graph edit counterfactual algorithms, which would not really be feasible for larger explanationd datasets (opposed to KGrules-H and the set edit counterfactual algorithm).

WordNet WordNet [137] is a lexical database and hierarchical semantic network designed to model the organization of words and their meanings in English. Synsets are the fundamental building blocks of WordNet. They represent groups of words that are synonymous or nearly synonymous, sharing a common meaning. For example, the synset for the word “car” in WordNet might include terms like “automobile”, “motorcar”, and “auto”. All these words are considered synonyms within this specific context. Synsets are organized into a hierarchy, based on their semantic relationships, where more general terms are defined as hypernyms of more specific ones (hyponyms). For example

"vehicle" is a hypernym of "wheeled vehicle", while "car" is a hyponym of "wheeled vehicle".

The structure of WordNet can easily be represented as the TBox of a description logics knowledge base, after first appropriately defining concept and role names (one for each synset), and encoding the hierarchy using subsumption axioms. For example

$$\mathcal{T} = \{\mathsf{car} \sqsubseteq \mathsf{wheeledvehicle}\}, \mathsf{wheeledvehicle} \sqsubseteq \mathsf{vehicle}\}$$

In our experiments we constructed the TBox from the entirety of WordNet that contains more than 100,000 synsets, however most do not appear in the ABox.

CUB

CUB is a dataset that contains images of birds across 200 different species. Each image is annotated with 15 bird parts, that we use as semantic descriptions for constructing the explanation dataset. For each bird part, there are multiple different attributes, that are rolled up into concepts, leading to an explanation dataset with $|\mathsf{CN}| = 312$. For example, from the annotation $\mathsf{hasWingColor} :: \mathsf{Blue}$ we define the concept BlueWingColor, and assign it to the appropriate exemplar. In this case, the TBox is empty.

Scene Graph Generation and Feature Extraction

We also constructed two explanation datasets by using automatic methods of creating the semantic descriptions.

For the first, we used RelTR: Relation Transformer for Scene Graph Generation [31] as a Scene Graph Generator and run it on Google Colab, using the default model parameters. The predicted classes for the scene graph nodes were 150 entities, and 50 relationship classes from WordNet. Furthermore, one prediction is considered valid if their confidence is greater than 0.3. The scene graph generator was used to create an explanation dataset from images "in the wild". Specifically, we searched the web for images satisfying our criteria and divided them into two classes, namely "driver" and "pedestrian". We did this for motorbike and bicycle riders since we want to avoid the role name being itself the descriptor of the class, e.g. "person driving car". We queried Google, Bing and Yahoo images for a combination of keywords containing "people", "motorbikes" and "bicycles", gathered the following creative-commons photographs, and manually split them into two classes. 1. {driver class} (63 images of people on bicycles and 127 images of people on motorbikes) 2. {pedestrian class} (31 images of people and parked motorbikes, 38 images of people and parked bicycles). Once we constructed our dataset, we extracted semantic descriptions with the scene graph generator. The ABox and TBox of this dataset use the same vocabulary and are similar in structure to the Visual Genome explanation dataset.

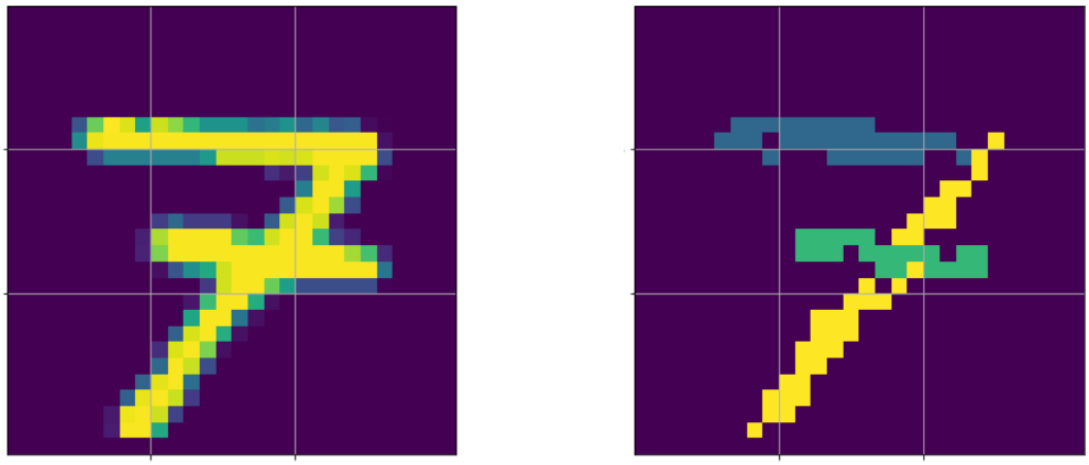
For the second, we applied a ridge detection algorithm [125] on MNIST to describe images as a collection of intersecting lines, varying in angle, length and location within the image. In Fig. 4.5 we show an example of an MNIST image, along with the results of the aforementioned information extraction procedure using ridge detection. After this procedure, we encode the information in a knowledge base, suited for use in an explanation dataset, where the vocabulary IN, CN, RN and the Tbox of our knowledge base are the following:

$$\mathsf{IN} = \{\mathsf{test\_zero1}, \mathsf{test\_zero1\_line0}, \dots, \mathsf{test\_zero1\_line7}, \mathsf{test\_zero6}, \mathsf{test\_zero6\_line0}, \dots, \mathsf{test\_nine979\_line7}, \dots, \mathsf{lineM}_{250}\}$$

$$\mathsf{CN} = \{\mathsf{Image}, \mathsf{Line}, \mathsf{Line0deg}, \mathsf{Line45deg}, \mathsf{Line90deg}, \mathsf{Line135deg}, \mathsf{TopLeft}, \mathsf{TopCenter}, \mathsf{TopRight}, \mathsf{MidLeft}, \mathsf{MidCenter}, \mathsf{MidRight}, \mathsf{BotLeft}, \mathsf{BotCenter}, \mathsf{BotRight}, \mathsf{Short}, \mathsf{Medium}, \mathsf{Long}\}$$

$$\mathsf{RN} = \{\mathsf{contains}, \mathsf{intersects}\}.$$

$$\mathcal{T} = \{C \sqsubseteq \mathsf{Line} \mid C \notin \{\mathsf{Image}, \mathsf{Line}\}\}.$$



```
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
```

Figure 4.5: An example of a digit, the results of ridge detection, and the corresponding description.

4.3.2 Rule-based Explanations

Explaining ImageNet Classifiers

We explain three different neural architectures[2]: VGG-16 [183], Wide-ResNet (WRN) [217] and ResNeXt [211], trained for classification on the ImageNet dataset, by using the curated visual genome explanation dataset. We define three super-classes of ImageNet classes which contain a) Domestic, b) Wild and c) Aquatic Animals , because they are more intuitive to perform a qualitative evaluation, when compared to the fine-grained ImageNet classes generated with the KGrules algorithm. Table 4.5 shows the correct rules of maximum recall for each class and each classifier. We discuss three key explanations:

1. Wide ResNet: surfboard(y) $\rightarrow$ *Aquatic(x)*. It seems that the classifier has a bias accepting surfer/surfboard images as aquatic animals probably due to the sea environment of the images; further investigation finds this claim to be consistent, showing the potential of this framework in detecting biases.

2. Wide ResNet: animal(y), wear(y, z), artifact(z) $\rightarrow$ *Domestic(x)*. It is interesting to compare this explanation with another correct rule for the same classifier with lower recall: animal(y), collar(z) $\rightarrow$ *Domestic(x)*. By considering roles between objects we get a more accurate (higher recall) and informative explanation, denoting the tendency of the classifier to classify as *Domestic* any animal that wears something man-made. This example shows how more complex queries enhance the insight (wearing an artifact) while less expressive ones might only see a part of it (collar). Here we can also see one of the effects of the TBox hierarchy on the explanations, since this rule covers many sub-cases (like dog wears collar, and cat wears bowtie) that would require multiple rules if it wasn't for the grouping that stems from the TBox.

3. ResNeXt: nose(y), plant(z), ear(w) $\rightarrow$ *Wild*. Although this explanation provides information that is related to the nature environment of the images classified as *Wild* (plant), we see also some rather odd concepts (nose, ear). While this could be a strange bias of the classifier, it is probably a flaw of the explanation dataset. As we discovered, images are not consistently annotated with body parts, like noses and ears. Thus, through the explanations we can also detect weaknesses of the explanation set. The rules are limited by the available knowledge, so we should constantly evaluate

[2] https://pytorch.org/vision/stable/models.html

the quality and expressivity of the knowledge that is used in order to produce accurate and useful explanations.

Explaining a PLACES Classifier

For the case of the Places365 [223] dataset and KGrules-H, as a black-box classifier we used the *ResNet50* classifier[3] provided by the official GitHub repository[4] for models pretrained on Places365, which classifies images to 365 different classes[5]. The top1 error is 44.82% and the top5 error is 14.71%. We used the confusion matrix to select the two most confused classes to generate explanations for, which were "Desert Sand" and "Desert Road". The best rule queries for each metric for each of the two classes is shown in Table 4.6. For both classes the generated queries have some unexpected conjuncts despite decent performance with respect to the three metrics. For example the conjunct giraffe.n.01(y3) appearing in the best correct rule for "Desert Road", and the best degree rule for the same class being simply Image(x), depicts(x, y1), animal.n.01(y1). Furthermore, for "Desert Sand", the concept road.n.01 stands out. The concept communicaton.n.02 has the hyopnyms sign.n.02 and written_communication.n.01 in the WordNet hierarchy which could refer to license plates and traffic signs. The second best query in terms of precision was Image(x), depicts(x, y1) ,depicts(x, y2), depicts(x, y3).motor_vehicle.n.01(y1), sky.n.01(y2), road.n.01(y3), along.r.01(y1, y3). Again the concept motor_vehicle.n.01 stands out. Given that the second highest value in the confusion matrix of this classifier was for the pair "Desert Sand", "Desert Vegetation" we conjencture that some mistakes were made during the training of this classifier or several images in the training set of the Places365 dataset are mislabeled. The classifier may have been fed images that should be described as "Desert Road" but with the target label being "Desert Sand" and images that should be described as "Desert Vegetation" but with the target label being "Desert Road". This would explain the weird associations exhibited and the conjuncts appearing in the explanations. It is worth mentioning at this point, that we do not see a way that this peculiar behaviour of the classifier could have been discovered by using different explanation techniques, that do not utilize explanation datasets, highlighting the usefulness of the proposed framework.

Explaining an MNIST Classifier

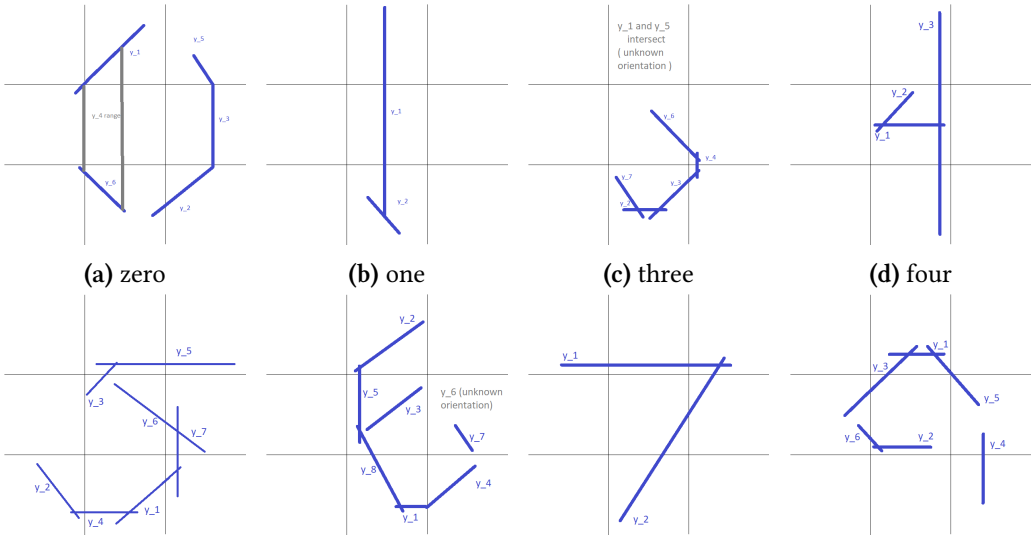
Using the MNIST explanation dataset where we utilized ridge detection for automatically generating semantic descriptions, and KGrules-H to generate rules, we were able to produce rule-based explanations for an example network provided by PyTorch[6]. For example, for the digit zero the rule consisted of conjuncts: contains(x, y_1), contains(x, y_2), contains(x, y_3), contains(x, y_4), contains(x, y_5), contains(x, y_6). For five of the six lines, the explanation rule query included their location in the image, indicated by the conjuncts TopCenter(y_1), BotRight(y_2), BotCenter(y_2), MidRight(y_3), TopRight(y_5), BotCenter(y_6). For all six lines the explanation rule included information about their orientation, indicated by the conjuncts Line45deg(y_1), Line45deg(y_2), Line90deg(y_3), Line90deg(y_4), Line135deg(y_5), Line135deg(y_6).

Finally, the rule-query included the following conjuncts which show which lines intersect each other intersects(y_1, y_4), intersects(y_2, y_3), intersects(y_3, y_5), intersects(y_4, y_6).

Such a rule would not be as good of an explanation as the ones shown in previous experiments, since it contains a large number of conjuncts, and the terminology is not immediately understandable. By visualizing the rules however, we can get some useful explanations that might help us to understand the classifier, as shown in figure 4.6. An observation we made, is the fact that some conjuncts were more understandable than others when they were part of explanation rules. For instance, knowing a line's location and orientation was imperative for understanding the rule via

[3] PyTorch model: http://places2.csail.mit.edu/models_places365/resnet50_places365.pth.tar
[4] https://github.com/CSAILVision/places365
[5] https://github.com/zhoubolei/places_devkit/blob/master/categories_places365.txt
[6] https://github.com/pytorch/examples/tree/master/mnist

visualization, while conjuncts involving line intersections and sizes seemed not that important, regardless of metrics. This is something which could be leveraged either in explanation dataset construction (for example domain experts weigh concepts and roles depending on their importance for understandability), or in algorithm design (for example a user could provide as input concepts and roles which they want to appear in explanation rules). We are considering these ideas as a main direction for future work which involves developing strategies for choosing which rules are best to show to a user. This experiment shows how the prerequisite for producing good explanations in this framework, using an explanation dataset, depends almost entirely on the properties and quality of the explanation dataset.



(a) zero **(b)** one **(c)** three **(d)** four

(e) five **(f)** six **(g)** seven **(h)** nine

Figure 4.6: Visualizations of best recall correct rules for digits

4.3.3 Counterfactual Explanations

Explaining a PLACES Classifier

We also provide counterfactual explanations for the PLACES classifier by using the proposed framework, and the explanation dataset created from the object annotations of COCO. In Figures 4.7, 4.8 we see two examples of global counterfactual explanations on the COCO dataset. As before, each bar's numeric value shows the importance of the insertion (positive) or removal (negative) of that specific concept, in the process of transforming from a source region of an explanation dataset, to a target class.

Without revealing the source region and the target class for each figure, we can try to work out what those are, just by looking at the most frequent additions and removals. On the first (fig.4.7), which is the more trivial of the two, we see that the most common removals from the source images were concepts relevant to {furniture, bed, animal, carnivore, dog}, while the most common additions were the concepts {home appliance, refrigerator, white goods, consumer goods}. From this, we can assume that the source region was likely bedroom images (with a bias towards pets) and the target class was probably a kitchen. The true classes were, indeed, "bedroom" and "kitchen". On the second (fig.4.8), we see that most frequent removals revolved around {instrumentality, artifact, electronic, furniture, telecommunications, TV, broadcasting, kitchen} and the most common additions around {carnivore, animal, mammal, feline, cat, dog}. Knowing that we are dealing with a classifier of rooms and places, we would probably guess a kitchen for the source and a location with domestic animals for the target. The actual classes were "bedroom" targeting "veterinarian", which raises an interesting question: why did we see "kitchen" instead of "bed" in the bedroom class? The answer

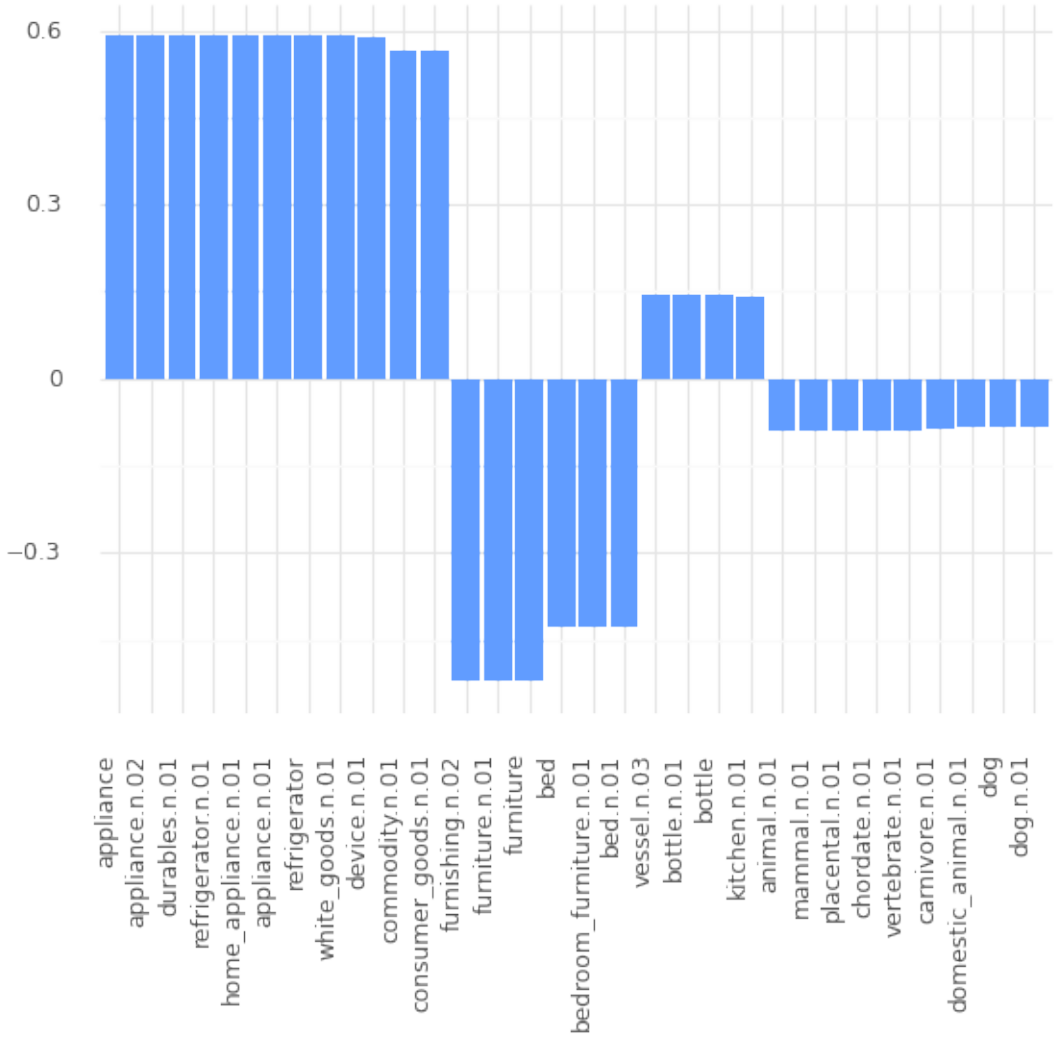


Figure 4.7: Generalized Counterfactual Explanations for the region of the explanation dataset for COCO which is classified as ”bedroom”, with the target class being ”kitchen”

is that no beds were actually removed, since veterinarian office images tend to include beds. On the other hand, our dataset contains a number of studio-apartment bedroom images which had part of the kitchen appearing in the photo - kitchens that are mostly missing from a vet's office and had to be removed. Another thing to note is that those examples were not cherry-picked. During our experiments we could, most of the time, estimate the source region and target class by looking at the edit frequencies. Notably, the most confusing results were when we tested the ”computer room” target and found out that the generalized counterfactual explanation was very often adding people, but never laptops or computers. After investigating what seemed like a bug, we realized that most images from our dataset which were classified as a ”computer room” had no computers in them, but people working in lab-appearing rooms.

In the first row of Figure 4.9 we show a local counterfactual explanation for an image classified as a “Bedroom” to the target class “Playhouse”, which requires only one Concept Edit ($e_{\top \to \text{Child}}$). This example is interesting because “Playhouse” is an erroneous prediction (the ground truth for the second image should be “Bedroom”), thus immediately we detect a potential bias of the classifier, that if a Child is added to an image of a “Bedroom” it might be classified as a “Playhouse”. Similarly, in the second row of Figure 4.9 we show a local counterfactual explanation for an image which is classified as “Bedroom” to the target class “Veterinarian's Office”, and the resulting target image is an erroneous prediction. The resulting edit is simply to add a Cat. Finally, in Figure 4.10 we

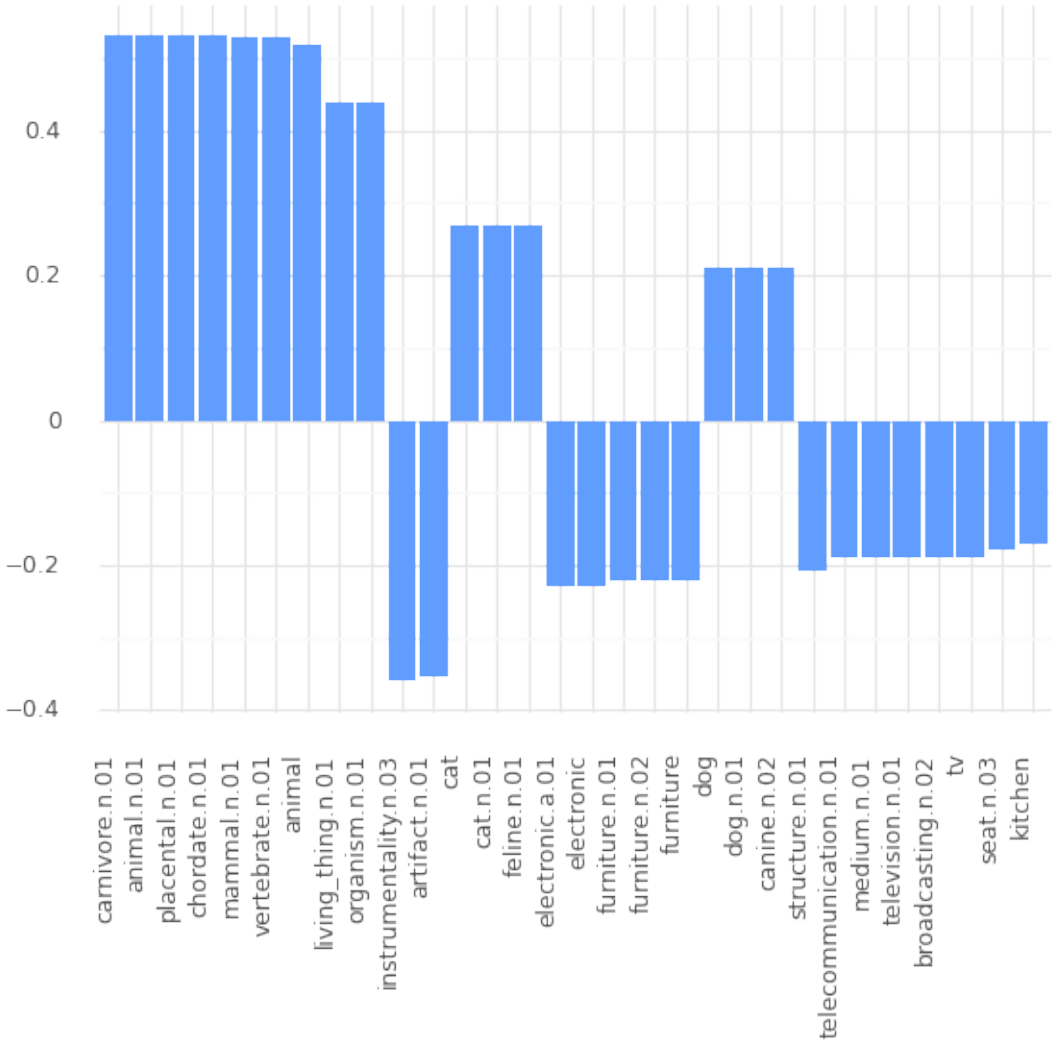


Figure 4.8: Generalized Counterfactual Explanations for the region of the explanation dataset for COCO which is classified as ”bedroom”, with target class ”veterinarian”

show a counterfactual explanation, where the path on the graph has two steps. The source image is classified as a “Bedroom” and the target class is “Computer Room”. This shows a smooth transition from the source image to the target class, by first adding a person (there are already two laptops in the source image), and then adding two more people and two more laptops.

This experiment demonstrates a more real-world use-case than CLEVR-Hans, in which we even detected unknown biases (for example the depiction of people was more important than that of laptops for the class “computer room”), and further insight into the classifier, which we had not thought about (for example that the classifier expects veterinarian’s offices to depict beds among other objects).

A crucial question at this point is how can we know where the biases that the counterfactual explanations uncover come from. We are assuming that they emerge from the classifier, but a biased explanation dataset could yield similarly biased results. A way to answer this question is to run the same task on a different dataset, to see how those results compare with the previous ones. As a cross-checking dataset, we will use Visual Genome since it is, along with COCO, one of the very few datasets containing semantically annotated images. The results of the Visual Genome experiment, overlayed on COCO’s results, are depicted in fig. 4.11. We can see that the classifier gave very similar predictions for both datasets , which validates the hypothesis that the biases did not arise from a possible irregular distribution within the explanation datasets but from the classifier itself.

Figure 4.9: Counterfactual explanation for changing the prediction of the image on the left from 'Bedroom' to 'Playhouse' is simply to add a child ($e_{\top \to \text{Child}}$) (top) and from 'Bedroom' to 'veterinarians office' is simply to add a cat ($e_{\top \to \text{Cat}}$) (bottom).

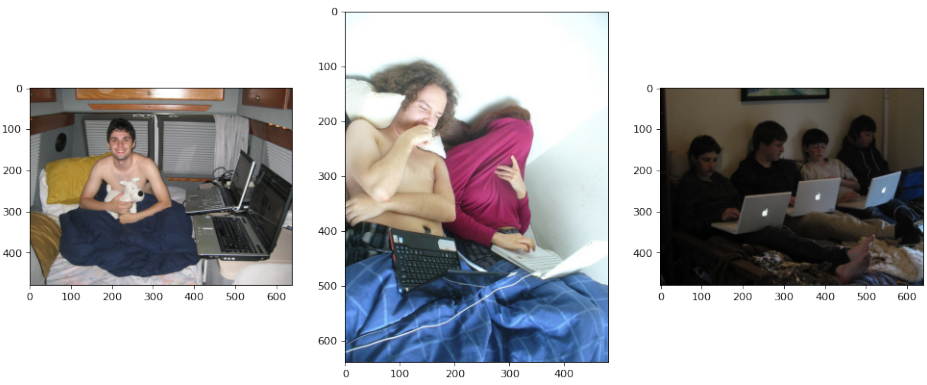


Figure 4.10: Counterfactual explanation for changing the prediction of the image on the left from "Bedroom" to "Computer Room", which requires two steps

Most of the important features that differentiated the classes in the previous experiment could be fully expressed by concepts alone, e.g. the existence of a bed or a dog. There are, though, many situations where this is not the case and where roles and relationships between objects should be taken into account. For example, classifying between "driver" and "pedestrian" classes on images containing the concepts "motorbike", "bicycle" and "person" cannot be done without knowing the relationship between the person and the vehicle. Thus, using the explanation dataset creating using the scene graph generator, we produced counterfactual explanations for the ground-truth labels of this manually curated set of images. The global counterfactuals transitioning from "pedestrian" to "driver", are depicted on fig.4.12 as concept set descriptions, i.e. concepts along with roles. The top addition by a very large margin is "$ride \hat{} wheeled\_vehicle$" as expected, which is the parent, and thus, the sum of "$ride \hat{} bicycle$" and "$ride \hat{} motorbike$". Next, we see additions of "$wearing \hat{} helmet$" and a smaller addition of the concept "helmet" by itself, presumably because in some driving photos the helmet was on the handlebars of the bike and not on the rider's head. We

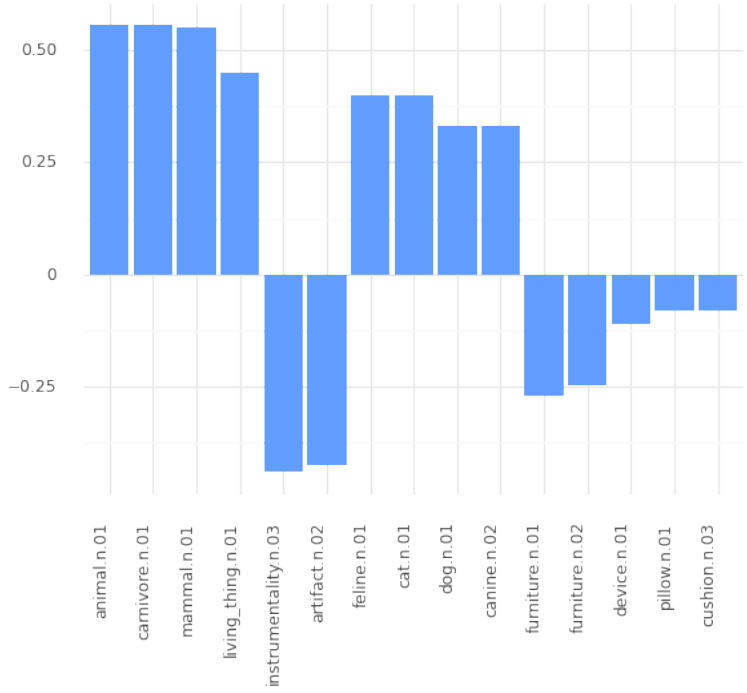


Figure 4.11: Global explanation for the subset of Visual Genome which is classified as "bedroom", with target class "vet"

also see that "$wear\hat{}hat$" is removed (the child of "$wear\hat{}clothing$"), which compliments the addition of "$wear\hat{}helmet$", and that "$have\hat{}seat$" is removed since bicycle seats are not visible when bikes are ridden. The rest of the edits are too scarce and, although we might be able to explain them, they can very likely be noise as well.

Human Evaluation on the CUB dataset

To assess how the counterfactual images retrieved by our algorithm fare against the state-of-the-art results [72], we set up a human study; since a widely accepted metric to evaluate the success of semantically consistent visual counterfactuals does not exist. For the human survey, we used the Label Studio platform [7], which offers high-level flexibility and functionality. A screenshot of the annotation page is depicted in the following Figure 4.13. The classifiers that we selected for this experiment had the same pretrained weights that [Vandenhende et al., 2022] used in their work. The 33 participants were mainly graduate students, and PhD candidates, who responded to the call for participation. They were not offered compensation, they were volunteers. No information was provided to the participants, besides the call for participation, and the instructions for the labeling procedure. The study was conducted online.

We first acquire two pre-trained classifiers (a VGG-16 [182], and a ResNet-50 [**he2016deep**]), and make predictions on the test set of CUB. This dataset is what we use as an *explanation dataset*, after encoding the annoatations of the images in a DL knowledge base.

In [196], the authors selected a number of bird images from the CUB dataset. Then, for each one, they retrieved its closest counterfactual image from the full dataset, with the restriction that it cannot belong to the same bird species (label) as the source. For our experiment, we executed the same methodology utilizing our algorithm to perform the same task on the same source images.

Then, to each of our 33 human evaluators, we presented a randomly selected source image along with its two corresponding counterfactual images - the one retrieved by the SOTA and by our algorithm. The evaluators were then asked which of the two counterfactual bird images more closely,

[7] https://labelstud.io/

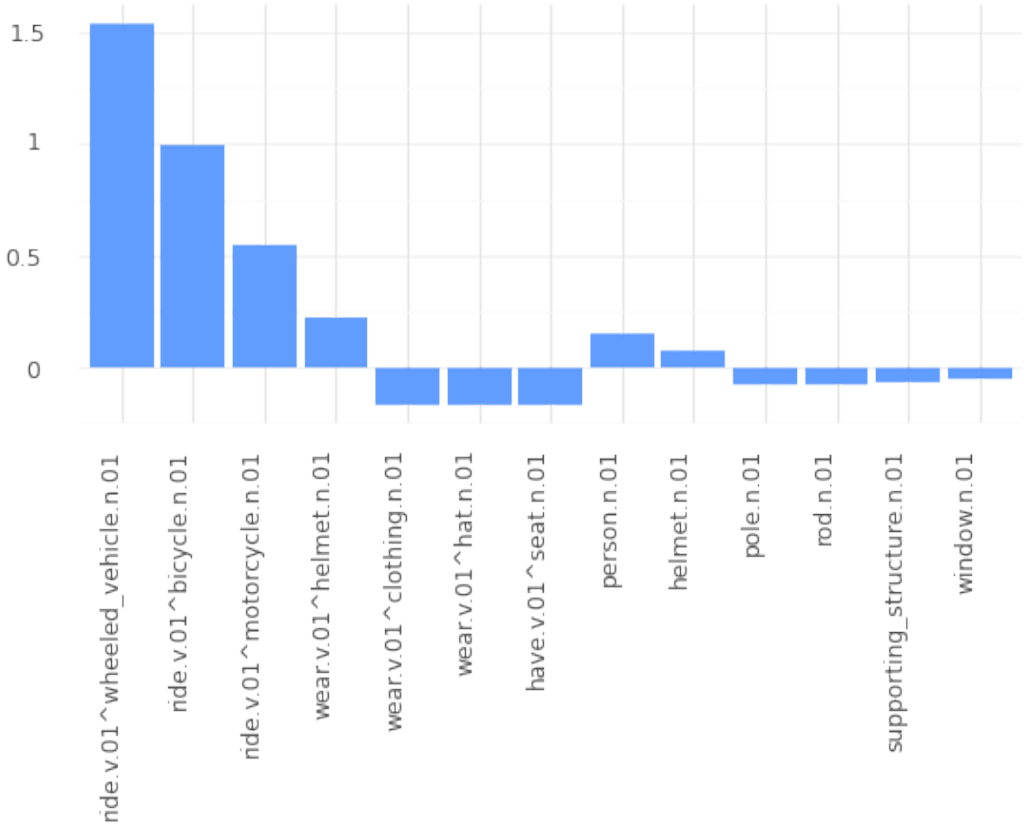


Figure 4.12: Flipping class form “pedestrian” to “driver”, the most important changes are: the addition of “ride^wheeled_vehicle”, “wear^helmet” and the removal of “wear^hat”.

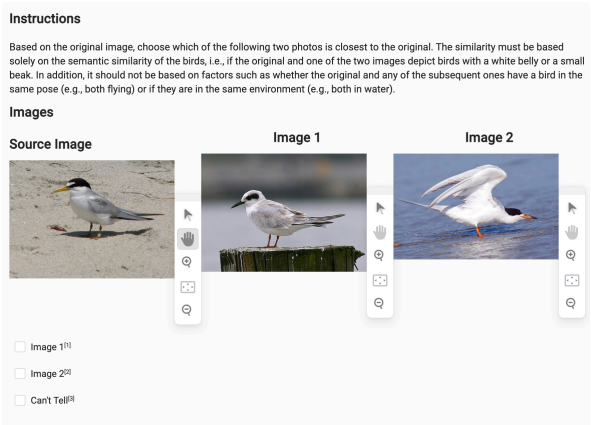


Figure 4.13: A screenshot from the annotating platform. The first image always depicts a source image, whereas the second and the third are randomly the counterexample produced by [Vandenhende et al., 2022] method and the proposed one.

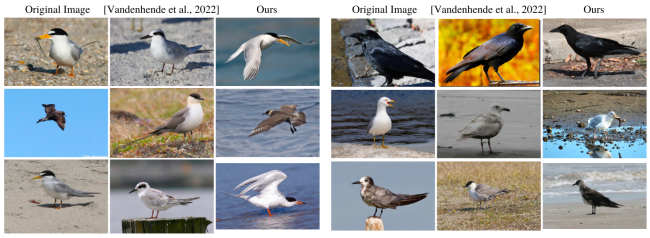


Figure 4.14: The first column shows the original image, the second one [196]'s retrieved image and the third one the image retrieved by our algorithm.

semantically, resembled the bird depicted in the first image (i.e. not taking into account the bird's posture or its background).

The images retrieved with both methods were largely similar and sometimes identical. As a result, the evaluators experienced difficulties deciding between the two counterfactual images, and the two methods achieved similar results (Table 4.7). It is important to note that our algorithm did not peek inside the model, contrarily to the SOTA algorithm. Our approach managed to attain equal results just by taking into account the semantic knowledge accompanying CUB images, without having white-box access to the classifiers. Further details about this experiment are available in the supplementary material[2].

4.4 Conclusion

Image classification, a corner stone of deep learning, and widely applicable, has also been widely explored regarding explainability. Few methods however are able to bridge the gap between the domain of pixels, and high level conceptual abstractions that humans use to comprehend images. Using the proposed framework, the explanations are *grounded* on human-defined, and human-understandable knowledge, and through our experiments we have shown how it can be used to produce useful and understandable explanations. Compared to other concept-based approaches, that mainly rely on "opening the black box" and utilizing the neural network's activations, the proposed approach relies solely on the explanation dataset. This means that all potential pitfalls of XAI, in our case, depend on the explanation dataset. As long as the explanation dataset is reliable and trustworthy, then the explanations will be also. Contrarily, the only way for the explanations to be misleading or in other ways problematic, is if these issues exist in the explanation dataset.

Table 4.4: Optimal explanations with regard to the three metrics on CLEVR-Hans3 produced by KGrules-H.

Metric	Explanation Rules	Precision	Recall	Degree	Positives
		Class 1			
Best Precision	y1 is Large, Cube, Gray. y2 is Large, Cylinder. y3 is Large, Metal.	1.00	0.66	0.66	83
Best Recall	y1 is Large, Cube.	0.09	1.00	0.09	125
Best Degree	y1 is Large, Cube, Gray. y2 is Large, Cylinder. y3 is Large, Metal.	1.00	0.66	0.66	83
		Class 2			
Best Precision	y1 is Small, Sphere. y2 is Large, Rubber. y3 is Small, Metal, Cube. y4 is Small, Brown. y5 is Small, Rubber, Cylinder.	1.00	0.09	0.09	116
Best Recall	y1 is Cube.	0.63	1.00	0.63	1247
Best Degree	y1 is Metal, Cube. y2 is Small, Metal.	0.78	0.8	0.65	1005
		Class 3			
Best Precision	y1 is Metal, Blue. y2 is Large, Blue, Sphere. y3 is Yellow, Small, Sphere. y4 is Small, Rubber. y5 is Metal, Sphere.	1.00	0.42	0.42	365
Best Recall	y1 is Large. y2 is Sphere.	0.42	1.00	0.42	878
Best Degree	y1 is Yellow, Small, Sphere. y2 is Large, Blue, Sphere.	0.99	0.85	0.85	748

Table 4.5: Explanation rules utilizing the animal explanation dataset. Rules are shown in condensed form: the full rules are obtained by adding the conjuncts contains(x,t) for all appearing variables $x \neq t$.

Network	Rules
VGG-16	$\mathsf{artifact}(y), \mathsf{dog}(z), \mathsf{brown}(w) \rightarrow \mathsf{Domestic}(x)$
	$\mathsf{green}(y), \mathsf{plant}(z), \mathsf{organ}(w) \rightarrow \mathsf{Wild}(x)$
	$\mathsf{whole}(y), \mathsf{ocean}(z) \rightarrow \mathsf{Aquatic}(x)$
WRN	$\mathsf{animal}(y), \mathsf{wear}(y, z), \mathsf{artifact}(z) \rightarrow \mathsf{Domestic}(x)$
	$\mathsf{green}(y), \mathsf{plant}(z), \mathsf{nose}(w) \rightarrow \mathsf{Wild}(x)$
	$\mathsf{surfboard}(y) \rightarrow \mathsf{Aquatic}(x)$
ResNext	$\mathsf{artifact}(y), \mathsf{dog}(z), \mathsf{brown}(w) \rightarrow \mathsf{Domestic}(x)$
	$\mathsf{ear}(y), \mathsf{plant}(z), \mathsf{nose}(w) \rightarrow \mathsf{Wild}(x)$
	$\mathsf{fish}(y), \mathsf{structure}(z) \rightarrow \mathsf{Aquatic}(x)$

Table 4.6: Optimal explanations produced by KGrules-H with regard to the three metrics using the VG explanation dataset.

Metric	Explanation Rules	Precision	Recall	Degree	Positives
	Desert Road				
Best Precision	y1 is field.n.01. y2 is natural_object.n.01. y3 is giraffe.n.01. y4 is body_part.n.01. y5 is woody_plant.n.01.	1.00	0.12	0.12	16
Best Recall	y1 is organism.n.01	0.54	1.00	0.54	139
Best Degree	y1 is animal.n.01	0.72	0.84	0.64	118
	Desert Sand				
Best Precision	y1 is instrumentality.n.03 y2 is road.n.01 y3 is communication.n.02 y4 is sky.n.01 y5 is tree.n.01	1.00	0.12	0.12	16
Best Recall	y1 is physical_entity.n.01	0.49	1.00	0.49	134
Best Degree	y1 is instrumentality.n.01 y2 is road.n.01	0.92	0.56	0.56	76

	ResNet-50	VGG-16
[196] S.O.T.A.	14.65%	13.68%
Ours	34.93%	23.65%
Can't Tell	**50.42%**	**62.67%**

Table 4.7: Human evaluation results on which of the two counterfactual bird images is semantically closer to the source image.

Chapter 5

Explainability and Evaluation of AI in the Domain of Symbolic Music

5.1 Introduction

Music is a domain that has been of interest to scientists and mathematicians for millennia, from the ideas of Pythagoras to modern mathematics [54], computational musicology [13] and AI generated music [45]. There have been numerous theoretical frameworks attempting to describe, analyze and explain music, developed by different cultures and during different time periods. In the current technological landscape, most practical applications regarding analysis and information extraction from music, rely on the abundance of data and machine learning, and are being used in the music industry more and more, as the technology matures. This includes recommender systems [24, 83], which are the driving component of most music streaming platforms where increasingly more music is consumed and discovered [144], music production software, such as AI powered audio mastering [16], and music education [204].

A first distinction for AI systems in the domain of music, is between creative and information extraction systems. The former tackle tasks such as music composition [128], accompaniment and continuation [164], style transfer and timbre change [53], synthesis [47] and more. The latter includes music information retrieval (MIR) tasks, such as chord recognition [151], genre classification [39], instrument recognition [186], beat tracking [70], audio tagging [18], music transcription [14], source separation [89] and more. Explainability of models is desired both for information extraction and for creative systems.

A second distinction for AI systems in the domain of music is between those that work with audio representations and those that work with symbolic representations of music. Audio representations are a digital waveforms stored in a WAV format, typically at a sample rate of 44100 Hz and a bit-depth of 16 bits (CD quality). It is a resource intensive format, since even one second of audio requires 44100×16 bits, or 88.2 kilobytes of memory. This makes it difficult to handle in the context of machine learning, which requires large amounts of data. Furthermore, typical sequential deep learning models cannot easily capture information across different time-scales of a waveform simultaneously, mainly due to the receptive field of the neural networks, which has led to the development of specialized architectures for audio [146]. These architectures however are still very resource intensive, and for this reason, most approaches for applying AI to audio of music work in the time-frequency domain, requiring a pre-processing/feature extraction stage in which Digital Signal Processing (DSP) methods are applied to transfer the audio information into a more workable representation. On the other hand, symbolic representations of music are typically much more lightweight, by encoding musical information using musical notation. The most common format for symbolically representing music, is the Musical Instrument Digital Interface (MIDI) [142], which is a protocol developed for controlling digital instruments. Other formats for symbolically representing music exist, such as musicXML [67], used for rendering musical scores in software such as Sibelius[1] and Musescore[2], and even simpler representations such as ASCII tablatures. These formats repre-

[1] https://www.avid.com/sibelius

[2] https://musescore.org/

sent the bare-bones musical information, such as what notes are being played, at what time (using musical time subdivisions), and with what velocity, and typically include meta-data, such as the tempo, time-signature, instruments etc. Symbolic representations of music are not audible, and to convert them to an audible format, synthesizers are required. This means that timbral characterstics of music, and nuanced aspects of music performance cannot be represented in this way.

In this chapter we describe our research involving both creative and information extraction systems in the domain of symbolic music. We show our approach for evaluating music composition models, based on notions from music theory, and describe the state-of-the-art model we developed for genre recognition from symbolic representations of music. For the latter, we also extensively studied *post hoc* explanations of the developed model, and compared to other systems that are explainable-by-design. This study of explanations is a main motivator for exploring explanations in terms of formally represented knowledge, in general. The chapter ends with ways of formally representing music-theoretical concepts using knowledge representation technologies, and a discussion about how this could aid both our evaluation approach, and the quality of explanations regarding symbolic music.

5.2 Evaluation of AI Generated Music

Evaluating musical creative systems is a difficult task, mainly due to the subjectivity of music, the unclear desiderata for a "good" creative model [153, 152], and the lack of evaluation metrics. Thus, the most widely accepted method of evaluation uses human studies. However, even these studies are not conclusive evaluation, as they depend on demographics, questions asked, time requirements, and more. For example BachBot [119], was evaluated via a "musical Turing test" [193]. They gathered demographics, and self-reported musical expertise from 721 participants, and asked them to differentiate between music composed by BachBot, and music composed by Bach. They conclude that BachBot is capable of composing music that the average participant cannot differentiate from Bach. Another example is the music transformer [95], which was evaluated by 180 comparisons between real data and AI generated music. In a third example, the creators of MuseGAN [48] ran a survey of 144 participants, where 44 of them were deemed "pros". They asked specific questions regarding i) harmony, ii) rhythm, iii) structure, iv) coherency, and v) overall score.

Other approaches to evaluating music generation systems, involve computing the log-likelihood on a real dataset (such as JSB Chorales [19]), that measures a model's ability to approximate the probability distribution of a dataset. Along these lines, the creators of MuseGAN propose a set of metrics that, that are computed on real and generated data, and their values are compared. These metrics are the ratio of empty bars in a composition, the number of pitches used, the number of pitch classes used, and the ratio of qualified notes (notes with a duration of at least a 32nd). They also propose a metric measuring drum patterns (ratio of notes in 8 or 16 beat patterns). Finally, they define Tonal Distance as an intra-track metric, measuring the harmonicity between tracks, where in the context of MIDI, a song consists of multiple tracks, typically one for each instrument. Tonal Distance is based on Euclidian Distance on a 6D space, where small distance represents harmonic proximity [86].

5.2.1 Tone Networks and Tonic Coordinate Systems

Inspired by the Tonal Distance intra-track metric, and the underlying musical structures, such as the Tonnetz, first proposed by Euler [30], we aspired to develop an evaluation methodology that does not depend on comparison with external data, as other evaluation metrics do [38]. The word "Tonnetz" translates to "tone network" from German. In our work we view such networks as knowledge graphs in which the twelve pitch classes are connected in some way that represents harmonic relationships between them.

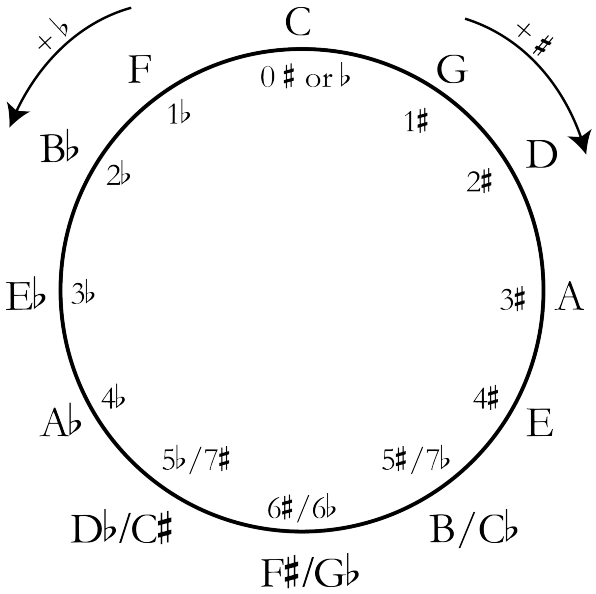


Figure 5.1: The circle of fifths, a 2-Degree tone network

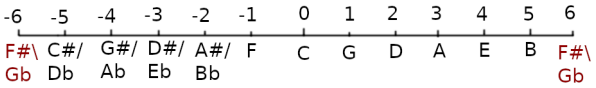


Figure 5.2: Tonic coordinate system from circle of fifths, with a tonic note C

Definition 7 (Tone Networks). *An N-Degree tone network is an undirected graph* (V, E)*, where* V *is the set of pitch classes and each vertex has exactly N neighbours (edges) of weight 1.*

For example, the circle of fifths (figure 5.1), one of the most widely used visualizations in music theory, can be viewed as a 2-Degree tone network, where each vertex representing a pitch class is connected with an edge to its perfect fifth and its perfect fourth. Another example is the Tonnetz (see figure 5.4), that can be viewed as a 6-Degree tone network, where each pitch class is connected to its perfect fourth, minor sixth/augmented fifth, minor third, perfect fifth, major third, major sixth. (clockwise).

A tone network may be used to define useful properties of sets of pitch classes, and the relationships between them. Examples of such properties are the span of vertices of a noteset, or the length of the minimum set of paths between two notesets. In our work, we transform tone networks of degree N into $\frac{N}{2}$-dimensional coordinate systems by arbitrarily assigning an origin and where each dimension represents an interval. We call the origin of the coordinate system the **tonic note**.

Definition 8 (Tonic Coordinate System). *An N-dimensional tonic coordinate system is a discrete Cartesian coordinate system in which every point represents a set of pitch classes. The origin of the coordinate system represents exactly one pitch class which will be called the tonic note.*

For example, the circle of fifths can be represented as a 1D coordinate system (figure 5.2). The perfect fifth interval has the property that every pitch class is accessible from every other pitch class by iteratively applying the perfect fifth. Specifically, since there are twelve pitch classes, the pitch classes that are reachable from a pitch class p by applying the interval v are:

$$\text{reachable}(p, v) = \{(p + n * v) \text{mod} 12\}, n \in \mathbb{N}$$

There are only four intervals that have this property that $|\text{reachable}(p, v)| = 12$. These are the perfect fifth ($v = 7$) and its complementary perfect fourth ($v = 5$), the semitone ($v = 1$), and its

complemantary major seventh ($v = 11$). These two intervals (fifth, semitone) and their complements also have musical significance. The perfect fifth above a note, in just intonation, represents a $\frac{3}{2}$ frequency ratio, which is the simplest ratio that appears between musical notes besides the octave that is an interval of 12 and leads to the same pitch class, with a $\frac{2}{1}$ ratio. This indicates that the interval "sounds the closest" to its origin. The fifth appears in most diatonic chords, with the exception of diminished chords which have a characteristically dissonant sound. The fifth interval is also often used in harmonic progressions to build tension and reach resolution, for example the authentic cadence goes from the fifth to the tonic, and the plagal cadence goes from the fourth to the tonic (which is a perfect fifth interval as the fourth is the complement of the fifth). On the other hand, the semitone ($v = 1$) is the smallest interval, and corresponds to the smallest frequency difference between notes. Musically, it has some common characteristics with the fifth, namely that it can be used to build tension and reach resolution, for example using leading tones. Unlike the fifth, however, it is a dissonant interval, and appears in chords mostly as a major seventh. These two intervals (fifth, semitone) represent different aspects of "closeness" between pitch classes, where the fifth is the closest harmonically, and the semitone is closest frequency wise. Based on this observation, we define a 2-dimensional tonic coordinate system called the "Tonic Cross", where one axis increments in perfect fifths and the other in semitones.

Definition 9 (Tonic Cross). *The tonic cross is a two-dimensional tonic coordinate system in which one dimension represents the circle of fifths and the other represents the chromatic scale. The point (a,b) represents the conjunction of the sets of pitch classes which are represented by (0,b) and (a,0),* $a, b \neq 0$.

An example of a tonic cross, with a tonic note of C is shown in figure 5.3. On this coordinate system, we can make some interesting observations. Firstly, the tritone interval is the furthest from the tonic note ($F\sharp$) on both axes. This is an interesting interval in that it is unstable [91] and has a unique role in music. A second observation is that cadences, ie transitions that resolve, have the same shape. These are $G \to C$ ($V \to I$), $B \to C$ ($vii^o \to I$), $F \to C$ ($IV \to I$), and $C\sharp \to C$ (tritone substitution of $V \to I$). A third observation is that minor third and major third intervals also have the same shape (square). The two larger squares are a major third above (E) and a major third below ($G\sharp$), while the two smaller squares are a minor third above ($D\sharp/E\flat$), and a minor third below (A). Finally, there are two more intervals that are represented as squares, the major second/ninth (D), which is a common extension to major chords, and the minor seventh ($B\flat$) which is a common extension to minor chords.

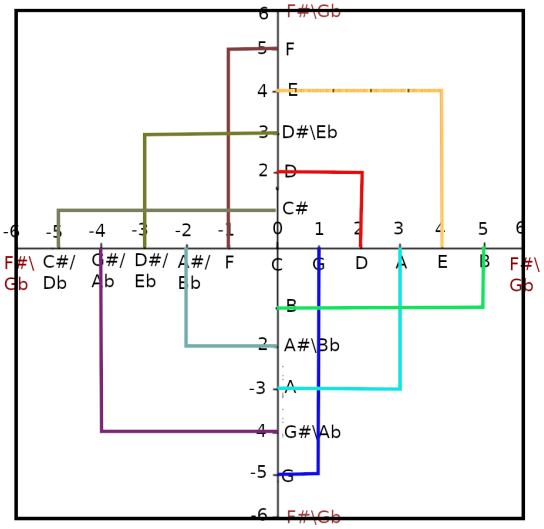


Figure 5.3: The tonic cross coordinate system with a tonic note of C.

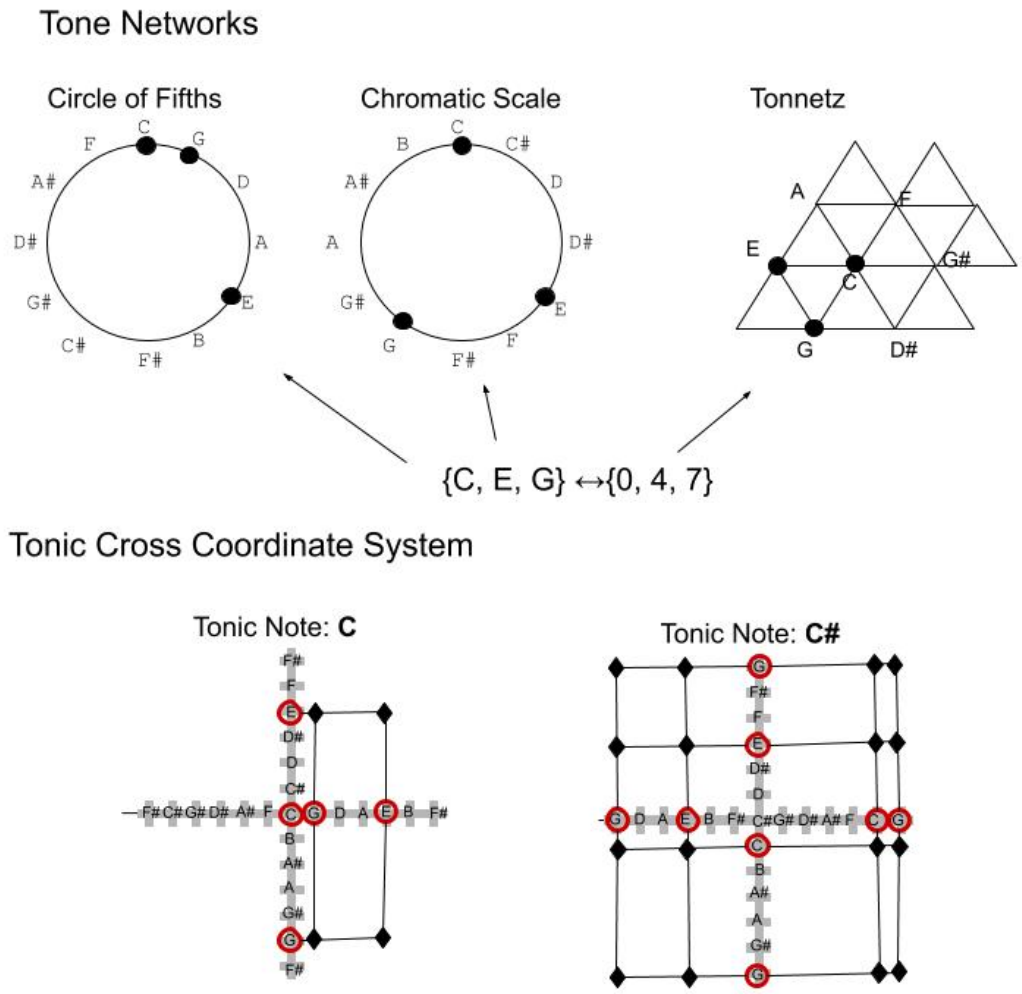


Figure 5.4: Tone networks and tonic coordniate systems

On the tonic cross, we can represent sets of pitch classes, as sets of points. We call these the harmonic points of the set of pitch classes. For example in figure 5.4 we show the harmonic points of a C major triad on two tonic crosses, one with a tonic note of C and one with a tonic note of $C\sharp$.

Definition 10 (Harmonic Points). *The harmonic points of a set of pitch classes x, given a tonic coordinate system C with tonic note t is the set of all points in the coordinate system which represent any set of pitch classes in the power set: $\mathcal{P}(x)$, in addition to the origin. The harmonic points will be symbolized $PP_t(x)$*

Our hypothesis is that geometrical properties of such sets of points can give us information about the sound of a set of pitch classes, in the context of a tonic note, and the corresponding tonic cross. The two properties we use in this work are the span of the harmonic points, defined as the Euclidean distance between the two furthest points, and the center offset of harmonic points, defined as the Euclidean distance from the center of the points to the origin. These two properties are visualized in figure 5.5.

Definition 11 (Span of set of pitch classes). *In the context of a tonic coordinate system, the span of a noteset x is a tonic property defined as the maximum distance between any two harmonic points:*

$$Sp_t(x) = max_{i,j}(||PP_t(x)_i - PP_t(x)_j||) \tag{5.1}$$

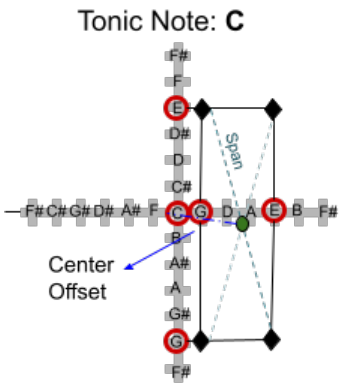


Figure 5.5: Visualization of Span, and Center Offset of a C major triad, and given a tonic note of C

Definition 12 (Center offset of set of pitch classes). *In the context of a tonic coordinate system, the center offset of a noteset x is the distance of the geometric center of all harmonic points to the origin*

$$Co_t(x) = ||\frac{1}{|PP_t(x)|}\sum_{p \in PP_t(x)} p|| \tag{5.2}$$

For example, sets of notes whose harmonic points are symmetric around the origin will have a center offset of 0. Given a tonic note of C, such sets could be $\{C, C\sharp, B\}$, $\{C, D, B\flat\}$, $\{C, E\flat, A\}$, and $\{C, F, G\}$. These sets of pitch classes all have dissonant characteristics, and can be used for building tension. The first would almost never be used in a composition, the second could be part of a $C9$ chord where the dissonance is caused by the tritone between $B\flat$ and the implied E, the third is a diminished chord, and the fourth is a sus4 chord. Contrarily, sets of pitch classes with a large center offset tend to have their harmonic points lie in the same quadrant, or on the same side of an axis. For example, given a tonic note of C, the cardinality 3 sets that include C with the highest value of center offset (≈ 3.16) are: $\{C, E, B\}$,$\{C, D\flat, A\flat\}$,$\{C, E, F\}$,$\{C, G, A\flat\}$. Each of these sets of notes has parts of a maj7 chord (Cmaj7, Dmaj7, Fmaj7, Gmaj7). Such chords have some dissonance, given the maj7 interval, however they are not usually used to build tension, instead the dissonance adds to the melancholic sound of these sets of pitch classes. Other chords, such as major and minor triads have values in between these two dissonant extremes (C major triad center offset ≈ 1.7, F major triad ≈ 0.94, C minor triad ≈ 0.94, with a tonic note of C). Regarding Span, the highest values are for the sets of pitch classes which contain the tritone above the tonic ($F\sharp$ with a tonic note of C), while the lowest values are for sets with pitch classes whose points are clustered together, such as $\{C, D, E\}$. Low span sets seem to contain notes that would be part of melodic lines, however we have not shown this in any way.

These two properties, seem to convey some musical information regarding the building of tension, and the amount of dissonance expected from sets of pitch classes. We have not clearly stated their musical interpretation, as it is out of the scope of this dissertation, but we hypothesize that the information encoded in these properties can be used to determine characteristics of music, which in turn could be used for the purpose of evaluation. This is an example of how knowledge encoded in graphs (such as the circle of fifths) can be used for evaluating AI systems.

5.2.2 Evaluation Metrics

The two metrics we have defined, and any other geometric properties of harmonic points that could be defined, are computed for a set of pitch classes x given a tonic note t (and we call them *tonic properties*).

Definition 13 (Tonic Property). *A tonic property Pr_t is a function of a set of pitch classes, dependant*

on a tonic note t:

$$Pr_t : \mathcal{P}(\mathbb{PC}) \times \mathbb{PC} \to \mathbb{R} \tag{5.3}$$

Where $\mathbb{PC}$ is the set of pitch classes and $\mathcal{P}$ denotes the power set.

We would however want a metric that is computed for a *sequence of sets of pitch classes*, and that is independent of a tonic note. We can derive such properties from tonic properties.

Definition 14 (Non-tonic Property). *A non-tonic property Pr is a function of a noteset x which does not depend on a tonic note:*

$$Pr : \mathcal{P}(\mathbb{PC}) \to \mathbb{R} \tag{5.4}$$

Where $\mathbb{PC}$ is the set of pitch classes and $\mathcal{P}$ denotes the power set.

To derive non-tonic properties from tonic properties, we use pooling functions. A pooling function is any function $\mathbf{F} : \mathcal{P}(\mathbb{R}) \to \mathbb{R}$. Pooling functions will be symbolized with **bold** capital letters. Some examples follow:

- The mean of a set, symbolized **E**
- The maximum of a set symbolized **M**
- The span of a set (max-min), symbolized **S**

We experimented with two approaches for this. The first, which we call *relevant pooling properties*, consist of computing a tonic property for a noteset x, $|x|$ times, where each time a different element of x serves as a tonic note. Then, the $|x|$ different values are pooled (either by averaging, or taking the max, or the span), leading to a single value that is independent of a tonic note. Similarly, the second approach for computing non-tonic properties, which we call *global pooling properties*, involves computing tonic properties 12 times, where each time a different pitch class serves as the tonic note.

Definition 15 (Relevant Pooling Property). *Given a pooling function **F** and a tonic-property Pr_t, the relevant pooling property $\mathbf{F}Pr$ is defined as the non-tonic:*

$$\mathbf{F}Pr(x) = \mathbf{F}(\{Pr_t(x)\}), t \in x \tag{5.5}$$

Definition 16 (Global Pooling Property). *Given a pooling function **F** and a tonic-property Pr_t, the global pooling property $\mathbf{F}Pr^*$ is defined as the non-tonic:*

$$\mathbf{F}Pr^*(x) = \mathbf{F}(\{Pr_t(x)\}), t \in \mathbb{PC} \tag{5.6}$$

An example of a non-tonic property is the relevant span of the spans of a noteset $\mathbf{S}Sp$:

$$\mathbf{S}Sp(x) = \mathsf{max}_{t,u \in x}(Sp_t(x) - Sp_u(x)) \tag{5.7}$$

and the corresponding global span of the spans of a noteset $\mathbf{S}Sp^*$:

$$\mathbf{S}Sp^*(x) = \mathsf{max}_{t,u \in \mathbb{PC}}(Sp_t(x) - Sp_u(x)) \tag{5.8}$$

Properties of sequences of sets of pitch classes

By using pooling functions we may also accumulate properties across entire sequences of sets of pitch classes (symbolized with capital letters), for instance the mean span of the span $\mathbf{ES}Sp$

$$\mathbf{ES}Sp(X) = \frac{1}{|X|} \sum_{x \in X} \mathbf{S}Sp(x) \tag{5.9}$$

or the mean cardinality $\mathbf{E}||(X)$

$$\mathbf{E}||(X) = \frac{1}{|X|}\sum_{x\in X}|x| \tag{5.10}$$

and the mean relevant span of center offsets:

$$\mathbf{ES}Co(X) = \frac{1}{|X|}\sum_{x\in X}\mathsf{max}_{t,u\in x}(Co_t(x) - Co_u(x)) \tag{5.11}$$

In addition, we define properties of sequences of sets of pitch classes based on the rate of change of properties of their constituent sets of pitch classes. An example of such a property is the variance in the rate of change of the cardinality of sets of pitch classes $\sigma^2\Delta||(X)$

$$\sigma^2\Delta||(X) = \sigma^2(\{|X_{i+1}| - |X_i|\}) \tag{5.12}$$

$$0 < i \leq |X| - 1$$

or the maximum cardinality change of consecutive sets of pitch classes:

$$\mathbf{M}\Delta||(X) = \mathsf{max}_{0<i\leq|X|-1}(|X_{i+1}| - |X_i|) \tag{5.13}$$

Important aspects of music exist in multiple time-scales. For this reason we will utilize lower resolution versions of the initial sequences of sets of pitch classes.

Definition 17 (Half resolution sequence). *Given a sequence of sets of pitch classes $X = x_1, x_2 \dots x_n$ the half resolution sequence, symbolized $X_{/2}$ is defined as the sequence:*

$$X_{/2} = x'_1, x'_2 \dots x'_{n'}$$

where

$$x'_i = x_{2i-1} \cup x_{2i}, 0 < i < \frac{n}{2}$$

In general we symbolize

$$X_{/2^i} = \underbrace{X_{/2_{/2_{\dots}}}}_{i}$$

and

$$X_{/1} = X$$

In order to quantify any musical information conveyed by lower resolution versions of the original sequence we define properties of sets of sequences $\{X_{/2^i}\}$. This can be done by accumulating the value for a property across different resolutions by use of pooling functions, similarly to how properties of sequences where defined by pooling properties of sets of pitch classes.

Definition 18 (Cumulative Resolution Property). *Given a property Pr of a sequence of sets of pitch classes X, and a pooling function $\mathbf{F}$ the cumulative resolution property $\alpha\mathbf{F}Pr$ is defined as:*

$$\alpha\mathbf{F}Pr(X) = \mathbf{F}(\{Pr(X_{/2^i})\}) \tag{5.14}$$

for some set of integers $\{i\}$, $i < log_2|X|$

In addition we may define properties by quantifying the change in a property of a sequence when resolution is lowered, similarly to how properties of sequences of sets of pitch classes where defined based on the rate of change of properties of sets of pitch classes. To this end we define resolution ratios. Note that for all ratio definitions below the value is set to zero if the denominator of the ratio is zero.

Definition 19 (Resolution Ratio of a Property). *Given a property Pr of a sequence of sets of pitch classes X and a pooling function **F** the resolution ratio property $r\mathbf{F}Pr$ is defined as:*

$$r\mathbf{F}Pr(X) = \mathbf{F}(\{\frac{Pr(X_{/2^{i+1}})}{Pr(X_{/2^i})}\}), i < log_2|X| \tag{5.15}$$

An example of such a property is the mean resolution ratio of the mean cardinality of a sequence of sets of pitch classes.

$$r\mathbf{EE}||(X) = \frac{1}{\log_2(|X|)} \sum_{i=0}^{\log_2(|X|)-1} \frac{\frac{1}{|X_{/2^{i+1}}|}\sum_{x\in X_{/}2^{i+1}}|x|}{\frac{1}{|X_{/2^i}|}\sum_{x\in X_{/2^i}}|x|} \tag{5.16}$$

Heuristics for evaluation

Ideally there would exist a property which reflects "musicality", where sequences of notesets which represent real music would have a higher value than sequences of notesets which represent AI generated music, or non-musical sequences. Our goal is to define properties which are heuristics for "musicality".

Numerous such properties may be defined in this framework, some of which will be useful for assessing musical qualities of a sequence of notesets and thus aid in the process of evaluating AI generated music. As an example, we constructed four heuristic properties which are based on intuition and empirical observations. These are not guaranteed to be optimal, or even effective, and their usefulness is demonstrated experimentally in the next section.

We expect the mean cardinality of notesets to increase for lower resolution versions of a sequence which represents real music. This is an indicator of variety in notes being played, and is quantified in heuristic measure H_1.

$$H_1(X) = r\mathbf{EE}||(X) \tag{5.17}$$

For the mean relevant span of center offsets, we expect the value to not decrease when resolution is lowered. This hypothesis is based on observations of the distribution of values of $\mathbf{S}Co$ of all 4096 notesets with respect to their cardinality (figure 5.6). Importantly, there is a local maximum for cardinality 7 which represents diatonic scales among other notesets. In addition, more musical notesets, such as named chords, tend to have higher values than other notesets of the same cardinality.

$$H_2(X) = \min(r\mathbf{EES}Co(X), 1) \tag{5.18}$$

With regards to the mean resolution ratio of the maximum difference in cardinality of consecutive notesets of a sequence $r\mathbf{EM}\Delta||(X)$ we expect the value to be closer to 1 for real music. Such a value would indicate that there are no "erratic" changes in the sequence of notesets, or that even for lower resolutions there exists some change in cardinality, otherwise $\mathbf{M}\Delta||(X_{/2^N})$, would be assigned a zero value.

$$H_3(X) = r\mathbf{EM}\Delta||(X) \tag{5.19}$$

Finally, the fourth heuristic is defined as the product of the three properties defined above

$$H_4(X) = H_1(X) * H_2(X) * H_3(X) \tag{5.20}$$

5.2.3 Experiments

In order to assess the usefulness of the heuristics for evaluating music we conducted the following experiment.

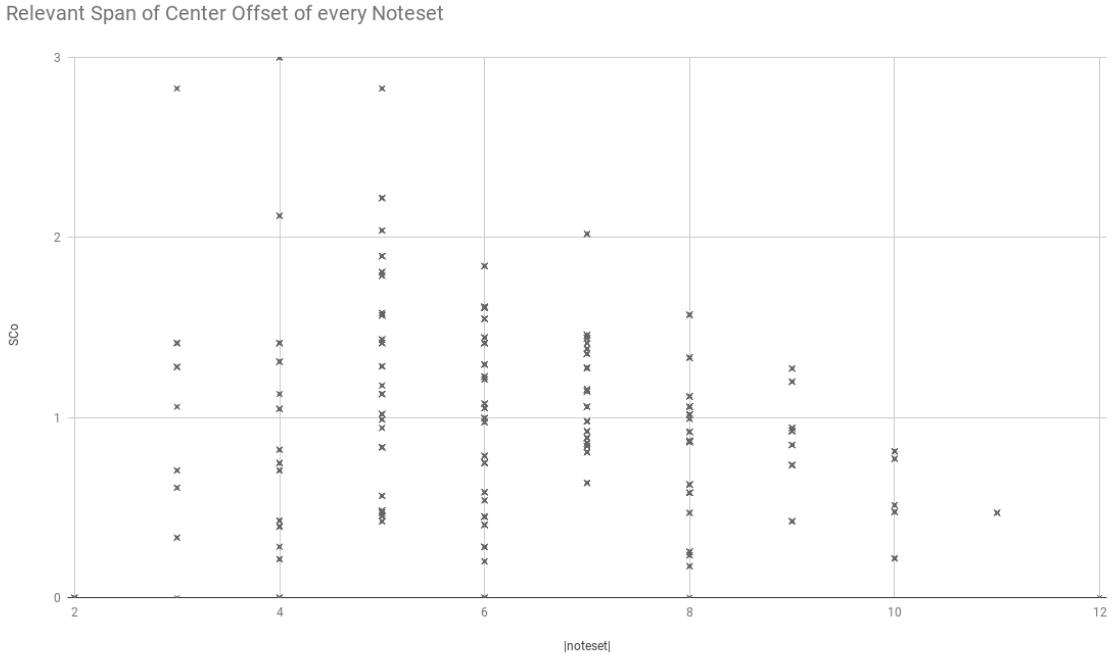


Figure 5.6: Relevant span of center offset of every noteset. x-axis represents noteset cardinality and y-axis represets the property SCo

Setup

We implemented five LSTM-based neural networks for autoregressive generation of symbolic music and developed a platform for crowd-sourcing evaluation[3]. We acquired a pretrained MuseGAN [48] along with two large datasets: Reddit MIDI [162] and Lakh Pianoroll Dataset [159][48]. We trained our networks on a small subset of classical guitar music, which will be referred to as the *train set* in tables and figures. The generated results where evaluated threefold: 1) user survey, 2) objective metrics from [48] and 3) heuristics from our framework.

In addition we created the *Jazz*, *Metal* and *Bach* datasets each containing 300 random MIDI files from the corresponding folders in the Reddit MIDI dataset, and used the pretrained MuseGAN to unconditionally generate 300 MIDI files with each of the two different inference modes. We then calculated the heuristics on these datasets for further comparison.

Data Representation

For training the neural networks, MIDI files were represented as sequences of tokens, where a different token is assigned to each combination of note and quantized duration which occurs in the dataset.

For calculating the various metrics, MIDI files were converted to pianoroll representations at a resolution of 12 slices per quarter note. The duration of these pianorolls was set to 512 slices (approximately 10 bars at $\frac{4}{4}$ time signature) and we only used files with greater or equal duration. All non-percussive tracks of a MIDI file were summed into a cumulative pianoroll. For the heuristic metrics of the proposed framework each slice of the pianoroll is converted to a noteset by observing the occurence of pitch classes in the slice.

[3] https://www.melodybot.com/vote

Neural Networks

For generating symbolic music we followed the autoregressive approach, such as in [27]. Sequential autoregressive models express predictions of observations based on past predictions. Specifically, given a sequence $X = [x_1, x_2, ..., x_N]$, an autoregressive model factorizes the likelihood into a forward product $p(x) = \prod_{t=1}^{N} p(x - t|x_{<t})$ or a backward one $p(x) = \prod_{t=N}^{1} p(x - t|x_{<t})$.

We implemented three LSTM-based neural network architectures: a Feed Forward Lstm, an Autoencoder and an Autoencoder with Self-attention. Based on them we modify their memory size and we finally built five different neural networks, shown in figure 5.7. These will be referred to as:

- LSTM256: Three stacked LSTM layers with 256 units each, following an embedding layer
- LSTM512: Three stacked LSTM layers with 512 units each, following an embedding layer
- AE256: An autoencoder configuration, where the encoder consists of one LSTM layer of 256 units, as does the decoder
- AE512: Similar autoencoder configuration but LSTM layers consist of 512 units each
- AEATT: Autoencoder with self-attention

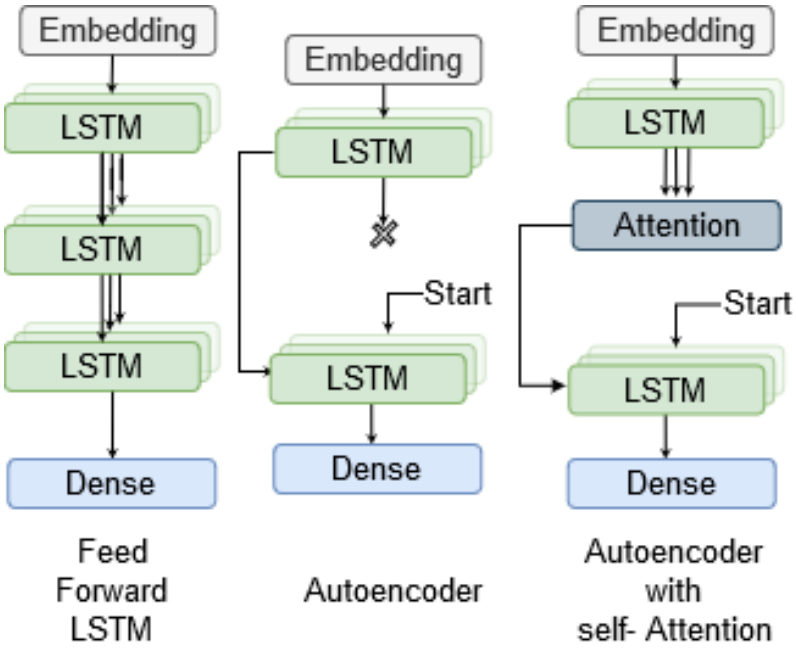


Figure 5.7: The three architectures we used for generation of music.

We trained all networks for 200 epochs with an early stopping criterion for categorical cross entropy on a validation set. We used Adam [105] as the optimization algorithm and the methodology described in [185] for setting the learning rate. We also included a dropout parameter of 0.2 for LSTM layers and used teacher forcing while training autoencoder architectures.

User Survey

We evaluated each neural network configuration by conducting a user survey online. In total 1,152 users participated, of which 569 stated they had musical knowledge, over a span of 100 days. Each user was presented with 20 different pieces of music, 10 of which were composed by a neural network and 10 were from the train set. For each piece of music, every user provided three ratings (from 1 to 5) pertaining to: 1) How much they like the piece, 2) How interesting they find the piece and 3) Whether

the composer of the piece was a human or a computer. In order to increase user engagement, after each question the user was notified if they answered the third question (composer) correctly or not. This had the effect of "training" some users to distinguish between human and computer composers, as is apparent from the distribution of mistakes which are more numerous for the first questions.

Objective Metrics

We calculated objective metrics proposed in [48] on the generated samples and on the train set. Closer values of these metrics between generated and real samples indicate a better model. Of the proposed objective metrics we used polyphonicity (PP) and used-pitch-classes per bar (UPC). For UPC since not all pieces are in $\frac{4}{4}$ time signature we arbitrarily set a bar to be 32 samples of a pianoroll. In addition we calculated the total number of pitches used in a piece (PU) and the total number of pitch classes used (PCU). Results are shown in table 5.2.

Results

The results of our experiments are summarized in the tables below. The baseline for evaluation is the user survey, specifically how much users liked the music that they listened to and how interesting they found it. Results are shown in table 5.1.

Table 5.1: Average "liked" (L) and "interesting" (I) votes for musicians (M) and non-musicians (NM)

Model	L (NM)	L (M)	I (NM)	I (M)
LSTM256	1.47	1.60	1.29	1.57
LSTM512	1.75	1.94	1.93	2.09
AE256	3.21	2.93	3.10	3.12
AE512	3.03	**3.22**	3.25	3.27
AEATT	**3.41**	3.18	**3.52**	**3.86**
Train set	3.21	3.57	3.71	3.63

Concerning the objective metrics from [48] results are shown in table 5.2. In general these agree with the user survey, validating their usefulness for evaluation.

Table 5.2: Objective metrics proposed in [48]

Model	PP	PCU	PU	UPC/32
LSTM256	0.05	5.05	7.65	3.18
LSTM512	0.08	6.07	10.01	3.49
AE256	0.24	7.87	14.09	4.07
AE512	**0.46**	9.10	18.31	4.89
AEATT	0.79	**9.41**	**19.14**	**6.32**
Train set	0.41	9.71	22.53	6.17

The results for our heuristics are shown in table 5.3. Our goal when defining these was for a larger value to indicate a more musical sequence of notesets without the need for comparison with the train set. This can be argued to have been achieved to an extent for H_2, H_3 and H_4. A histogram for **non-zero** observed values of H_4 is shown in figure 5.8.

Finally, we measure how well each evaluation method separates the train set from generated music via F1 score in table 5.4. For users we use the data from the survey. For the heuristics we find

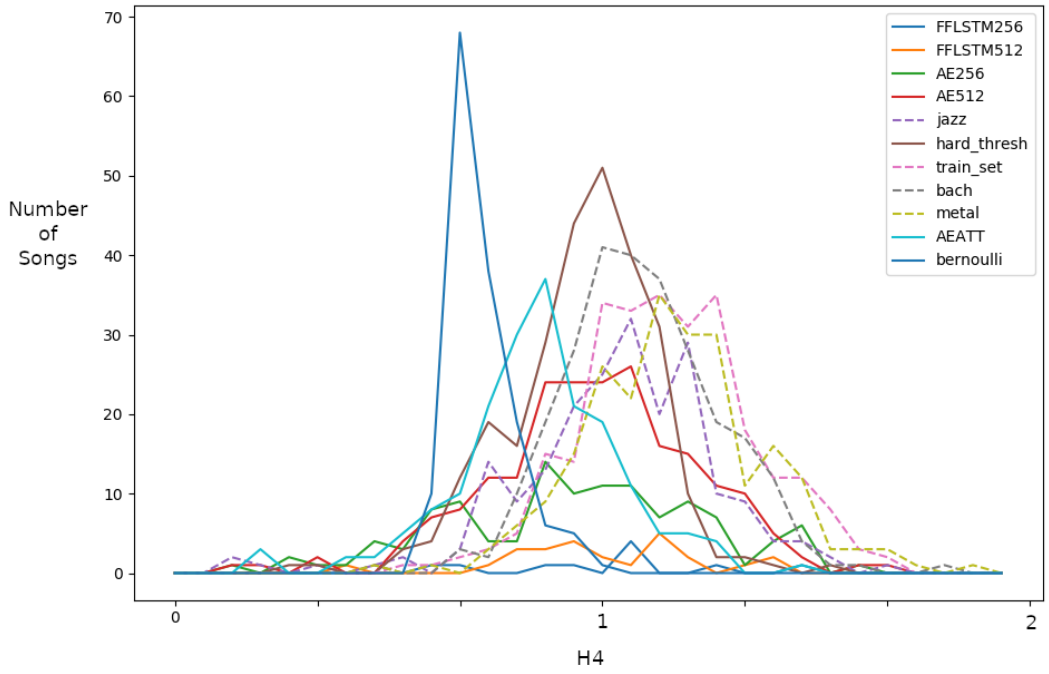


Figure 5.8: Histogram of non-zero observed values for H_4 (best viewed in colour) - many of the "poor" samples had H4=0

Table 5.3: Heuristic metrics calculated on generated and real samples. MG refers to MuseGAN, HT and BS refer to MuseGAN inference modes (hard thresholding and Bernoulli sampling). Reported values are mean (standard deviation)

Model	H_1	H_2	H_3	H_4
LSTM256	1.18 (0.1)	0.56 (0.42)	0.03 (0.15)	0.03 (0.18)
LSTM512	1.19 (0.1)	0.68 (0.39)	0.07 (0.24)	0.08 (0.28)
AE256	**1.20 (0.09)**	0.85 (0.27)	0.33 (0.43)	0.38 (0.51)
AE512	1.18 (0.08)	**0.94 (0.15)**	**0.61 (0.45)**	**0.70 (0.51)**
AEATT	1.09 (0.06)	**0.94 (0.07)**	0.52 (0.43)	0.54 (0.45)
Train set	1.21 (0.04)	0.99 (0.05)	0.87 (0.35)	1.04 (0.42)
Bach	1.77 (0.05)	0.98 (0.05)	0.86 (0.34)	0.99 (0.40)
Metal	1.18 (0.05)	0.97 (0.11)	0.78 (0.45)	0.91 (0.54)
Jazz	1.25 (0.1)	0.94 (0.15)	0.62 (0.45)	0.72 (0.54)
MG (HT)	1.18 (0.02)	0.96 (0.04)	0.77 (0.30)	0.77 (0.30)
MG (BS)	1.20 (0.01)	0.73 (0.05)	0.41 (0.42)	0.37 (0.37)

the threshold which maximizes F1 score when samples above the threshold are classified as real and below as computer generated.

Discussion

Even though the three different evaluation approaches agree with each other in general, there are some interesting discrepancies, in particular when comparing the two best performing neural net-

Table 5.4: Turing test classification F1 scores

Model	Users	H_1	H_2	H_3	H_4
LSTM256	0.93	0.76	0.83	0.92	0.92
LSTM512	0.83	0.73	0.78	0.90	0.90
AE256	0.70	0.73	0.72	0.79	0.80
AE512	0.70	0.75	0.71	0.70	0.72
AEATT	0.65	0.92	0.78	0.72	0.79

works: AE512 and AEATT. In our survey, listeners who stated they have some musical knowledge liked music generated by AE512 more than that generated by AEATT, although they found music generated by AEATT more interesting. By observing table 5.2, it is apparent that the total number of pitches used (PU) and total number of pitch classes used (PCU) are similar for both neural networks and the train-set. However AEATT, on average, uses significantly more pitch classes per 32 steps (UPC/32) than AE512, and is significantly more polyphonic (PP) than both AE512 and the train-set. This indicates that AEATT uses a variety of pitch classes and pitches similar to the train-set, but they sound polyphonically in the generated music more often than in real music. Intuitively this would mean a more obfuscated melody, which could lead to musicians liking it less, in addition to more complex chords and chord progressions which could be the reason for which musicians found the generated samples interesting.

Regarding H_1, AEATT had the smallest value on average. This metric measures the mean increase of noteset cardinality when resolution is halved. The fact that its value is closer to 1 indicates that on average cardinality does not increase as much when accumulating notesets across more time-steps, which would be consistent with overly polyphonic music. However this cannot be deduced directly from H_1, since it does not take into consideration the absolute number of pitch classes used at each resolution. For instance a piece of music which uses very few pitch classes over a long duration would also have a value of H_1 close to 1 even if there were no polyphonicity. In addition, the highest scores for H_1 were observed on the Bach, Jazz and Train datasets, which agrees with our intuition that harmonically rich music should tend to have a higher value of H_1. A notable exception is the Metal dataset, for which the value is similar to that of the neural networks.

For the proposed heuristic H_2, both neural networks (AE512 and AEATT) on average score the same, with AE512 having slightly more variance. When defining the heuristic, we hypothesized that when resolution is decreased, the value of $\mathbf{ES}Co$ should not decrease as much for real music when compared to AI generated music. This hypothesis was based on the fact that named notesets from music theory, such as diatonic scales, tend to have a larger value of $\mathbf{S}Co$ when compared to notesets of the same cardinality, in addition to the existence of a local maximum at cardinality 7, which represents diatonic scales. Thus, in our interpretation, a value of H_2 closer to 1 could indicate either a) that noteset cardinality does not change much when resolution is decreased, such as for the case AEATT or b) the music is "more diatonic" (tends to use notes which comprise a diatonic scale) than if it had a lower value of H_2. This is confirmed by observing the values of H_2 for the datasets of real music, in particular that the lowest value of H_2 is for Jazz music, which tends to be "less diatonic", when compared to Metal, or Classical music.

The heuristic H_3 had a lot of variance for both generated and real samples when compared to the other metrics. Low values of H_3, such as for LSTM256 and LSTM512 would indicate that either a) there are no changes in cardinality for many of the $log_2|X|$ resolutions, for which case the ratio at these resolutions is assigned a zero value, or b) there exists an erratic change in noteset cardinality in the original sequence. For the case of the worst performing models the value is close to zero due to a), since from table 5.2 it is apparent that these models utilize very few pitch classes (PCU) and are almost exclusively monophonic (PP) - which means most time-steps have notesets of cardinality

1.

Finally, it is interesting to compare the best performing MuseGAN (MG-HT) with the autoregressive models which we trained and the datasets which we collected. In particular with regards to H_2, H_3 and H_4 it seems to slightly outperform our models, while for H_1 the performance is similar. However for three of the metrics it also scores higher than the Jazz dataset, which highlights some weaknesses of the proposed heuristics, and challenges which we will face in future work.

5.3 Genre Recognition from Symbolic Music

Technologies for automatic music classification and tagging tasks have advanced considerably in recent years, spurred on by their application in industry. Such tasks, for instance genre recognition [147, 216, 133, 215], or mood classification [163, 213, 150] are typically approached in the audio domain. As mentioned in section 5.1, raw audio data in WAV format is difficult to handle, and specialized neural architectures tend to be resource intensive [146], and thus most works for music classification and tagging utilize digital signal processing (DSP) for extracting features from the audio, such as spectrograms, and then applying machine learning methodologies to learn from the extracted features [156]. In our work, we wanted to explore, the much less active research area of classifying and tagging *symbolic representations* of music.

Symbolic representations of music can be very useful for AI research, as they are abundantly available for free on the web, are light-weight, and they encode the bare-bones musical information of a piece of music (what notes are being played, at what time, and with what dynamics). Recently, approaches that incorporate symbolic representations when analyzing audio of music have shown promising results [207], while tasks for converting audio to symbolic (transcription)[17, 203] and vice versa (synthesis)[53, 208] are also gaining traction. Nonetheless, there have not yet been developed specialized neural architectures for handling symbolic representations of music, and most works adapt networks from natural language processing [95] and from computer vision [48]. In our work, we experimented with one-dimensional convolutional neural networks [190], and explored how hyper-parameters such as network width, and depth affect the performance of a genre recognition model, by keeping number of parameters and receptive field constant.

The state-of-the-art approach for MIDI genre classification presented by Ferraro and Lemström in [56] uses an algorithm for recognizing patterns of notes in an input sequence and then performs classification based on recognized patterns. These patterns are local, with the best results achieved from extracting four or five note patterns. This, along with the success of 1D CNNs for other tasks in the symbolic music domain [48] and their suitability for sequence pattern recognition, as has been shown in multiple domains, such as pattern recognition in DNA sequences by Lanchantin et al in [111], motivates us to explore 1D CNNs for the task of recognizing genres in symbolic music. For details on 1D CNNs, that are used in this section, we refer to section 2.4 of the background chapter of this dissertation.

5.3.1 Related Work

Most works related to genre classification involve music in audio (raw) format. The most up-to-date techniques for the classification of music use neural networks on MFCCs or spectrograms [198, 147], some of them focus on feature selection [180, 171], or introducing new architectures [133, 215, 127, 216].

The abundant availability of MIDI files online, from multiple sources, has given rise to the challenge of automatically organizing such large collections of MIDI files. One criterion for organization is music genre, among others such as music style, similarity, and emotion. In [132] McKay and Fujinaga argue in favor of genre classification, despite inherent difficulties such as ground truth reliability.

There are many different approaches in the literature for genre recognition in the symbolic music

domain. Dannenberg et al [34] used a machine learning approach, including a simple neural network, on a custom dataset for successful genre recognition in the symbolic domain. In [101], Karydis et al combine pattern recognition with statistical approaches to successfully achieve genre recognition for five subgenres of classical music. Kotsifakos et al in [107] compute a sequence similarity between all pairs of channels of two MIDI files and then use a k-NN classifier for genre recognition, on a dataset of 100 songs and four genres. Zheng et al. [222] extract features related to the melody and the bass through a musicological perspective, incorporating text classification techniques and using Multinomial Naive Bayes as the principal probabilistic classifier, in a self-collected dataset of 273 records.

These approaches were experimentally validated on relatively small datasets compared to, for example, the openly available Lakh MIDI dataset [159]. For large scale datasets, Ferraro and Lemström [56] utilize pattern recognition algorithms SIA [136] and P-2 [195] in addition to a logistic regression classifier to solve the task. The benefit of this approach is interpretability since the authors have created a large corpus of genre-specific patterns, which could also be utilized for other music related tasks. Duggirala and Moh [52] apply Hierarchical Attention Networks in music genre classification, after converting the audio files into a word embedding representation. Liang et al. [120] propose four word embedding models consisting of three vocabularies (chroma, velocity, and note state) and apply these models in three MIR tasks: melody completion, accompaniment suggestion, and genre classification, concluding the robustness and effectiveness of their embeddings.

5.3.2 Multiple Sequence Resolution Networks

Symbolic music is typically represented as sequences for deep learning approaches, but there exist multiple ways to represent music in a tree structure in order to capture information across multiple time scales. For instance, hierarchical models for music analysis proposed by Schenker [175] converted to trees by Marsden in [130] and of Lerdahl and Jackendoff [118] which partially formalise Schenker's ideas, parsing musical structures into strict trees. Rizo et al. propose a non-linear representation of a melody based on trees and they study the influence of different tree representations on classification rates in three corpora with monophonic melodies, concluding that tree coding gives better results [167]. Rizo and Marsden extend the Music Encoding Initiative (MEI) [169] representation with "semantic" and "non-semantic" encodings which allows the tight association of the analytical information and the information in the score [168]. This motivates us to explore ways in which such trees may be given as input to a neural network instead of sequences. In this work, we present a fully convolutional approach called MuSeReNet (**Mu**ltiple **Se**quence **Re**solution **Net**work), in which each level of the input tree represents the original sequence at a different resolution.

In this work, we set a baseline for tree representation utilization for information retrieval from symbolic music, by using full binary trees as input structures and by then treating each level of the tree as a separate input. Each node of the binary tree has as a value the average of its children, thus each level of the tree is equivalent to the original sequence - which is represented by the leaves of the tree - at a lower resolution. This means that MuSeReNets presented in this paper are similar to multiple resolution CNNs which have been successful for some computer vision tasks such as skin lesion recognition by Kawahara and Hamarneh in [102].

Intuitively, the first levels of a CNN detect low-level local features in the data, and more complex higher-level features are captured in deeper layers. However, there could exist high-level features which may be simply extracted from a lower-resolution representation of the input without requiring increasing the depth of the network. For instance, in the case of symbolic music, a simple feature that could be extracted from higher levels of a tree (closer to the root) would be the key signature of a large segment of the input - which relates to the set of notes which appear in the segment. Such features may then be combined with lower-level features extracted from levels closer to the leaves of the tree and fed to deeper layers of the network for further feature extraction and eventually solving a task, which in our case is genre classification.

The first module of a MuSeReNet is a set of average pooling operations which act on the original

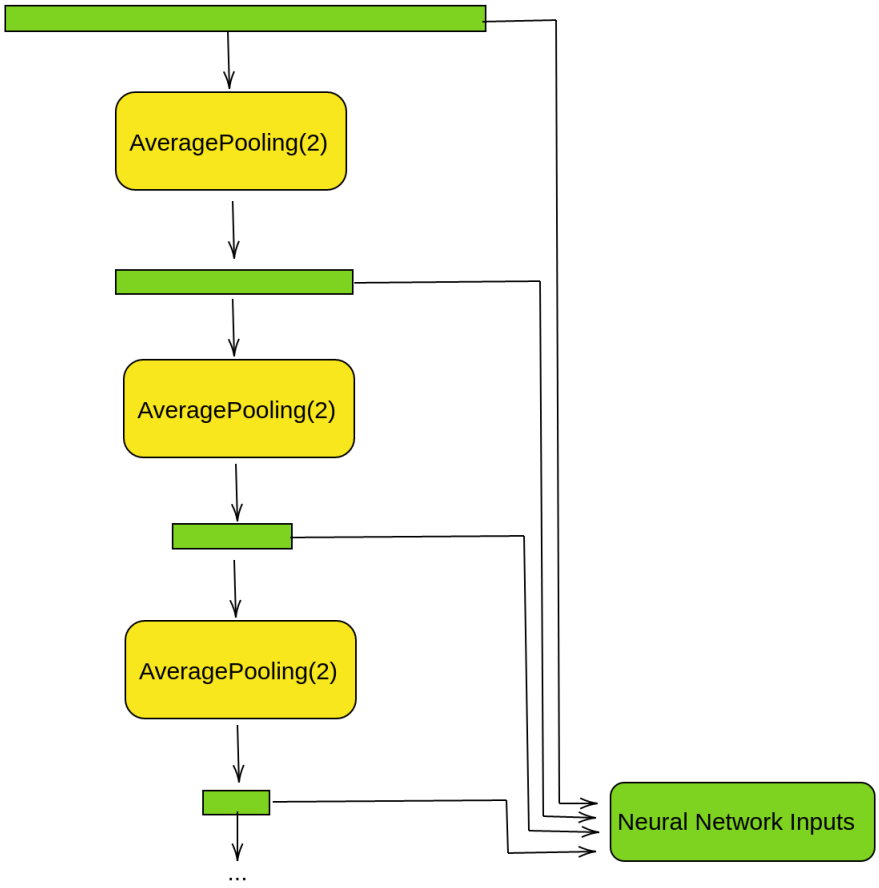


Figure 5.9: Constructing a binary tree where levels are equivalent to the input sequence at lower resolutions

sequence, each producing a version of the original sequence at a different resolution (Figure 5.9). Each of these is treated as a separate input for the neural network.

There are many different ways to make use of these inputs. For MuSeReNets we distinguish between two cases: When information flows from the leaves to the root (Figure 5.10) and when information flows from the root to the leaves (Figure 5.11).

In the first case, in which information flows from the leaves to the root (Figure 5.10), each input is fed through a block which consists of convolutional layers followed by a max pooling operation with the same stride and kernel size as the average pooling operation which generated the specific input from its higher resolution counterpart. This way, and by using 'same' padding for convolutional operations, the output of a specific block has the same sequence length as the original input at the previous resolution level and may be concatenated along their second axis, producing a sequence of the same length and with more channels. The result of concatenation is the original sequence at a lower resolution augmented with features extracted from the convolutional block which processed the input at a higher resolution. This process is repeated until we reach the root of the tree, where the sequence length is 1, and the vector consisting of the root and features extracted from the previous convolutional block is fed to a fully connected layer with the goal of solving a specific task.

In the second case, in which information flows from the root to the leaves (Figure 5.11), max pooling operations are replaced with upsampling operations, and the order with which inputs are fed to the network is reversed (the root first instead of the leaves first). In this case, the result is a sequence of length equal to the original sequence, but is augmented with features that were extracted

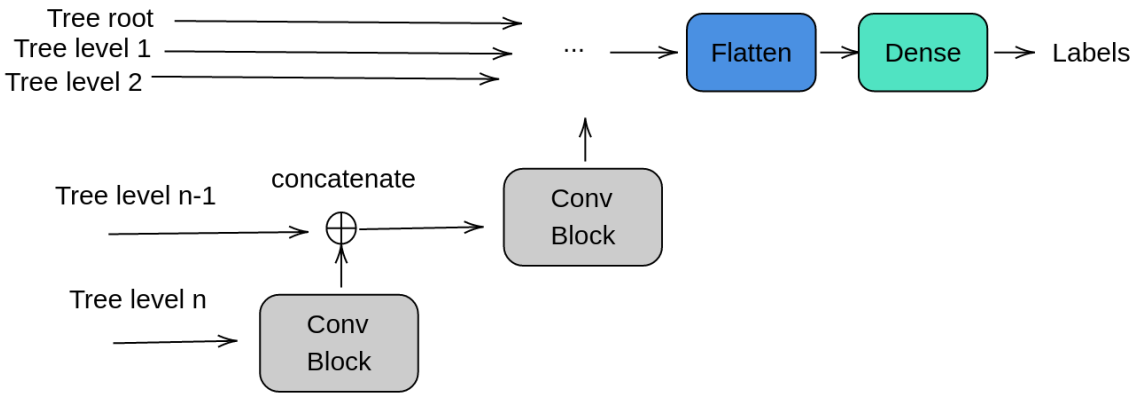


Figure 5.10: A MuSeReNet where the information flows from the leaves to the root.

by convolutions on lower resolution versions of the sequence. Intuitively, via the upsampling and concatenation operations, this network could learn features that are local but are affected by the context provided by the lower resolution version at a previous layer. Such an architecture is similar to U-nets [170] which is used for image segmentation. This resulting augmented sequence may then be fed to further neural network layers in order to solve a specific task.

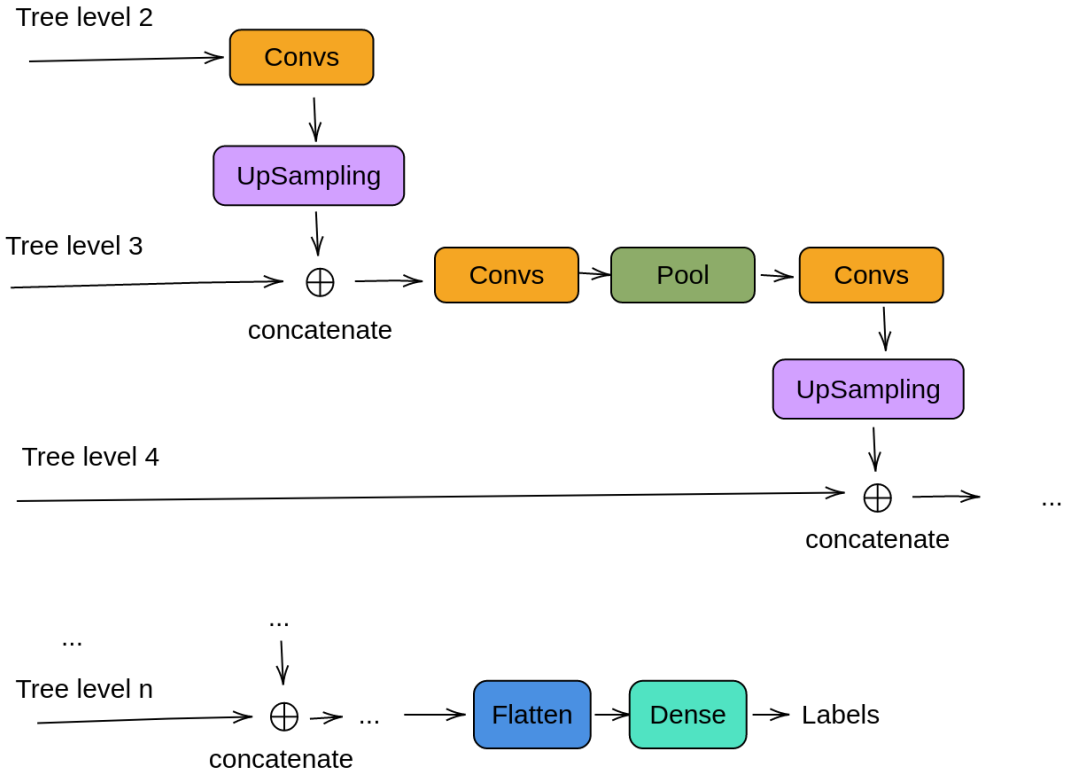


Figure 5.11: A MuSeReNet where information flows from the root to the leaves.

5.3.3 Experiments

In order to explore the effectiveness of our architecture for information retrieval from symbolic music and to check the compatibility of 1D CNNs for the task, along with the effect of allocating

resources to network depth or to kernel size we conducted a set of experiments [4].

Data

For our experiments, we use the Lakh Pianoroll Dataset as presented by Dong et al in [48], specifically the LMD-matched subset. This dataset consists of pianoroll representations of MIDI files in the Lakh MIDI Dataset presented by Raffel in [159]. The pianoroll is an array representation of music in which columns represent time at a sample rate of n samples per quarter note and rows represent pitch in the form of MIDI note numbers. The LMD-matched subset contains pianorolls that have been linked with the Million Song Dataset (MSD) [15]. The Million Song Dataset is the largest currently available collection of audio features and metadata for a million contemporary popular music tracks. We use labels acquired by MSD to construct the MASD and top-MAGD datasets presented by Schindler et al in [176], so we can compare our results with existing work. At the time of writing, Ferraro and Lemström in [56] have achieved the best results with regards to genre classification of symbolic music for the MASD and top-MAGD datasets. Finally, we randomly split each dataset into a train and test set (.75/.25), we use the train set for training our models and the test set for evaluating them.

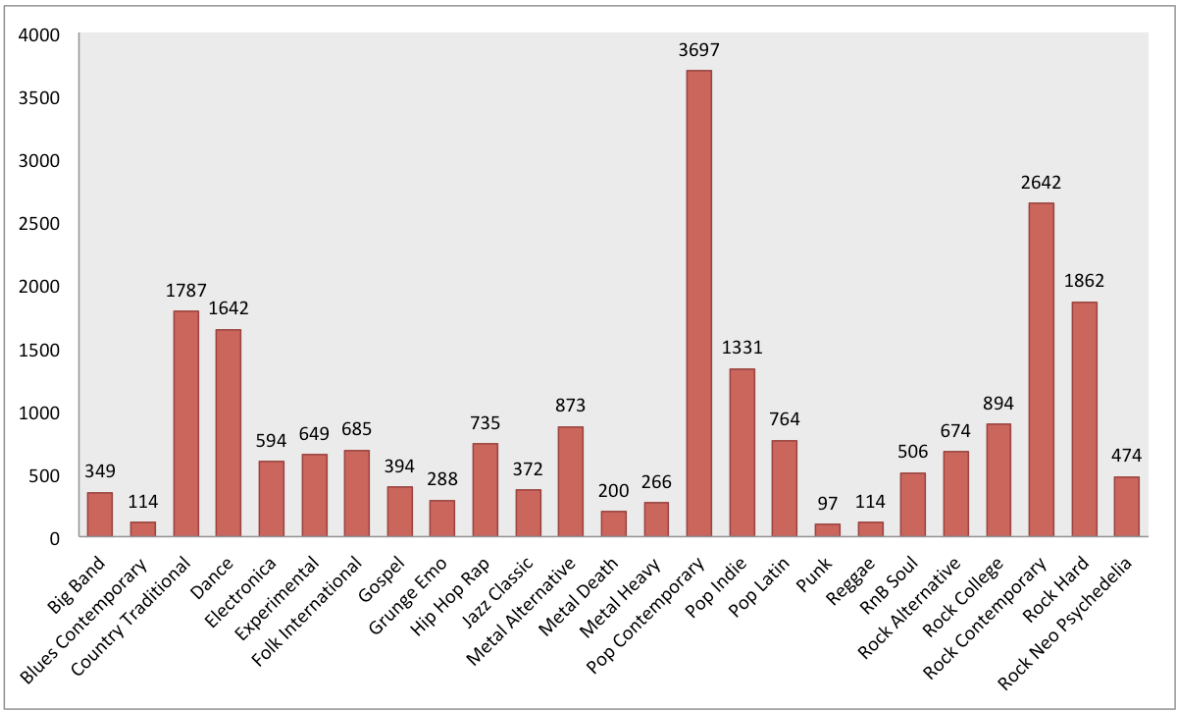


Figure 5.12: Number of files in the LPD dataset per label of the MASD dataset

Both datasets are imbalanced with regard to the number of files corresponding to each label (Figures 5.12 and 5.13). Methods such as over-sampling rare classes and under-sampling common classes could be used to potentially improve the generalization ability of trained models, but it is left for future work since we are interested in observing the behaviour of different network configurations for this task.

Models

We construct neural networks by using blocks of 1-D convolutions followed by max-pooling operations of kernel size and stride 2. Specifically, we use a shallow block, which consists of only one

[4] The code is available in the following GitHub repository: https://github.com/kinezodin/cnn-midi-genre

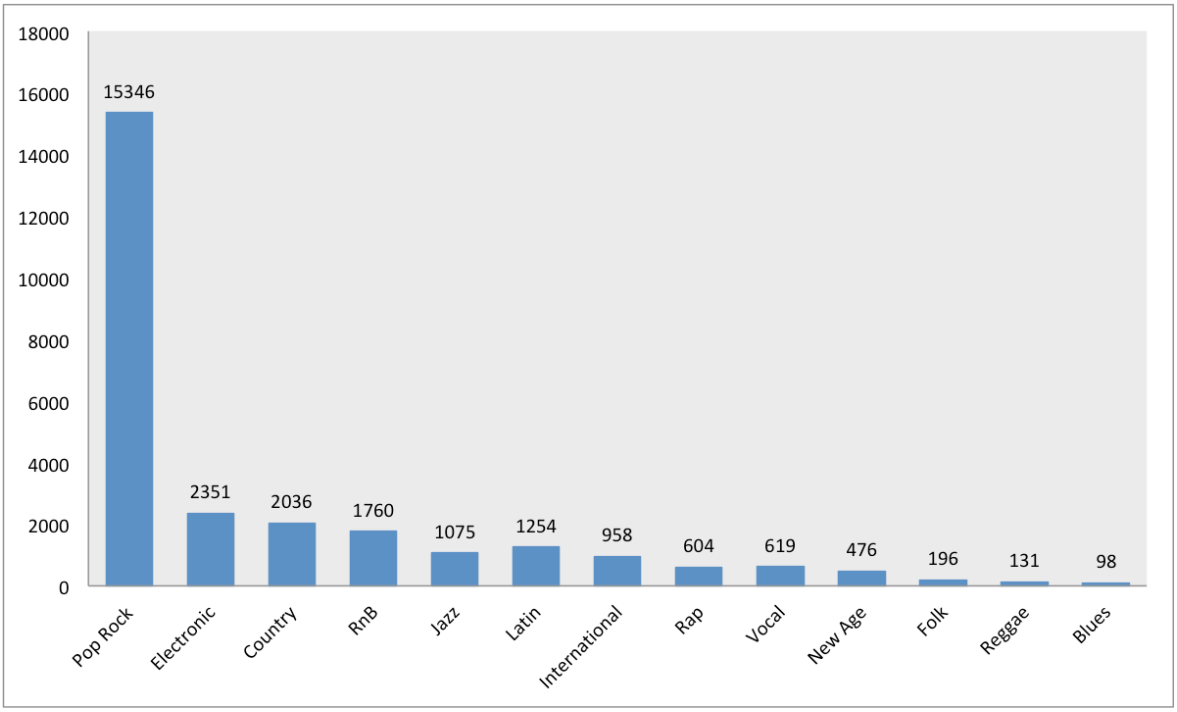


Figure 5.13: Number of files in the LPD dataset per label of the topMAGD dataset

convolutional layer prior to the pooling operation, and a deep block which consists of three convolutional layers before each pooling operation (Figure 5.14). A network built with shallow blocks will be referred to with the prefix 'shallow', and those built with deep blocks 'deep'.

In addition, each network has a 'Sequence' version in which blocks are stacked depth wise and the first block receives as its input the original sequence, and a 'MuSeRe' (Multiple Sequence Resolution) version in which a level of a tree constructed from the input sequence is concatenated to the output of each block. The first case represents a traditional CNN architecture, while the second represents MuSeReNets in which information flows from the leaves to the root (Figure 5.10).

Shallow vs Deep All blocks are individually set to have a similar receptive field of 24 samples, thus stacking the same number of blocks will lead to CNNs with the same receptive field regardless of which type of block is used. In the shallow block case, this implies a kernel size of 24. For the deep block case, assuming all convolutions have the same kernel size k, if $k = 9$ the receptive field at the output of the third layer is 25 input samples. We arbitrarily chose the smallest kernel size $k = 9$ which has the same number of trainable parameters as a 3×3 kernel which is popular for computer vision two-dimensional CNNs.

All blocks are also set to have a similar number of trainable parameters. Given fixed input dimensions of $(|x|, f_{in})$, a single convolutional layer with f_{out} kernels of size k will have n_p trainable parameters:

$$n_p = (f_{in} * k + 1) * f_{out}$$

We set $f_{out} = f_{in} = 128$ for all blocks, thus the number of trainable parameters for a shallow block is:

$$n_{shallow} = 393,344$$

For the deep block, we set the number of kernels of the third layer $f_{out_3} = 128$, so that the output of each block is the same shape as with the shallow block. In order to satisfy the condition of having an equal number of trainable parameters to the shallow case

$$n_{deep} = n_{shallow}$$

$$1153f_{o_1} + 9f_{o_1}f_{o_2} + f_{o_2} + 1153f_{o_2} + 128 = n_{shallow}$$

, where f_{o_1} and f_{o_2} are the number of kernels for the first and second layers of the deep block. By arbitrarily setting $f_{o_1} = f_{o_2}$ we get 117 kernels per layer. This way we end up with the two blocks shown in Figure 5.14.

For the different models, we use powers of 2 as input sequence lengths l, ranging from $l = 64$ to $l = 2048$. In the context of our dataset, these lengths represent musical time from approximately 5 quarter notes to 170 quarter notes, or 42 bars for a $\frac{4}{4}$ time signature (around one to two minutes for typical values of a song's tempo). Then each network will consist of $\log_2 l$ blocks stacked depth-wise, followed by a fully connected layer at the output, with as many sigmoid-activated units as there are different labels in each dataset. For all convolutional layers, we used ReLu activations.

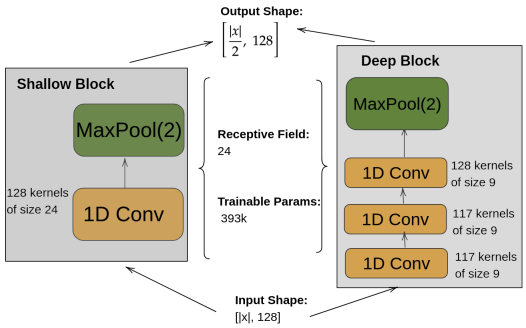


Figure 5.14: The convolutional blocks used to construct the CNNs for use in our experiments

Sequence vs MuSeRe The two versions of each network (Figure 5.15) differ with regard to the inputs of each block which are sequences of 128-dimensional vectors in the 'Sequence' case and 256-dimensional vectors in the 'MuSeRe' case, which are a result concatenation of a previous block's output with the original sequence at a lower resolution and leads to an increase of trainable parameters for the first layer of each block. The 'Sequence' networks are a typical 1D CNN architecture, which however, to our knowledge, have not been used in an end-to-end approach for symbolic music inputs and genre recognition.

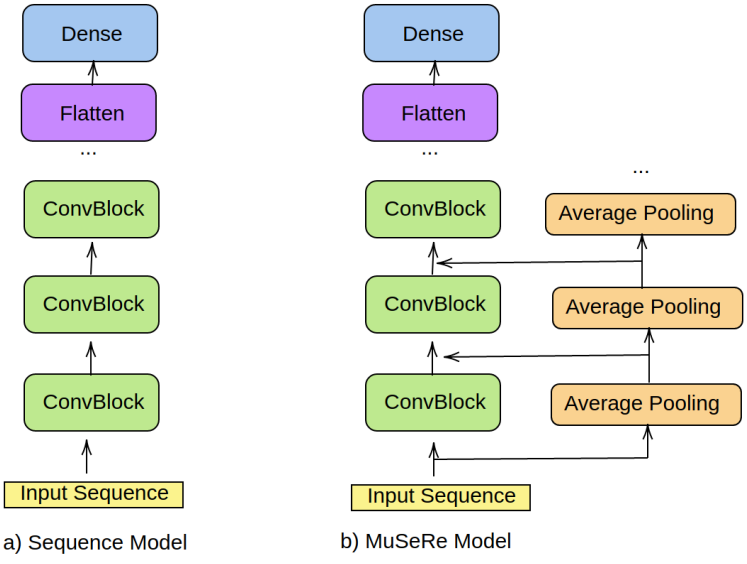


Figure 5.15: a) Sequence architecture and b) MuSeRe architecture used in experiments

Data Preparation and Training

Every piano-roll is a fixed length sequence of vectors $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_t}]$, where each vector x_i has 128 dimensions representing MIDI note numbers. During training, before feeding a sequence to a network, we perform a random transposition by shifting elements of every vector of a sequence by a random integer in $[-6, 6]$. This corresponds to transpositions up to a tritone below or above and is done as a data augmentation step which helps to avoid bias with respect to a particular tonal center.

The way multi-track MIDI files are handled is by averaging the pianorolls of all instrumental tracks and all percussive tracks separately into two cumulative pianorolls. The downside is that we lose information pertaining to the different instruments and to some extent different voices become convoluted, but this way we are able to process a larger number of MIDI files regardless of the number of tracks.

We use a data generator to create batches for training and apply any data transformations on the fly. For finding trainable parameters of networks which minimize the cross entropy between predicted genres and real genres we use Adam [105] as the optimization algorithm during training with a learning rate of $\alpha = 10^{-5}$ and for the other hyper-parameters of the optimizer $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a batch size of 32. Before training a model, we further split the original train set into a validation set and a train set (0.2/0.8) randomly. We then trained our models for up to 300 epochs while using an early stopping criterion for each model's F1-score on the validation set.

Evaluation and Post Processing

We evaluate our models on the held-out test set for each of the MASD and topMAGD datasets by computing precision, recall, and micro f1 metric. Due to the varying sequence lengths of pianorolls in the test set, we use the post-processing procedure described below to aggregate a model's prediction across whole sequences, which are of greater length than the expected neural network inputs.

Given a pianoroll x_p of sequence length N and a model with input length $N_m < N$, we retrieve sequences of length N_m from x_p by using a sliding window of N_m samples and a hop size of $\frac{N_m}{2}$ samples on the sequence. Each window is then fed to the model for inference, which returns a vector where each element represents a probability that a specific label is assigned to x_p. Then we assign as predicted labels those with a probability value greater than 0.5. If no labels have a probability greater than 0.5 then we assign as a single label the element of the vector which has the maximum probability, since there are no unlabeled samples in the dataset. These predictions are then used to calculate false and true positives and negatives, recall, precision, and F1 score.

Results

The results of our experiments are shown in Table 5.5 which lists the micro F1 scores of each trained model on the test set. In general, all CNNs which were trained on sequences longer than 256 samples surpassed Ferraro and Lemström's pattern recognition approach, as well as Liang et al. model with regards to F1 metric [56] [120]. Increasing input length by a factor of two along with increasing the number of blocks by one in most cases improved performance, with a notable exception for the longest sequence lengths that we experimented on (1024 vs 2048). In general, MuSeRe models outperform sequence models for shorter input lengths. The poor performance of MuSeRe models for larger inputs could be a result of overfitting, but requires further experimentation.

In addition, we present precision, recall, and F1 scores for each label in the topMAGD dataset for the best performing model (Table 5.6). On the one hand, the effect of the imbalanced dataset is apparent in the network's performance for the most common label (Pop Rock) when compared to those with fewer files in the dataset such as Blues, Reggae, and Folk. It is interesting that genres such as Jazz which have little representation in the dataset are better classified than genres such as Electronic which has almost double the support. This could be due to distinguishing musical characteristics of each genre, which are apparent in symbolic representations of music - for instance,

Table 5.5: micro F1 scores on the test sets of the MASD and topMAGD dataset for each of our architectures P2-4 and P2-5 refer to the best performing configuration of those presented in [56] and PiRhDy_GM refer to the best performing configuration of those presented in [120]

Length	Block	Input	MASD	topMAGD
64	Deep	Sequence	0.258	0.620
		MuSeRe	0.265	0.622
	Shallow	Sequence	0.295	**0.623**
		MuSeRe	**0.308**	0.622
128	Deep	Sequence	0.315	0.624
		MuSeRe	0.317	0.631
	Shallow	Sequence	0.361	0.632
		MuSeRe	**0.407**	**0.639**
256	Deep	Sequence	0.411	0.654
		MuSeRe	0.404	0.639
	Shallow	Sequence	0.335	0.663
		MuSeRe	**0.491**	**0.668**
512	Deep	Sequence	0.456	0.661
		MuSeRe	0.374	0.653
	Shallow	Sequence	**0.545**	**0.711**
		MuSeRe	0.525	0.703
1024	Deep	Sequence	0.507	0.673
		MuSeRe	0.337	0.641
	Shallow	Sequence	**0.581**	**0**.777
		MuSeRe	0.526	0.737
2048	Deep	Sequence	0.456	0.696
		MuSeRe	0.264	0.627
	Shallow	Sequence	**0.593**	**0.759**
		MuSeRe	0.444	0.733
P2-4			0.468	0.662
P2-5			0.431	0.649
PiRhDy_GM			0.471	0.668

jazz music tends to have complex harmony and utilize more notes, while electronic music tends to contain loops of very few notes.

5.4 Explainability for Genre Recognition

Even though we have shown that deep convolutional neural networks can perform better than other methods for genre recognition from symbolic music, they suffer from drawbacks that most deep learning approaches do, importantly lack of explainability and interpretability. This is not an imperative issue for genre recognition, since explainability is not critical for the method to be utilized, however, it would be a useful feature that could help improve performance in a future iteration by helping detect potential biases or flaws of the genre recognition model. Furthermore, since genres

Table 5.6: Per label precision recall and F1 score on the test set for Shallow Sequence model with input length 1024 (best performing model) on the topMAGD dataset

Label	F1	Precision	Recall	Support
Pop Rock	0.86	0.81	0.96	3705
Electronic	0.58	0.74	0.47	557
Country	0.67	0.83	0.56	502
RnB	0.61	0.92	0.45	432
Jazz	0.76	0.91	0.65	281
Latin	0.45	0.78	0.32	338
International	0.53	0.77	0.41	236
Rap	0.34	0.78	0.22	133
Vocal	0.65	0.90	0.51	150
New Age	0.66	0.94	0.51	116
Folk	0.48	1.00	0.32	44
Reggae	0.48	1.00	0.31	38
Blues	0.55	0.73	0.44	18
micro avg	0.78	0.81	0.74	6550

themselves are not well-defined terms, and their characteristics can vastly change over time, explanations of predictions could be valuable for understanding both the model and the dataset. In this section, we generate explanations by adapting various explainability frameworks and tools from the area of explainable AI (XAI) to the domain of symbolic music. As these are *post hoc* explanation methods, which treat the model as a black box, we chose to present a high-level description of multiple different methodologies, since such methods have been shown to occasionally produce misleading results [172]. We analyze and qualitatively compare the usefulness of different explanation methods when applied to symbolic music.

5.4.1 Local Explanation Methods

Methods for explaining the prediction of a black-box model on a specific sample are called local explanation methods [75]. There are many such methods in the relevant literature, for various types of data, however, none have been designed specifically for symbolic representations of music. We show results generated by Grad-CAM [179], LIME [165] and the genetic programming based GPX [57]. Grad-CAM generates visual explanations and is intended for images, LIME works on any type of data, while GPX is more suited for tabular data with a relatively small number of features. Here we show visual explanations for Grad-CAM and LIME, where the image depicts the cumulative pianoroll of each piece of music, while for GPX we modify the explanation pipeline for it to be utilized on symbolic representations of music.

The results concern four hand-picked samples from the reddit MIDI dataset[5]. These are (a) Beethoven's *Moonlight Sonata*, (b) The Beatles - *Here Comes the Sun*, (c) Eminem - *The Real Slim Shady* and (d) Queen - *Bohemian Rhapsody*. We fed the first 1024 time-steps of each sample through the best performing (on the topMAGD dataset) CNN. The predictions are shown in Table 5.7. We chose Moonlight Sonata, since it is out of domain as its genre is not included in the dataset's labels. The top two predictions of *International* and *New Age* music for the specific sample are interesting and merit an explanation. We chose (b) and (c) as examples of certain predictions, while (d) shows

[5] `https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/`

Table 5.7: Predictions for the top-4 genres for the first 1024 time-steps of each of the four songs for which local explanations were generated. The tracks are: i) Beethoven - Moonlight Sonata, ii) The Beatles - Here Comes the Sun, iii) Eminem - The Real Slim Shady, iv) Queen - Bohemian Rhapsody

Beethoven		**Beatles**		**Eminem**		**Queen**	
International	0.69	Pop - Rock	0.83	Rap	0.89	Electronic	0.54
New Age	0.40	Jazz	0.06	Electronic	0.03	Pop - Rock	0.20
Rap	0.25	RnB	0.04	Jazz	0.02	Vocal	0.06
Pop - Rock	0.24	Country	0.037	Vocal	0.003	RnB	0.04

an erroneous prediction by the CNN (Electronic), and a relatively high value of 0.06 for the vocal genre which is interesting since the introduction of the song features an *a capella* chorus.

Grad-CAM

Grad-CAM is a method proposed for generating visual explanations for large 2D CNNs which are applied in the image domain. It works by computing the gradient of the target output neuron with respect to the activation-map of the final convolutional layer of the CNN, averaging over the channels, thus leading to a coarse heatmap of the same size as the convolutional feature-maps, which in our case is always a sequence of length 4, due to the application of max-pooling operations within our networks. The resulting explanations for the top-2 predicted classes, for each of the four samples are shown in Figure 5.16.

Due to the coarseness of the heatmap, these explanations are not very useful in their own right, and are used in this context as a baseline. For *Moonlight Sonata* the explanations show uniform contribution of each timestep towards the International genre and no contribution towards the New Age genre. For *Here comes the Sun* the first quarter of the pianoroll seems to contribute more towards a *Jazz* prediction, while the third quarter contributes towards a *Pop-Rock* prediction. For *The Real Slim Shady* the explanations are similar to those of *Moonlight Sonata* and are not very useful. Finally, for *Bohemian Rhapsody*, the second half of the pianoroll contributes more towards the *Electronic* genre, after the piano part is introduced.

LIME

LIME is a technique for generating explanations for any classifier, in any data domain. In order to explain a classifier $f : \mathbb{R}^d \rightarrow \mathbb{R}$, LIME searches for a function $\xi : \mathbb{R}^d \rightarrow \mathbb{R}$ which is a) Interepretable and b) Approximates f locally. It is formulated as:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

where G is a family of explainable functions (such as linear models or decision trees), π_x is a proximity measure that measures locality around x, Ω is a measure of how interpretable a function g is (for instance the number of weights in a linear model, or the depth of a decision tree) and $\mathcal{L}$ is a distance function showing how closely g approximates f in the locality defined by π_x. In Figure 5.17 we show explanations generated by LIME using the official python package [6] provided by the authors, and specifically the lime_image object, with default parameters. For *Moonlight Sonata* LIME has highlighted the fifth measure without the melody note, the melody of the seventh and eighth measures and the bass notes of measures 13 and 14 as contributing towards an *International* genre prediction. For *New Age* only the fourth and fifth measures are highlighted by LIME. For *Here*

[6] https://github.com/marcotcr/lime

Figure 5.16: Explanations generated by Grad-CAM for the top-2 predicted genres, for the best performing CNN, on the first 1024 time-steps of Beethoven's Moonlight Sonata (a,b), The Beatles - Here Comes the Sun (c,d), Eminem - The Real Slim Shady (e,f), and Queen - Bohemian Rhapsody (g,h). The highlighted area with green represents the important for the prediction segments - according to Grad-CAM

Comes the Sun, the fifth and sixth bars, along with their repetition four bars later contribute more towards the *Pop-Rock* genre. For *The Real Slim Shady* the same three notes have been highlighted as contributing towards the *Rap* genre across two repetitions. A different repetition of the same notes has been highlighted as contributing towards *Jazz*. Finally, the first measure of the piano part of *Bohemian Rhapsody* along with its preceding measure contribute towards *Electronic*, while the third measure after the piano is introduced contributes towards *Pop-Rock*. These explanations seem to generally agree with those generated by Grad-CAM and are not very intuitive visually. However, by changing the representation of the MIDI files to either music notation (score) or audio, the high-

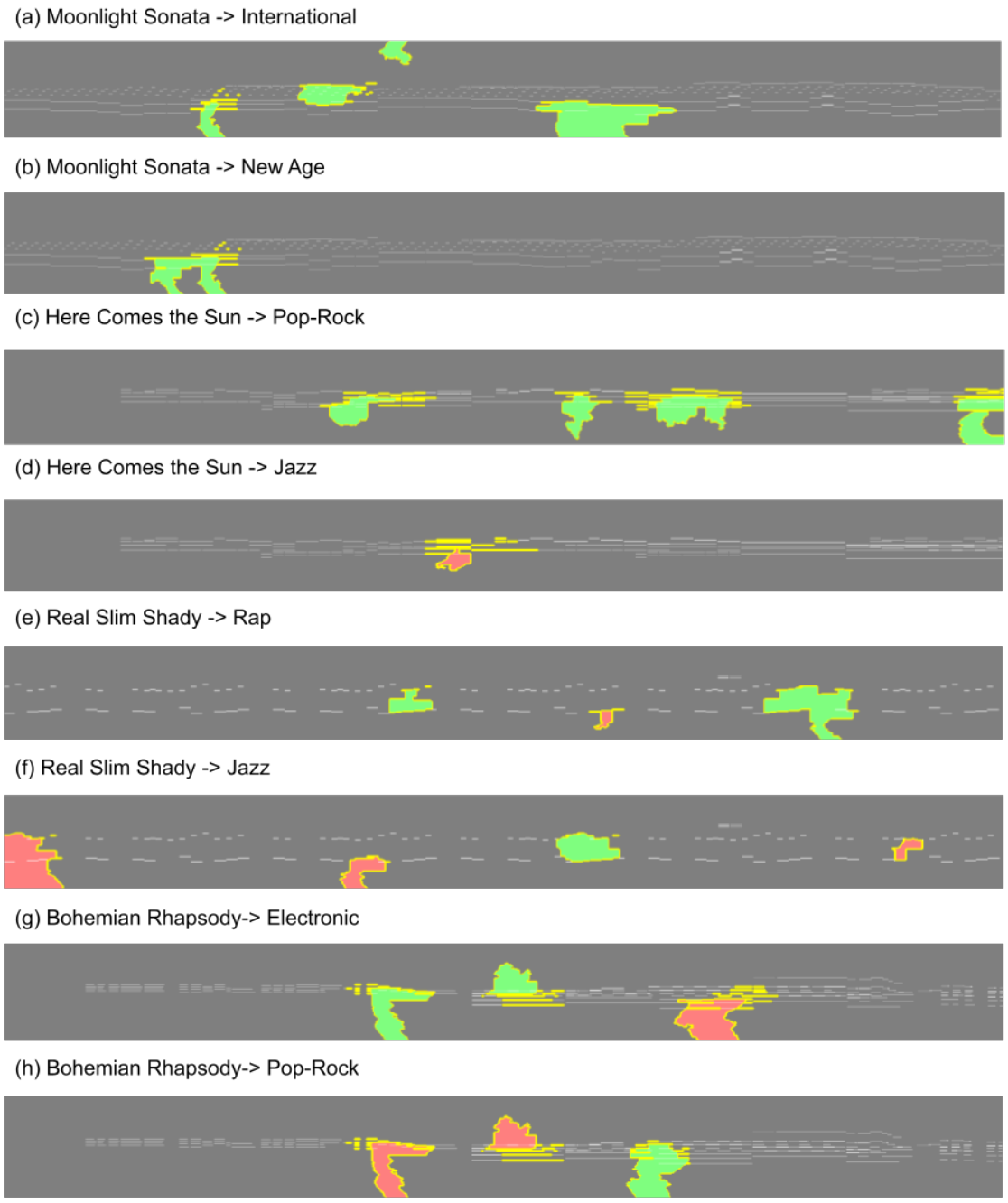


Figure 5.17: Explanations generated by LIME for the top-2 predicted genres, for the best performing CNN, on the first 1024 time-steps of Beethoven's Moonlight Sonata (a,b), The Beatles - Here Comes the Sun (c,d), Eminem - The Real Slim Shady (e,f), and Queen - Bohemian Rhapsody (g,h). With green are highlighted the areas of the pianoroll that positively contribute to the label prediction, and with red those that negatively contribute, according to LIME.

lighted parts may be observed in more detail and listened to. It would be interesting to analyze them from a music theoretical perspective, however, this is beyond the scope of this work.

Modified GPX

GPX [57] is a methodology for generating local explanations by utilizing genetic programming [108]. It is formulated similarly to LIME, in that GPX searches for a function $\xi : \mathbb{R}^n \to \mathbb{R}$ which attempts to mimic the original complex model $f : \mathbb{R}^d \to \mathbb{R}$ on a given sample set η, which is typically defined locally around the sample to be explained. The goal of GPX is to find:

$$\xi = \arg \min_{g \in G, s_i \in \eta} d([g(s_1), \dots, g(s_m)] - [f(s_1), \dots, f(s_m)]) \tag{5.21}$$

where d is a distance function, such as the l_2-norm. GPX generates functions g which are non-linear algebraic expressions in the form of binary trees on the given feature set. Genetic Programming (GP), in general, generates a random population and evaluates the fitness of each individual, in terms of effectiveness in solving the problem, favouring the better individuals. In this work, the GP evolves symbolic expressions for local explanation in the genre classification task.

We modified GPX in two ways in order to generate more meaningful explanations for our application: a) the way the local sample set η is generated and b) what features are used within the genetic program, and for producing explanations.

Local Sample Set In order to generate the sample set η around the sample x to be explained, GPX samples from a multivariate Gaussian distribution centered at x with a covariance matrix computed from the training data. By experimenting with this approach we realized that on the one hand, the predictions of the classifier on such a sample set were not varying significantly, and on the other hand, the samples generated with this approach were not meaningful as pianoroll representations, and the resulting samples did not have any musical meaning. Instead, we generate each η_i by randomly choosing a set of pitches which have a value > 0 in the cumulative pianoroll of x and transposing them by a random number of semitones in $[-10, 10]$. This way η, which is supposed to consist of samples in the neighborhood of x, will contain pianorolls that are similar to x and differ only with regard to some pitches. Notably, this approach does not impact the rhythmic characteristics of the pianoroll, since the only changes are made in the pitch dimension, so even if pitches are changed drastically for some samples, they can still be considered to be "local".

Feature Extraction Each cumulative pianoroll consists of 1024 time-steps of 128-sized vectors, which would translate to $128 \times 1024 = 131,072$ features per sample. Such a large amount of features makes the application of GPX infeasible, and even if time and memory complexity were not an issue, the explanations which would be generated would not be very informative, as GPX is a feature importance explainability method, and it is difficult to draw conclusions from such a large feature set. Instead, we incorporate a feature extraction function $h : \mathbb{R}^n \to \mathbb{R}^p$ within the explainer function, where n is the original number of features and $p < n$ is the number of extracted features, and reformulate Equation 5.21 as:

$$\xi = \arg \min_{g \in G, s_i \in \eta} d([g(h(s_1)), \dots, g(h(s_m))] - [f(s_1), \dots, f(s_m)]) \tag{5.22}$$

For applying this idea on symbolic music, we define h to extract thirteen features, which are intuitively linked to musical harmony. Specifically, for the first twelve features, we first get the prevalence of each pitch by summing x over the time dimension, leading to a 128-sized vector. Then from this vector, we get the prevalence of each pitch class, by summing each dimension modulo 12, leading to a vector of 12 elements, each of which represents the prevalence of a pitch class within the pianoroll. Finally, we transpose this vector by rolling it, such that the largest element is at position 0. The thirteenth feature is the average pitch across the whole pianoroll. This way, the function h, acts on a pianoroll x to produce a vector $h(x)$ which shows the prevalence of each musical interval when compared to the most used pitch class. The features $h(x)$ are then utilized by the explainer which attempts to mimic the black-box classifier, and are the features that appear in generated explanations.

For the genetic programming algorithm, we used a population size of 110 evolved over 110 generations, to attempt to mimic the classifier in a local neighbourhood of 11,000 samples η_i. For other hyperparameters, we used the same as those used in [57]. We ran the algorithm 5 times for each sample, and show the best results with regard to the explainer accuracy. We comment on the consistency of the results.

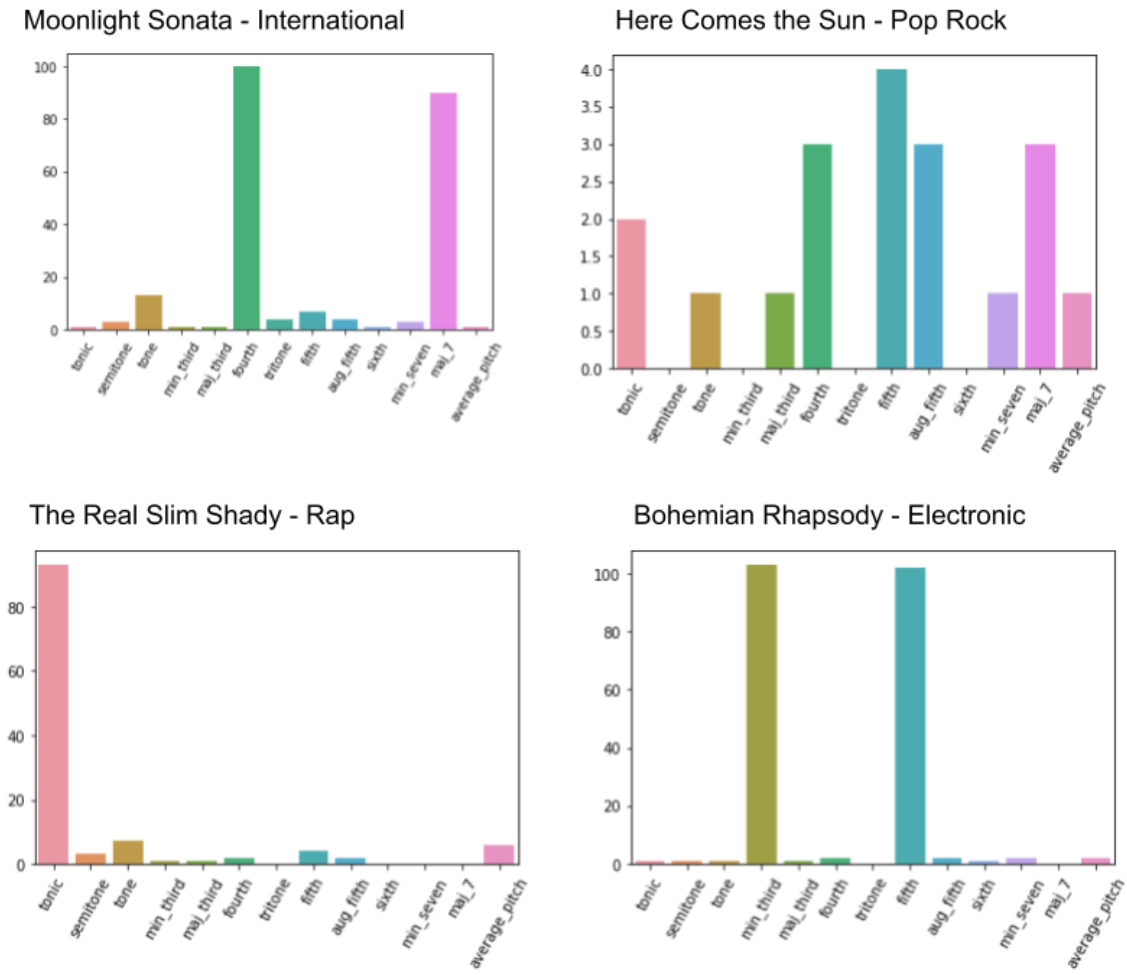


Figure 5.18: Feature importance for the top-1 predicted genre as generated by the modified GPX

In Figure 5.18 we show the feature importance generated by GPX for each prediction of the CNN. The feature importance is calculated as the number of appearances of each feature in the final population, which consists of 110 algebraic expressions. For Bohemian Rhapsody-Electronic and Moonlight Sonata-International, which are both erroneous predictions, GPX has shown two intervals as important features. This result was consistent for the case of Bohemian Rhapsody while for Moonlight Sonata the *maj_7* interval appeared in all 5 runs, but not the *fourth* interval. For the other two samples, the best explainer was a constant function, however, the existence of features in the final population can give us some musical insight. In Table 5.8 we show a summary of results for the top prediction for each of the four selected samples. For the local sample set generated around Here Comes the Sun and The Real Slim Shady, the CNN classified 0.75 and 0.93 as *Pop-Rock* and *Rap* respectively (percentage of positive η_i). Even though the best explainer for these two samples was (essentially) a constant function, the prevalence of the tonic note as an important feature for *Rap* classification in the final population makes sense intuitively.

In Figure 5.19 we show the final program evolved by GPX with the best explainer accuracy for each sample. For Moonlight Sonata (explainer accuracy 0.8) the program is $fourth$ - maj_7, where

the intervals are based on the extracted tonic of B. However we know that Moonlight Sonata is actually in $C\sharp$ minor, so in this context, the program would be $min\_third - sixth$. For the actual sample, the prevalence of the first interval is 0.59 while of the second it is 0.08, which would indicate that when the pitch class $B\flat$ is more prevalent, then the sample is no longer classified as *International*. In music-theoretical terms, $B\flat$ appears as a Dorian substitute and it would be interesting to explore the distribution of the Dorian mode within our dataset of *International* music, and determine if this is a bias learned by the classifier, or if it is an actual feature of the genre. These results are also somewhat consistent with the explanation generated by LIME (Fig 5.17) since for measures 13 and 14 in which the pitch class $B\flat$ appears for the first time, LIME has only highlighted the bass notes (which lack the pitch class) as contributing towards the *International* genre. For Bohemian Rhapsody (explainer accuracy 0.75), the best program generated by GPX is $\min(min\_third, fifth)$. In the actual sample the prevalence of the first interval is 0.63 and of the second is 0.51. This rule says that if both the minor third and the fifth appear at least half as often as the tonic, then the sample is classified as *Electronic*, which makes intuitive sense since *Electronic* music tends not to have complex harmony, and the rule represents the prevalence of a minor triad in the pitches which appear in the pianoroll. Again this merits further exploration to determine if it is a bias, or something useful that the CNN has learned. For the other two samples, the explanations generated are trivial. For Here Comes the Sun, we attribute the failure of GPX to the bias of the classifier towards the *Pop-Rock* genre which is a result of dataset imbalance. For The Real Slim Shady, we believe that all samples in the local sample set were classified as *Rap* due to the fact that the sample uses very few pitches, which doesn't change after randomly transposing them, and the sampling method does not affect rhythmic characteristics, which we believe to be a key feature for *Rap* classification.

Table 5.8: Summary of results of applying GPX with the feature extraction addition, for the top prediction for each sample.

	Beethoven	Beatles	Eminem	Queen
percentage of positive η_i	**0.69**	0.75	0.93	**0.62**
Explainer Accuracy	**0.80**	0.75	0.93	**0.75**
Extracted Tonic	B	A	E♭/D♯	B♭/A♯
Actual Key	C♯ min	A Maj	C min	B♭ Maj

5.4.2 Global Explanation Methods

Global explanation methods aim to explain the overall behaviour of a black-box, contrary to local explanations which concern the predictions of the model on a specific instance.

MMD-critic[103]

MMD-critic is a methodology for analyzing the distribution of a dataset in order to find specific samples which are prototypes, and others which are criticisms. The former are samples that are characteristic for a specific distribution while the latter are outliers. MMD-critic is based on the idea of Bayesian Model Criticism [61] and produces explanations by calculating *Maximum Mean Discrepancy* (MMD).

In order to produce global explanations for a black-box model with MMD-critic, we first calculate prototypes and criticisms for the test set, by feeding the algorithm cumulative pianorolls. Then, for each genre, we get the set of positive examples as predicted by the black-box and compute prototypes

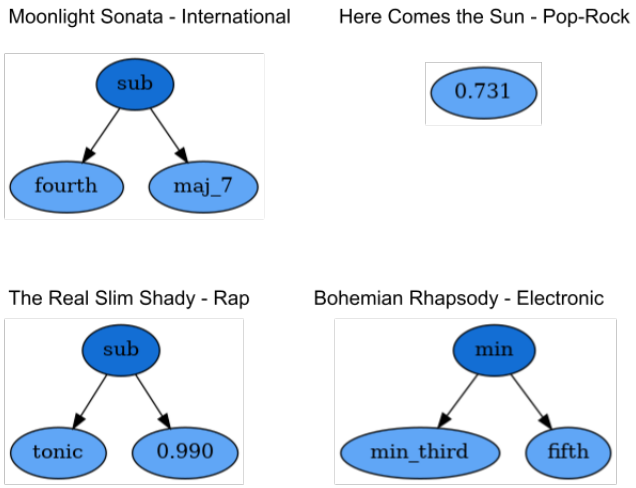


Figure 5.19: Final programs generated by GPX as explanations for the top prediction for each sample.

and criticisms for each of these sets of positive examples. This way we get prototypes and criticisms according to the real distribution (test set) in addition to the distribution learned by the black-box (predictions on the test set). In Table 5.9 we show a prototype and a criticism for each genre, as computed by MMD-critic on the test set of topMAGD, and in Table 5.10 we show a prototype and a criticism for each genre, as predicted by the CNN.

Regarding the results on the test set (Table 5.9), we can gain some insight about the dataset and by extension the performance of the CNNs. Firstly, this table raises the issue of ground truth reliability, which is one of the main difficulties for genre recognition. For instance, none of the *Reggae* samples in the table are actually *Reggae*, but would probably be considered Soul/RnB/Gospel. Furthermore, the prototype for the *Latin* genre doesn't have any characteristics of *Latin* music besides the language and would be considered Pop. Similarly, the prototype for *Rap* could be considered RnB (which often contains Rap in modern music), while the prototype for *New Age* could be considered New Wave/Pop instead. A second issue raised by Table 5.9 is that of dataset imbalance. This is not only regarding the number of samples for each genre, but also the range of different music each genre encompasses. For instance, the *Pop-Rock* genre is represented by a very diverse set of samples, ranging from Hard Rock to Disco. A result of this is that almost half of the selected prototypes and criticisms are labeled as *Pop-Rock* among other labels. Finally, for those genres with very low support, we cannot expect MMD-critic to produce meaningful explanations since it is a statistics-based approach that requires a sufficiently large dataset.

By studying the resulting prototypes and criticisms from the predictions of the CNN (Table 5.10), along with the performance of the CNN on each genre (Table 5.6) we are able to better understand what the CNN has learned. For *Pop-Rock* the prototype chosen by MMD-critic is a power ballad by Abba which is closer to Pop than Rock, which is interesting when compared to the prototype selected from the ground-truth labels: a Nine Inch Nails song which is a lot closer to Rock. For genres with more than 100 samples in the test set, in which the CNN does not perform well (Electronic, Latin, International, Rap) the generated prototypes are, as expected, far from representative of each genre, however, they are still useful for gaining insight on what the CNN has learned. For instance, the choice of Queen - *We Will Rock You* as a prototype for *Rap* could be due to the rhythmic qualities of the vocal track, the looping music, and the repeating patterns which are prevalent in a lot of different music, including *Rap*. This could help us understand the poor performance for the specific genre (0.34 F1 Score), along with the high precision (0.78).

Table 5.9: Prototypes and Criticisms for each genre, generated by MMD-critic on the test set of topMAGD. In parentheses are the ground truth labels.

Genre	Test Set Prototype	Test Set Criticism
Pop-Rock	Nine Inch Nails - Piggy (Pop-Rock)	Spice Girls - Wannabe (Pop-Rock)
Electronic	Toy-Box - Tarzan & Jane (Pop-Rock, Electronic)	George Michael - Fast Love (Electronic)
Country	Session Americana - John Brown (Country)	Olivia Newton-John - Everything Love Is (Pop-Rock, Country)
RnB	Mariah Carey - If It's Over (Pop-Rock, RnB)	Tina Turner - Steamy Windows (RnB)
Jazz	Lee Ritenour - Papa Was A Rolling Stone (Jazz)	Procol Harum - A Whiter Shade Of Pale (Pop-Rock, Jazz)
Latin	Yahir - Fue Ella, Fui Yo (Latin)	Os Paralamas Do sucesso - Romance Ideal (Latin)
International	Mamonas Assassinas - Pelados Em Santos (International)	Mamonas Assassinas - Pelados Em Santos (International)
Rap	50 Cent - Baby By Me (Pop-Rock, Electronic, Rap)	Bobby Brown - Don't Be Cruel (Pop-Rock, Jazz)
Vocal	Nana Mouskouri - Habanera (International, Vocal)	Salvatore Licitra - E Lucevan Le Stelle (Vocal)
New Age	Cock Robin - The Promise You Made (New Age)	Slavic Soul Party! - Never Gonna Let You Go (Jazz, New Age)
Folk	Judy Collins - Send In The Clowns (Folk)	Edison Lighthouse - Love Grows (Folk)
Reggae	Johnnie Taylor - For Your Precious Love (Pop-Rock, RnB, Reggae)	The Elgins - When A Man Loves A Woman (Pop-Rock, Country, Latin, Reggae)
Blues	Deborah Coleman - Long Time (Pop-Rock, Blues)	Jim Reeves - I Won't Forget You (Country, Blues)

5.5 Discussion: Knowledge Representation for Music

So far in this chapter, we have described our approach to evaluation of AI generated music, our methodology for genre recognition from symbolic music, and an analysis of *post hoc* explanations in this domain. From the above research, it became apparent to us how useful utilizing music theoretical notions is for music related tasks. For the evaluation procedure (section 5.2) the notions used from music theory were the circle of fifths, the tonnetz, and underlying ideas such as pitch classes, intervals etc. For the genre recognition methodology, we tried to utilize structural characteristics of music, via the structure of the neural architecture, making use of multiple resolutions. For explainability, we again used simple notions such as intervals and key signatures to get more meaningful explanations when using methods such as GPX. Furthermore, the explanations were interpreted by us in many cases, under the prism of music theory. However, all of the above utilizations of music theory were done in an ad hoc fashion, where the knowledge was represented differently in each case, to suit each methodology.

The above makes apparent the need for encoding music theoretical knowledge in a standardized, computer-readble, and human-understandable format. To this end, there exists related work, such as for example the work of G. Widmer [205] from 1994, that shows the potential of combining AI with

Table 5.10: Prototypes and Criticisms for each genre, generated by MMD-critic on the positive examples as predicted by the black box, on the topMAGD test set. In parentheses are the ground truth labels.

Genre	Black-box Prototype	**Black-box Criticism**
Pop-Rock	Abba - The Winner Takes It All (Pop-Rock, Vocal)	Donny Osmond - This Guy's In Love With You (Pop-Rock)
Electronic	Siniestro Total - C'est Chic (Pop-Rock)	Crystal Waters - 100% Pure Love (Electronic)
Country	Boots Randolph - Bridge Over Troubled Water (Country)	John Fogerty - Big Train (Pop-Rock)
RnB	Stevie Wonder - You Are The Sunshine Of My Life (RnB)	Whitney Houston - So Emotional (RnB)
Jazz	Abba - Take A Chance On Me (Pop-Rock)	Vince Guaraldi Trio - Christmas Time Is Here (Jazz)
Latin	Luis Miguel - El Dia Que Me Quieras (Latin)	Os Paralamas Do Sucesso - Romance Ideal (Latin)
International	Brasilian Tropical Orchestra - Yesterday (International)	Uniting Nations - Uniting Nations (Electronic)
Rap	Queen - We Will Rock You (Pop-Rock)	Phish - Wading In The Velvet (Pop-Rock)
Vocal	Michael Crawford - The Phantom Of The Opera (Pop-Rock, Vocal)	Collin Raye - Little Rock (Country, Vocal)
New Age	Lionel Richie - Hello (New Age)	Enya - China Roses (New Age)
Folk	The Roches - Do You Hear What I Hear? (Folk)	The Roches - It Came Upon A Midnight Clear (Folk)
Reggae	The Elgins - When A Man Loves A Woman (Pop-Rock, RnB, Reggae)	Johnnie Taylor - For Your Precious Love (Pop-Rock, RnB, Reggae)
Blues	Bill Quinn - He'll Have To Go (Country, Blues)	Jim Reeves - He'll Have To Go (Country)

music theory. Another example is HarmTrace [35], that has encoded tonal harmony into a context free grammar, and implemented in Haskell, to be used for the automatic harmonic analysis of chord progressions. There also exist description logics knowledge bases, and ontologies written in the web ontology language OWL [93], such as the music theory ontology [160] that defines useful musical terminology, and the functional harmony ontology [99] which we have developed, and allows for automatic harmonic analysis of chord progressions, according to the theory of modal harmony.

There is however a long way to go, both for extending these knowledge bases, especially to be inclusive to music from other cultures, and to not be focused on western music, and for including additional useful musical information, such as rhythm, structure and melody. Furthermore, research concerning the utliziation of such knowledge in machine learning pipelines, seems promising for the future of music and AI.

5.6 Conclusion

While new methodologies for solving music related tasks are rapidly evolving, following the general trend of artificial intelligence, there still exist important aspects of it that are underexplored. The main problems we uncovered during our research were: The lack of a consistent evaluation framework for music composition systems, the lack of a reliable ground truth in baseline datasets, and the unsuitability of out-of-the-box explainability methods for symbolic representations of mu-

sic. For all three problems, we believe that formal knowledge representation, and knowledge graphs can serve as tools for mitigating them. Concerning the evaluation framework, we can use encoded music-theory to analyze AI generated music, and consequently come up with metrics, similar to the heuristics of section 5.2, that may be used for evaluation. Regarding the unreliability of ground truth for music datasets, such as ill-defined genres whose characteristics change over the decades, or mood and emotion tags that can be subjective and unclear, formal knowledge representation can aid by encoding the definition of the labels in the knowledge. For example what makes a song “Pop” or “Rock”.

Chapter 6

Explainability for COVID-19 audio classification

6.1 Introduction

The COVID-19 pandemic brought forth many challenges to the scientific community, from testing, diagnosing and contact tracing, to prevention, treatment and identification of risk factors. Over the past decade there have been significant technological advancements, especially in the area of Artificial Intelligence and Deep Learning, which motivates researchers to find novel solutions to the aformentioned challenges by making use of these technologies. Specifically, when it comes to testing, quickly diagnosing new infections is crucial. However, the current methods, like RT-PCR tests and CT scans, have limitations. They can vary in accuracy, take a long time, and require trained staff, specialized labs, and expensive equipment, while antigen tests, as an alternative, are not very sensitive [3]. One promising approach is using mobile health (m-health) to make testing faster, more affordable, and accessible for multiple rounds. This could help control the spread and prevent resurgence [80]. In this context, the idea is to harness AI and mobile technology to create an easy-to-use and widely available COVID-19 detection method. This involves analyzing audio recordings of coughs, voices, and breath to identify new COVID-19-related markers [3, 63].

In recent studies, most efforts to predict COVID-19 risk from audio recordings use deep learning models. These models need a lot of data to work well. So, creating well-organized COVID-19 audio datasets is really important to make predictions accurate and dependable [63]. Many studies have tried to gather audio recordings from people using the internet. The first project doing this was the COVID-19 Sounds project [209]. They collected 53,449 audio samples, each with 3 to 5 deep breaths, 3 coughs, and 3 voice repeats of a set sentence. Another project called Coswara also collected sounds like breaths, coughs, voice, and counting numbers from people [181]. There's also Coughvid, which is a big database with cough sounds [148]. The latest version of Coughvid has 27,550 cough recordings. These databases have different numbers of audio samples, from 2,030 to 53,449. However, it is important to know that the number of COVID-19 cases in these datasets is relatively low, especially in Coughvid and COVID-19 Sounds. These databases also have information about the people's demographics, symptoms, and other health conditions to help find COVID-19 [63].

Developing accurate machine learning models to detect COVID-19 is challenging due to various factors. These include differences in available datasets, a low number of COVID-19 cases compared to controls, variations caused by COVID-19 variants, and the influence of factors like vaccination status. Additionally, there are biases in the data that need investigation, and complex modeling strategies can lead to over-fitting [80]. Researchers have tested the performance of audio-based COVID-19 testing by intentionally introducing biases into the dataset, such as gender bias, to see how it affects the model's effectiveness [80]. Another challenge is organizing and interpreting the data effectively to ensure transparency and trust. This involves creating user-friendly interfaces that explain COVID-19 risk estimates in understandable ways, making the AI more human-centered. To develop responsible AI models, data needs to be well-annotated with metadata and expert labels. This additional information can be used to train explainable AI models, which are easier for humans to understand than raw audio data. It can also be used to analyze black-box classifiers, which are commonly used for COVID-19 detection from audio recordings [20, 25, 29].

In this chapter, we first present an outline of our system that jointly won the IEEE COVID-19

Sensor Informatics challenge. Notably, one of the criteria for judging the entries to the hackathon was explainability, which in this work relied on *post hoc* methods from literature, such as LIME [165]. Next, we describe how we developed a crowd-sourced dataset within the context of the smarty4covid project [219]. This data was then utilized for inherently interpretable COVID-19 classification from tabular data, where included are features such as demographics, symptoms, pre-existing conditions, in addition to audio features that were extracted with signal processing. Furthermore, we utilize this crowd-sourced dataset to create an explanation dataset from scratch, and show resulting explanations of the classifier that was developed for the hackathon using the proposed explainability framework. Finally, we compare the three approaches, and discuss their differences (Deep Learning and feature-based post hoc explanations, interpretable machine learning on tabular data, and the proposed knowledge-based post hoc explanations).

6.2 COVID-19 Audio Classification

Classification of audio signals is in general a difficult problem, with important applications in various areas such as speech and music. Audio data is typically represented as a time-series of one (mono) or two (stereo) channels, at a resolution of 44,100 samples per second (CD quality). The high resolution of CD quality audio data is almost prohibitive for general time-series models, and even though some specialized neural network architectures have been proposed [146, 116, 145] to handle raw audio data, most audio classification pipelines operate in the time-frequency domain [197, 66]. This implies stages of feature extraction and signal processing, which makes the development of audio classification pipelines more challenging than for instance image classification.

Given audio representations in the time-frequency domain, the problem can be approached by adapting methods from computer vision, such as 2D Convolutional Neural Networks (CNN) [90], or general time-series prediction models such as Recurrent Neural Networks, 1D CNNs [10, 221] or transformers [66]. However there does not exist a "go-to" architecture for audio classification, and in the literature different approaches work best for different audio classification tasks, depending also on the domain.

By using deep learning to solve an audio classification task, we inadvertently sacrifice interpretability and tranceparency of the classifier, since the size and complexity of deep neural architectures makes them essentially black boxes. In critical domains such as medicine or law, interpretability is essential for deep learning to be ethically utilized. Some of XAI methodologies have been adapted to the audio domain [140, 12], however these are usually visual explanations which are on the one hand hard to understand, and on the other hand might be misleading [158]. Other approaches offer listenable explanations [87, 135], however these are specialized for the domain of music.

In this section we describe a pipeline for audio classification for COVID-19 diagnosis [20, 96], which operates in the time-frequency domain, uses a 2D CNN architecture and is, to an extent, interpretable by design. This pipeline jointly won the IEEE Covid-19 sensor informatics challenge.

6.2.1 System Description

Our system [1] consists of three identical simple CNNs: one accepts cough data, one speech and one breathing. The inputs to the CNNs are short segments of mel scaled spectrograms generated with librosa's [2] melspectrogram feature with the default parameters. Each timestep of the spectrogram corresponds to about 0.01 seconds of audio, while each segment correpsonds to about 1.5 seconds. There are three reasons for which we chose to use short segments combined with simple CNNs.

- **Receptive Field** : Our intuition is that low-level features are more important for the given task, thus we designed a pipeline where the networks have a small temporal receptive field.

[1] https://github.com/kinezodin/ntuautn
[2] https://librosa.org/

- **Data Scarcity** : The development dataset provided [181] has data from only 965 subjects, which would make a more complex model more prone to overfitting.

- **Interpretability** : By feeding short segments to the black-box CNN only low-level features are obfuscated and it is easy for us to determine which segments contribute more to a positive diagnosis. Furthermore, having identified important segments, we can use XAI methodologies to further explain a prediction on a specific segment.

An overview of our system is shown in Figure 6.1.

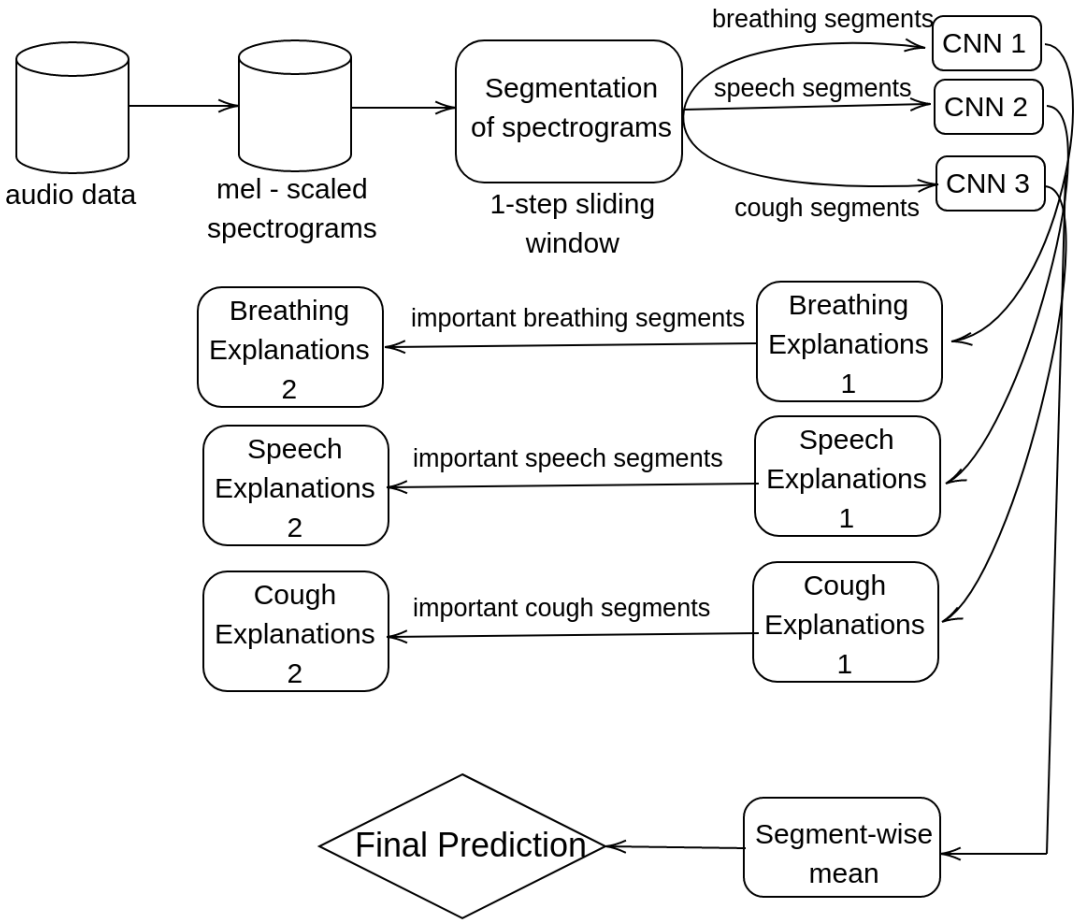


Figure 6.1: Overview of the system

Neural Network Architecture

Each CNN has only 3 layers of $[32, 64, 64]$ kernels of size 16×16 and ReLU activations, with a max pooling layer of stride 2 between each convolutional layer. The output of the final convolutional layer is flattened and fed through a sigmoid activated neuron for the final prediction. The input to the CNN is a short segment of 128 timesteps of a spectrogram, which corresponds to 1.5 seconds, with a frequency resolution of 128.

Training

Each CNN was trained independently from the others on relevant data. The training procedure for a single CNN is shown in Figure 6.2. Specifically, we use a data generator to create batches on the fly during training. These batches are balanced by oversampling the rare class (positive subjects). Of each audio file, a single random segment is added to the batch at each step. Each segment is then augmented by randomly shifting and by adding noise before being fed to the corresponding CNN. We use binary cross-entropy as the loss function and Adam [105] as the optimization algorithm.

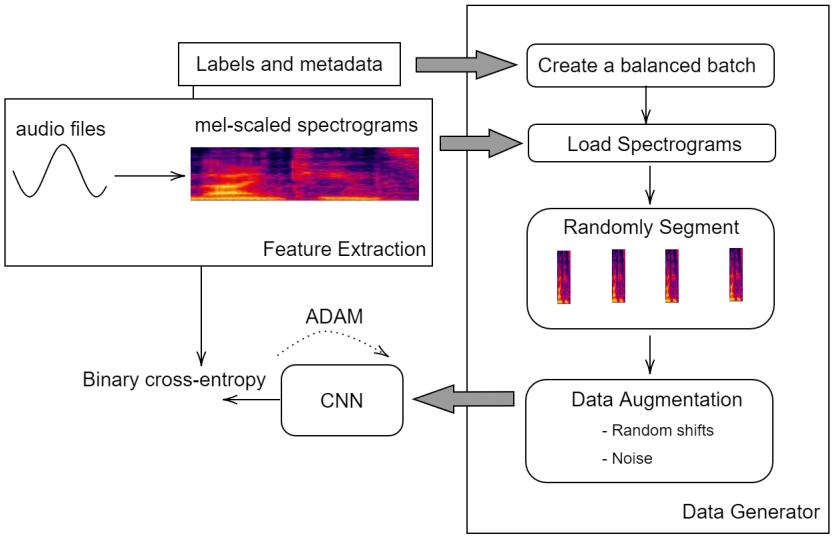


Figure 6.2: Training pipeline for a single CNN

Inference

During inference, given three audio files corresponding to coughing, breathing and speech, we first compute a mel-scaled spectrogram for each and then split them into overlapping segments with maximum overlap. For instance a spectrogram of N timesteps would be split into $N-128$ segments of length 128. Each segment is then fed to the appropriate CNN, and the output is interpreted as the probability of each segment to have originated from a positive subject. These probabilities are then used to determine whether or not to classify the subject as positive by taking their mean across all segments, and across all three CNNs and audio files.

6.2.2 Results

While training, we use the validation set differently than during inference, since each CNN is trained independantly. Specifically we split validation spectrograms into non-overlapping segments, feed them to the CNN, and make a final prediction by taking the mean of the predictions on individual segments. We chose to use non-overlapping segments during validation, as opposed to during inference, to save time. We then use early stopping based on validation ROC-AUC score to keep the best performing weights on the validation set. An example of results while validating is shown in the first three columns of Table 6.1.

As mentioned in the previous subsection, during inference we feed overlapping segments of spectrograms to the neural networks, and take the mean prediction across all segments, across all three neural networks and types of inputs (cough, breathing, speech). The resulting ROC-AUC score for networks trained on each fold is shown in the final column of Table 6.1. Based on these results, we used the networks trained on Fold 4 for our submission to the datathon.

Table 6.1: Area under the receiver operating characteristic curve for the best epoch on the validation set while training - for each neural network, and for the final ensemble of models during inference. While validating the segments are non-overlapping, while inferring the segments have maximum overlap.

	Validation			Inference
Data	**Breathing**	**Cough**	**Speech**	**All**
fold 0	0.77	0.76	0.77	0.79
fold 1	0.80	0.75	0.85	0.84
fold 2	0.76	0.73	0.77	0.82
fold 3	0.81	0.75	0.81	0.83
fold 4	0.80	0.83	0.85	0.88

6.2.3 Interpretability

Interpretability is achieved in two stages. In the first stage we present spectrograms corresponding to a subject, overlayed with the prediction probability for each segment, as shown in Figure 6.3. A medical expert might then choose to listen to specific segments which led to a high prediction probability (for instance a single cough). In the second stage, one can select a specific segment of a spectrogram and through some *post hoc* explainability technique get more information about why the specific segment was classified as positive. In Figure 6.4 we show an explanation generated by LIME [165] on the segment corresponding to the highest probability cough of a subject.

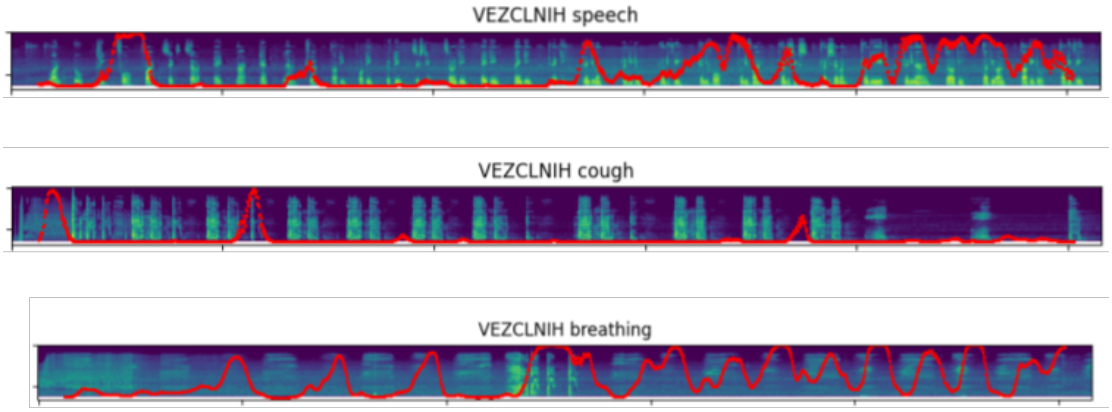


Figure 6.3: Predicted probabilities on each segment overlayed on the mel spectrogram (for a positive subject from the validation set)

Explanations generated in the first stage (Figure 6.3) are intuitively useful, since they show which parts of the audio file contribute towards a prediction. Explanations generated in the second stage are more difficult to evaluate without the opinion of a medical expert. For instance the explanation for the cough shown in Figure 6.4 would indicate that low-frequency sound exactly before the cough is important for the prediction, in addition to mid and high frequencies during the cough and mid frequencies after the cough.

6.3 Developing an Explanation Dataset

For applying our proposed explainability and evaluation framework on this task, the requirement is the existence of an explanation dataset. To this end, we developed our own explanation dataset,

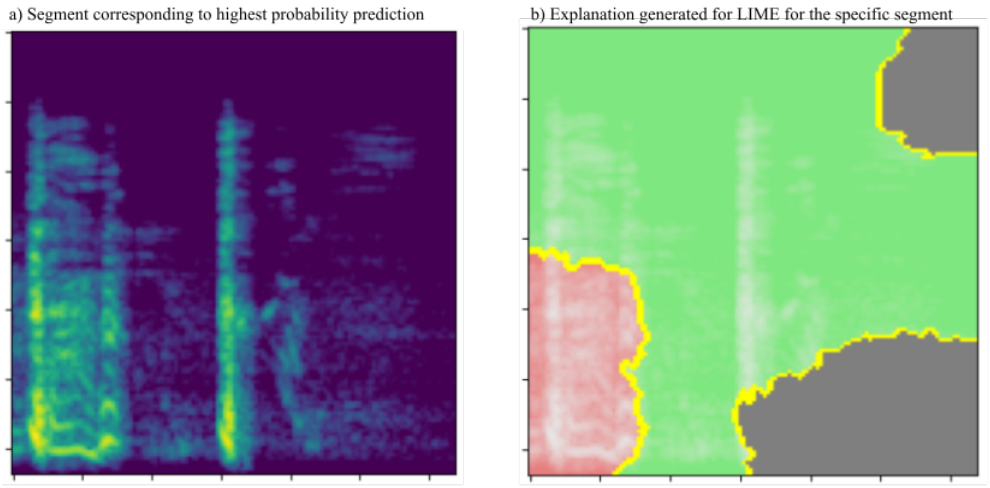


Figure 6.4: (left) Segment corresponding to the highest probability prediction from the cough audio file of subject VEZCLNIH. (right) Explanation generated by LIME for the specific segment. Green areas contribute towards a positive prediction and red area towards a negative prediction.

by utilizing crowd-sourcing and expert annotations.

6.3.1 Data Collection and Curation

The data was collected in the context of the smarty4covid project [219]. Specifically, we created a user-friendly web application (`www.smarty4covid.org`) for Greek and Cypriot citizens aged 18 and older. The questionnaire in smarty4covid had different sections and instructions for users. It asked users to record their voice, breath, and cough sounds and provide information about themselves, like their age, COVID-19 vaccination status, medical history, vital signs (measured using devices), COVID-19 symptoms, smoking habits, hospitalization, emotions, and working conditions. There were four types of audio recordings: reading a sentence, taking deep breaths, regular breathing close to the microphone, and voluntary coughs. To protect user data and follow ethical guidelines, they had user terms and a privacy policy that explained how data would be used, user rights, and data protection measures. Users had to agree to this before taking the questionnaire. The smarty4covid project started in January 2022, during the omicron wave in Greece when COVID-19 cases were high. More than 10,000 people provided information about themselves, but only about half (4,679) gave permission for audio recordings. Among users, 17.3% tested positive for COVID-19. A comprehensive description of all data that was collected, and is available in the public zenodo repository is outlined in table 6.2.

Field name	Description	Type	Values
participantid	Participant's identification number.	String	UUID
submissionid	Questionnaire's Identification number.	String	UUID
covid_status	Tested for COVID-19.	String	”positive”: Positive, ”negative”: Negative, ”no”: Not tested
pcr_test	Tested with PCR.	Bool	

rapid_test	Tested with a Rapid Antigen test.	Bool	
self_test	Tested with a Rapid Antigen Self test.	Bool	
test_last_3_days	Tested in the last 3 days.	Bool	
last_negative_test_date	Date of the last negative test.	String	”yyyy-mm-dd”
first_positive_test_date	Date of the first positive test.	String	”yyyy-mm-dd”
vaccination_status	COVID-19 vaccination status.	String	”no”: No, ”partially”: One of two shots, ”fully”: Fully, ”booster1”: Fully and Booster dose, ”booster2”: Fully and two Booster doses
latest_vaccination_date	Date of the last vaccination dose.	String	”yyyy-mm-dd”
hospitalization	Whether the user was hospitalised for COVID-19.	String	”0”: ”No”, ”1”: ”I am currently hospitalized”, ”2”: ”Yes, discharged a week ago”, ”3”: ”Yes, discharged more than a month ago”
exposure_to_someone_with_covid	Whether the user was exposed to a confirmed COVID-19 case.	String	”No” / ”Maybe” / ”Yes”
travelled_abroad	Whether the user has travelled abroad the last 14 days.	String	”0”: No, ”1”: Yes
submission_timestamp	Timestamp when the submission was received	String	

Table 6.2: Main questionnaire json file description (Part 1/3: COVID-19 related information)

Field name	Description	Type	Values

sore_throat	Symptoms	Sore Throat	Bool	
dry_cough		Dry Cough	Bool	
wet_cough		Productive Cough	Bool	
sputum		Sputum	Bool	
runny_nose		Nasal congestion	Bool	
breath_discomfort		Dyspnea	Bool	
has_fever		Fever	Bool	
tremble		Chills	Bool	
fatigue		Fatigue	Bool	
headache		Headache	Bool	
dizziness		Dizziness/ confusion	Bool	
myalgias_arthralgias		Myalgias, arthralgias	Bool	
taste_smell_loss		Loss of taste/smell	Bool	
diarrhea_upset_stomach		Stomach upset/ Diarrhea	Bool	
sneezing		Sneezing	Bool	
dry_throat		Dry Throat	Bool	
oxymeter	Vital Signs	Oximetry test	Bool	
oxygenSaturation		Oxygen Saturation	Int	[60, 99]
bpm		Beats per minute (BPM)	Int	[30, 250]
blood_pressure_meter		Blood pressure test	Bool	
systolic_pressure		Systolic Pressure	Int	[30, 260]
diastolic_pressure		Diastolic Pressure	Int	[30, 260]
breath_holding		Seconds of breath holding	Int	$[0, \infty)$
leave_bed	Difficulty to	Leave Bed	Bool	
leave_home		Leave Home	Bool	
prepare_meal		Prepare Meal	Bool	
concentrate		Concentrate	Bool	
self_care		Self Care	Bool	
other_difficulty		Every day activites	Bool	

Table 6.3: Main questionnaire json file description (Part 2/3: Symptoms and vital signs)

Field name	Description	Description	Type	Values
smoking	Smoking habits	Smoking status	String	”nev”: Never smoked, ”ex”: Ex-smoker, ”yes”: Smoker
years_of_quitting_smoking	Smoking habits	Years of quitting smoking	Int	$[0, \infty)$
years_of_smoking	Smoking habits	Years of smoking	Int	$[0, \infty)$
no_cigarettes	Smoking habits	Number of cigarettes per day	String	”1u”: less than 1, ”10u”: 1-10, ”20u”: 11-20, ”20o”: more than 20
vaping	Smoking habits	Vaping	String	”0”: No, ”1”: Yes
anxiety	Level of anxiety about the pandemic		String	”0”: None, ”1”: Low, ”2”: Moderate, ”3”: High, ”4”: Very High
working	Working Status		String	”home”: Working from home, ”hospital”: Working in hospital, ”store”: Working in an essential goods store (pharmacy, supermarket), ”social”: Working in a service with increased contact with the general public, ”no”: Not working

Table 6.4: Main questionnaire json file description (Part 3/3: Smoking habits, anxiety level, and working status)

The collected data were also further cleaned and curated by utilizing the data labeling platform Label Studio. Specifically, four campaigns were launched for cleaning the data, and four for additional labeling by medical experts. For the data cleaning process, there were three campaigns pertaining to audio quality and validity of the three different audio types. Labelers were asked if an audio file was valid (e.g. if it were submitted as a cough, is it really a cough?), and were also asked to rate the clarity of the audio file. The goal of the fourth data cleaning campaign was to exclude from the public repository any breathing, or cough audio files, that also contained potentially personal identifiable information (e.g. the subjects voice). For expert annotations, four campaigns were launched, one for each type of audio file, where annotators described the audio using medical terminology, such as audible symptoms, while the fourth showed them all audio files, and self-reported information, and the medical experts were asked to assess the possibility of COVID-19 infection.

6.3.2 Constructing the knowledge base

A web-ontology language (OWL) knowledge base was developed motivated by the need of data consolidation from different relevant databases (i.e. Coughvid, COVID-19 sounds, Coswara) and

the application of complex queries for the detection of users with specific characteristics. All available information resulting from the crowd-sourcing, data cleaning and data labeling procedures were also released in the form of the smarty4covid OWL knowledge base. The smarty4covid OWL knowledge base is hosted on the same Zenodo Repository as the data records [218]. In general, using a vocabulary $\mathcal{V} = \langle \mathsf{CN}, \mathsf{RN}, \mathsf{IN} \rangle$ where CN, RN, IN are mutually disjoint sets of concept names, role names and individual names respectively, a knowledge base ($\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$) can be built through creating the Assertional Database (ABox - $\mathcal{A}$) and the Terminology Database (TBox - $\mathcal{T}$). The ABox includes assertions of the form $C(a), r(a, b)$ where $C \in \mathsf{CN}$, $r \in \mathsf{RN}$, and $a, b \in \mathsf{IN}$. The TBox is a set of terminological axioms of the form $C \sqsubseteq D$, where $C, D \in \mathsf{CN}$, $r \sqsubseteq s$ and $r, s \in \mathsf{RN}$. Based on these axioms, the hierarchies of concepts and roles can be defined in the TBox.

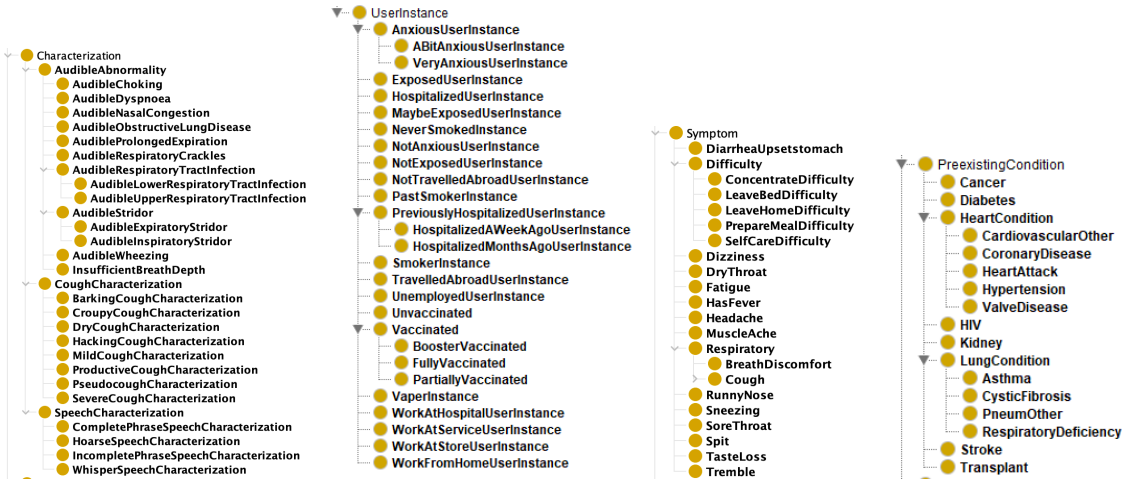
In the smarty4covid OWL knowledge base, the set of individual names (IN) contains a unique name indicative to each participant, questionnaire, audio file, healthcare professional that participated in the labeling procedure and the corresponding characterizations of the audio records. (IN) also includes unique names for each declared symptom, COVID-19 test and preexisting condition that is linked to the corresponding questionnaire (e.g symptom, COVID-19 test) and participant (i.e. underlying condition), respectively. These individuals are linked through appropriately defined roles. The role names RN and their defined hierarchy is depicted in Figure 6.5h. Each role is associated with a domain and a range indicative to the types of the individuals that can be linked through this role. In particular, the role hasCharacterization links audio files to characterizations as labelled by the healthcare professionals, and characterizedBy links characterizations to instances of the healthcare professionals. The role hasAudio and its children link questionnaires to audio files. The roles hasCovidTest and hasSymptom link questionnaires to instances of COVID-19 tests, self-reported symptoms, and vaccination status, respectively. The role hasPreexistingCondition links participants to preexisting conditions, while hasUserInstance links participants to their submitted questionnaires.

The set of concept names CN involves concepts that describe instances of audio, COVID-19 tests, preexisting conditions, symptoms, users and questionnaires. For audio related concepts, their hierarchy is shown is Figure 6.5f. Specifically, there is a concept for each type of audio recording (i.e. regular breathing, deep breathing, voice, cough), and concepts regarding the audio quality. Audio instances can additionally be linked, via the hasCharacterization role to audible abnormalities, for which the hierarchy of concepts is shown in Figure 6.5a. Similarly, all preexisting conditions that appear in the questionnaire are organized as concepts in a hierarchy as shown in Figure 6.5d, and all symptoms are part of the symptom hierarchy, shown in Figure 6.5c. Furthermore, the User concept subsumes concepts related to the different age and gender of the participants, as shown in Figure 6.5e, while the UserInstance concept that corresponds to a specific questionnaire submitted by a user, also subsumes a hierarchy based on the different possible answers in the questionnaire, shown in Figure 6.5b. Finally, the concepts related to COVID-19 tests, shown in Figure 6.5g, are used to define the type of test and its outcome.

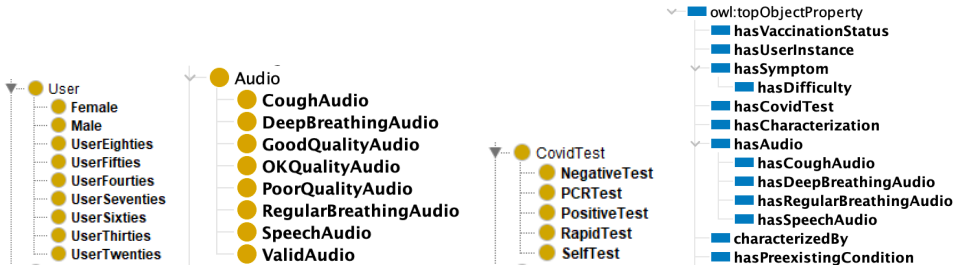
The described hierarchies of concepts and roles are provided in OWL format in the file [smarty-ontology.owl]. Using this terminology, all information presented in the dataset [218] is asserted in the form of triples, provided in the file [smarty-triples.nt]. An example of a smarty4covid user is depicted in Figure 6.6. This user who is a female (20-30 years old) and has asthma, has submitted a questionnaire declaring a positive PCR test and a headache while being a smoker. Her audio recording of cough has been labeled by medical professionals as featuring audible choking.

6.4 Interpretable COVID-19 Classifiers from Tabular data

A school of thought when it comes to explainable AI is that we should not rely on post hoc methods, but instead strive to develop inherently interpretable models [172]. In this section, we utilize the crowd-sourced dataset to develop inherently interpretable COVID-19 classifiers, using tabular data.



(a) Concepts related to expert characterizations. **(b)** Questionnaire related concepts. **(c)** Symptom related concepts. **(d)** Preexsting conditions related concepts.



(e) Concepts for the different types of users. **(f)** Audio related concepts. **(g)** Concepts related to covid tests **(h)** Hierarchy of roles.

Figure 6.5: Hierarchies of concepts and roles from the smarty4covid knowledge base.

6.4.1 Feature Selection and Engineering

For training and evaluating the classifiers, we use a combination of self-reported features from the submitted questionnaires, and audio features extracted from the submitted audio files using signal processing.

Self-reported features

From all information gathered for the smarty4covid dataset, we first excluded those features that would not be useful for classification, or that had mostly missing values, such as "ParticipantID", "oxygen saturation", and the received date of the questionnaire. Boolean information was represented as binary features, while categorical information (e.g. age, sex, vaccination statues) was one-hot encoded. Finally, numerical features such as BMI, and how long one can hold their breath were initially normalized by dividing all values with the maximum for each feature, leading to values in the range $[0, 1]$. After this procedure, we end up with a table of 4303 rows and 70 columns, 69 of which are the aforementioned features, and the 70'th is used as a label (COVID-19 positive or negative).

Signal-processing features

Besides the self-reported features pertaining to pre-existing conditions, demographics, and symptoms, we also extracted features using signal processing. Specifically, we use the same features and

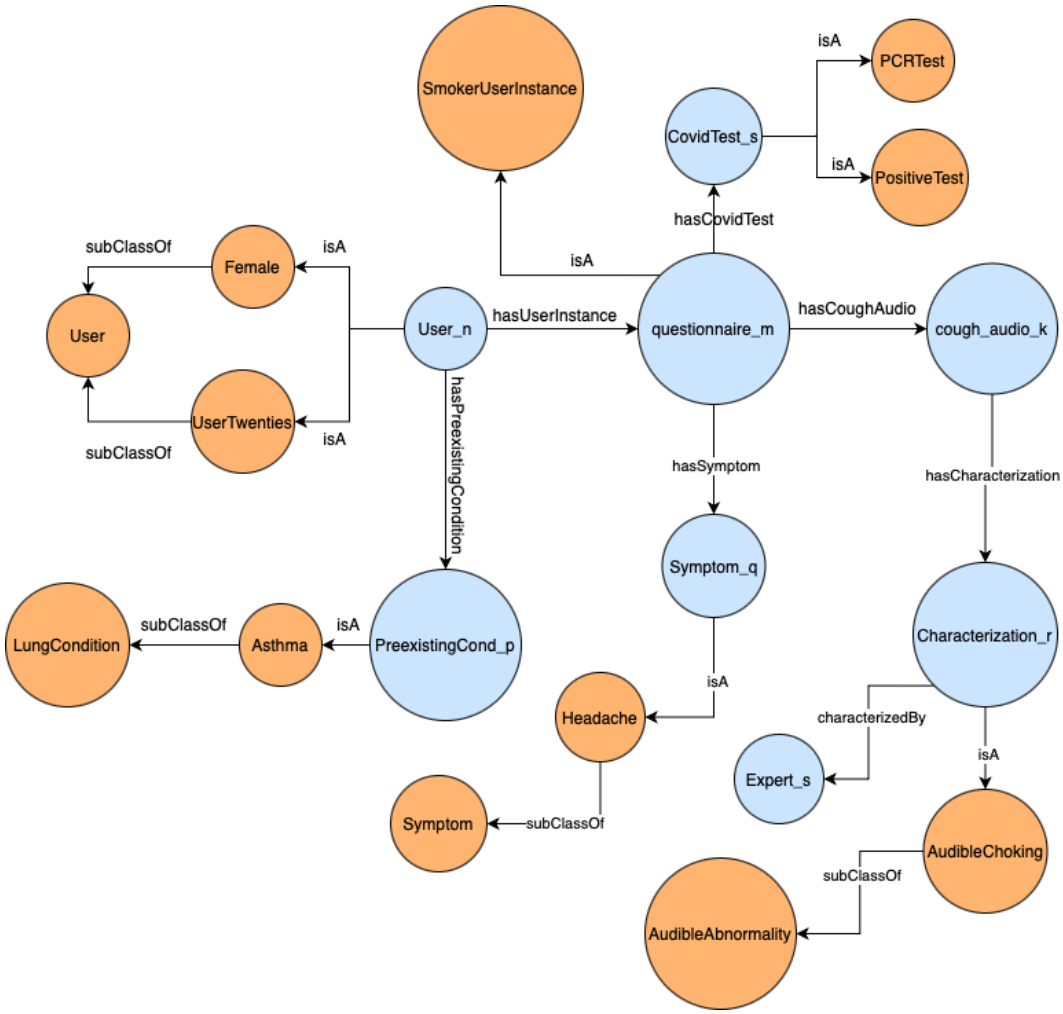


Figure 6.6: Example of the structure of the smarty4covid knowledge base. Blue nodes represent individuals, and orange nodes concepts. Edges labeled as IsA represent concept assertions from the ABox, and subClassOf edges represent inclusion axioms from the TBox.

code from Coughvid. It is worth noting that since these features are meant to be used in conjunction with inherently interpretable machine learning classifiers, they themselves need to be understandable to end-users. Here we provide a short description of the audio features that were extracted using signal processing.

Envelope Energy Peak Detection Returns the number of peaks detected, after applying a band pass filter. We do this from 50Hz to 1kHz with a step of 50Hz

Zero Crossing Rate How many times does the signal cross zero (normalized by number of samples)

Root mean square power Average power of the signal

Dominant Frequency Out of 128 frequencies, returns the one with the most power.

Spectral Centroid Weighted mean of frequencies wrt FFT value at each frequency.

Spectral Spread Weighted standard deviation of frequencies wrt FFT value.

Spectral Rolloff The frequency below which a certain percentage (typically a threshold, such as 85% or 95%) of the total spectral energy of the signal is contained.

Spectral Skewness Distribution of the spectrum around its mean.

Spectral Kurtosis Flatness of the spectrum around its mean.

Spectral Bandwidth Weighted spectral standard deviation.

Spectral Flatness The ratio of the geometric mean to the arithmetic mean of the spectrum.

Spectral Standard Deviation Standard deviation of power spectral density.

Spectral Slope Slope of line of best fit on the spectrum

Mel Frequency Cepstral Coefficients (MFCCs) A small set of features that describe the overall shape of the spectral envelope

Crest Factor Peak value divided by RMS value

Length Length of signal in seconds

Preprocessing

Many of these features return scalars, while others are vectors. When combining them with the self-reported features, we end up with a feature set consisting of 344 values for each sample. This dataset was then split into a development set (75%) and a test set (25%), and the development set was further split into training (75%) and validation (25%) when a validation set was necessary, e.g. for hyperparameter tuning. All data was scaled using a standard scaler by subtracting the mean value (computed on the training set), and dividing with the standard deviation.

6.4.2 Interpretable Machine Learning Classifiers

Our goal when experimenting with interpretable classifiers was not to optimize evaluation metrics, rather, given adequate performance we attempt to interpret the models, to then qualitatively compare this approach with the aforementioned feature-based post hoc explanations from section 6.2.3, and the explanations generated within the proposed framework in section 6.5.

Naive Bayes

Bernoulli Naive Bayes is a simple yet effective probabilistic classification algorithm used primarily for binary classification tasks. It is particularly well-suited for classification applications, where the features are binary, indicating the presence or absence of specific features in data samples. The algorithm assumes that the features are conditionally independent given the class label, making it "naive." Bernoulli Naive Bayes leverages Bayes' theorem to calculate the probabilities of a data point belonging to one of two classes based on the binary features. It estimates the likelihood of each feature being present in each class and combines this information with prior class probabilities to make predictions. As this method is meant for binary features, prior to training features were normalized to $\{0, 1\}$. Notably, the predictions of this classifier can be interpreted by looking at the involved probabilities probabilities $P(C|x)$, $P(x|C)$, for class C and feature x which is useful for gaining insight both for the classifier and for the data it was trained on. Even the probability

$P(C)$ could be considered useful towards interpretability, as it provides information making the behaviour of the classifier more transparent and understandable. Note, we also experimented with the Gaussian Naive Bayes which did not perform as well, so the results are ommited.

The Naive Bayes classifier achieved a 0.71 macro F1 score, with a 0.81 f1 score for negatives and only 0.55 for positives. To interpret this classifier we take a look at the probabilites that were computed. For the negative class, the highest conditional probabilites $P(\text{negative}|x)$ were:

$$P(\text{negative}|\text{AgeCategory\_3}) = 0.9475$$

$$P(\text{negative}|\text{CardiovascularOther}) = 0.9287$$

$$P(\text{negative}|\text{Hypertension}) = 0.9050$$

$$P(\text{negative}|\text{ValveDisease}) = 0.9002$$

$$P(\text{negative}|\text{CysticFibrosis}) = 0.8849$$

These probabilities seem to be very high, and the features do not really seem related to someone being negative to COVID-19. As the negative class is also the most prevalent one, this could be attributed to the rarity of the specific features. Concerning signal processing features, the highest probabilities appear to be:

$$P(\text{negative}|\text{DeepBreathSpectralKurtosis}) = 0.8337$$

$$P(\text{negative}|\text{SpeechPowerSpectralDensity\_950-1150}) = 0.8849$$

$$P(\text{negative}|\text{SpeechPowerSpectralDensity\_500-650}) = 0.8849$$

Regarding spectral kurtosis, a high value indicates that the spectral components have heavier tails, and there may be impulsive or transient events in the signal's frequency domain. It is not really clear why this feature leads to a high probability of negative prediction. We could hypothesize that if a symptom such as wheezing was audible in the breathing recording, then the Kurtosis would be low, as opposed to a normal breath, but it is not really based on any concrete evidence. Regarding power spectral density of speech, as the fundamental frequency of speech for males is around 100 Hz and for females is around 200 Hz, these frequency bands of the top probabilities (950-110 Hz, 500-650 Hz) are not really interpretpretable without further investigation, and good knowledge of the signal processing methods, and harmonics of human speech.

For the positive class, the five highest probabilities were:

$$P(\text{positive}|\text{Dizziness}) = 0.7852$$

$$P(\text{positive}|\text{RunnyNose}) = 0.6967$$

$$P(\text{positive}|\text{DiarrheaUpsetstomach}) = 0.6869$$

$$P(\text{positive}|\text{LeaveBed}) = 0.6046$$

$$P(\text{positive}|\text{Fatigue}) = 0.5539$$

All of these probabilities indicate the importance of the existence of symptoms for a positive prediction. However, two (Dizziness, Diarrhea) of the five symptoms that appear are not known to be strongly associated with COVID-19. This combined with the low f1 score for the positive prediction indicate that this classifier is probably not reliable, despite the acceptable performance when viewing only macro F1 as a metric. Concerning signal processing features for a positive prediction, the probabilities were all low enough to not be worth discussing.

Logistic Regression

Logistic regression is one of the simplest methodologies for classification. For binary classification, the model estimates the probability of the positive class as a sigmoid function applied on a linear combination of the features. During training, the coefficients of the linear combination are computed such that the negative log likelihood is minimized, in which case the predicted distribution best fits the real one. Given a trained logistic regression classifiers, it can be interpreted by looking at the coefficients of the linear combination. Specifically, the absolute value of a coefficient indicates the importance of the corresponding feature. Negative coefficients indicate importance of the feature for a negative prediction, while positive ones indicate importance for positive prediction.

Even though this classifier seems to be more suited for the task than Bernoulli Naive Bayes, the results are marginally worse, with a macro F1 score of 0.69, F1 score for negatives 0.88 and for positives 0.51. Compared to the Naive Bayes classifier, the f1 score is higher for the negatives and lower for the positives, indicating the effect of class imbalance. For interpreting the predictions of this classifier, we can take a look at the coefficients with the largest absolute values. For positive coefficients, which are indicate importance for a positive prediction, they highest values were:

$$\text{CoughEnvelopeEnergyPeakDetection\_400\_450} = 1.2476$$

$$\text{DeepBreathEnvelopeEnergyPeakDetection\_450\_500} = 1.1205$$

$$\text{SpeechEnvelopeEnergyPeakDetection\_250\_300} = 0.9778$$

$$\text{SpeechEnvelopeEnergyPeakDetection\_350\_400} = 0.7112$$

$$\text{SpeechSpectralFlatness} = 0.6830$$

Interestingly, four of the five highest coefficients are Envelope Energy Peak Detection features, that indicate the number of peaks in a specific frequency range, while all five features originate from signal processing. Similarly to before, we have difficulties interpreting the results, due to lack of expertise. Specifically, we do not know if the frequency ranges (e.g. 400-450 Hz for coughs) would have any significance for a medical expert.

The top 5 coefficients for self-reported features, which are more interpretable than the signal processing ones, were:

$$\text{Sneezing} = 0.5830$$

$$\text{OtherDifficulty} = 0.5766$$

$$\text{Exposed\_maybe} = 0.4407$$

$$\text{Dizziness} = 0.4325$$

$$\text{RunnyNose} = 0.3571$$

Similarly to the interpretation of the Naive Bayes classifier, the high value for these coefficients makes sense and as the features are understandable, they can be easily interpreted. Specifically, the symptoms Sneezing and RunnyNose should be correlated with COVID-19 infection. Regarding Dizziness, which was also the most important feature for the Naive Bayes classifier, its relevance to COVID-19 should be discussed by a medical professional, but if it should not be corelated, then it indicates a bias of the smarty4covid dataset. Finally, it is interesting that the feature Exposed_maybe appears, but not the Exposed feature, whose coefficient had a value of -0.05, meaning that it was not important either for positive or for negative prediction.

Regarding negative coefficients, the top 15 features were all signal processing features, which we have difficulty interpreting. For completeness, these were the top 5:

$$\text{CoughPowerSpectralDensity\_3800\_3900} = -0.8884$$

$$\text{BreathSpectralFlatness} = -0.9005$$

$$\text{CoughEnvelopeEnergyPeakDetection\_550\_600} = -1.0970$$

$$\text{SpeechEnvelopeEnergyPeakDetection\_550\_600} = -1.1005$$

$$\text{DeepBreathPowerSpectralDensity\_2300\_2400} = -1.1009$$

XGBoost

XGBoost [26], short for Extreme Gradient Boosting, is a powerful and widely used machine learning algorithm renowned for its exceptional performance in various predictive modeling tasks. It achieves this by sequentially building a multitude of decision trees, each compensating for the errors of its predecessor, ultimately producing a robust and highly predictive model.

This classifier achieved the best performance, with 0.71 macro F1 score, 0.51 for positives, 0.90 for negatives. Regarding interpretability, the most straight forward way to understand the decisions of the classifier would be to view the trees themselves. There are also three distinct ways of measuring feature importance:

- **Cover** measures the relative frequency with which a feature is used in constructing decision trees across all boosting rounds. It quantifies how often a particular feature is chosen as a split point in the trees. Feature cover is useful for understanding how frequently a feature is considered by the model during training. High feature cover suggests that the feature is frequently used for splitting, indicating its importance in the model's overall decision-making process.

- **Gain** represents the average improvement in the model's loss function achieved by using a particular feature for splitting. It quantifies the contribution of each feature to reducing prediction errors. Feature gain is particularly valuable in identifying the features that have the most substantial impact on reducing the model's prediction errors. Features with higher gain scores are more influential in improving the model's overall accuracy.

- **Weight** combines the count of times a feature is used (weight) with the average gain of that feature when used as a split. It is essentially the product of feature cover and feature gain. Feature weight provides a balanced perspective by considering both how often a feature is used and the quality of those splits (measured by gain). Features with higher feature weights are those that not only appear frequently but also provide substantial information gain when used.

The **cover** feature importance of the XGBoost classifier is shown in figure 6.7. The results indicate, that features such as weather the user was exposed, or if they had symptoms, were used frequently in the model's decision making process during training. There is a notable lack of signal processing features, with only Spectral Rolloff of speech audio appearing as important.

The **gain** feature importance of the classifier is shown in figure 6.8. Notably, all values of the Exposed feature appear as important, in addition to symptom features that have appeared in multiple previous interpretations of classifiers (HasFever, WetCough, HasSymptoms). These features all make sense as being important for COVID-19 classification. Furthermore, various Envelope Energy Peak Detection features appear, specifically Cough at 100-150 Hz, 400-450 Hz, and speech at 250-300 Hz. Interestingly, two of three of these features also appeared as important in the logistic regression classifier. However, as mentioned, in order to interpret the importance of these features, and what they actually indicate (e.g. are they a bias of the dataset, or are they truly important for prediction of COVID-19) would require domain expertise.

Finally, the **weight** feature importance, that essentially combines gain and cover, is shown in figure 6.9. Of these, the HasFever and RunnyNose seem to be consistently important across feature importance methods and, to an extent, classifiers. Conversely, the BMI feature appears as the most

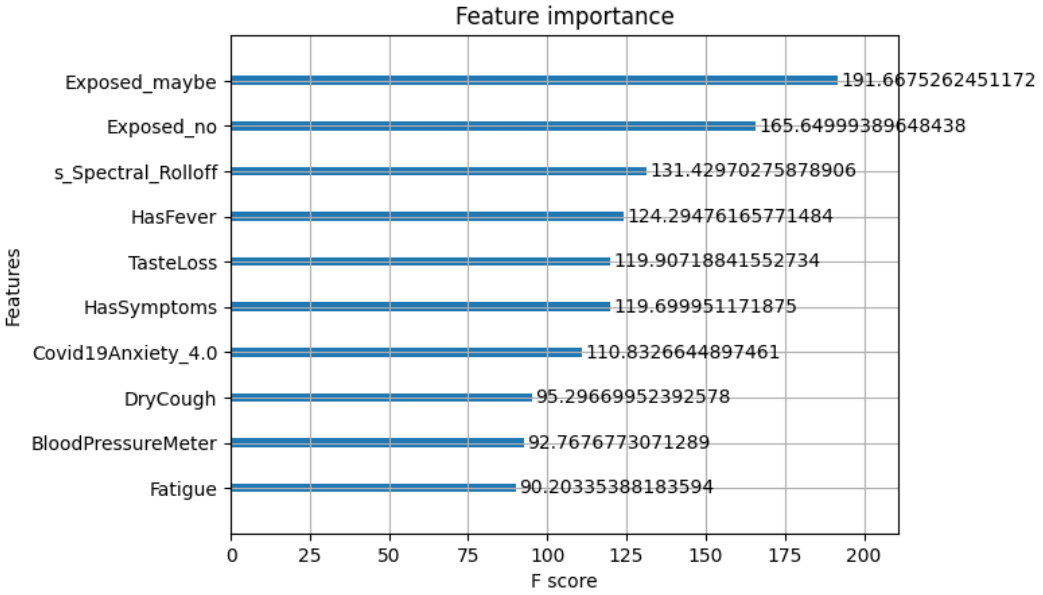


Figure 6.7: Cover feature importance for the XGBoost classifier. The prefixes of the signal processing features $s\_$, $c\_$, $b1\_$, $b2\_$ indicate speech, cough, breathing and deep breathing audio files respectively.

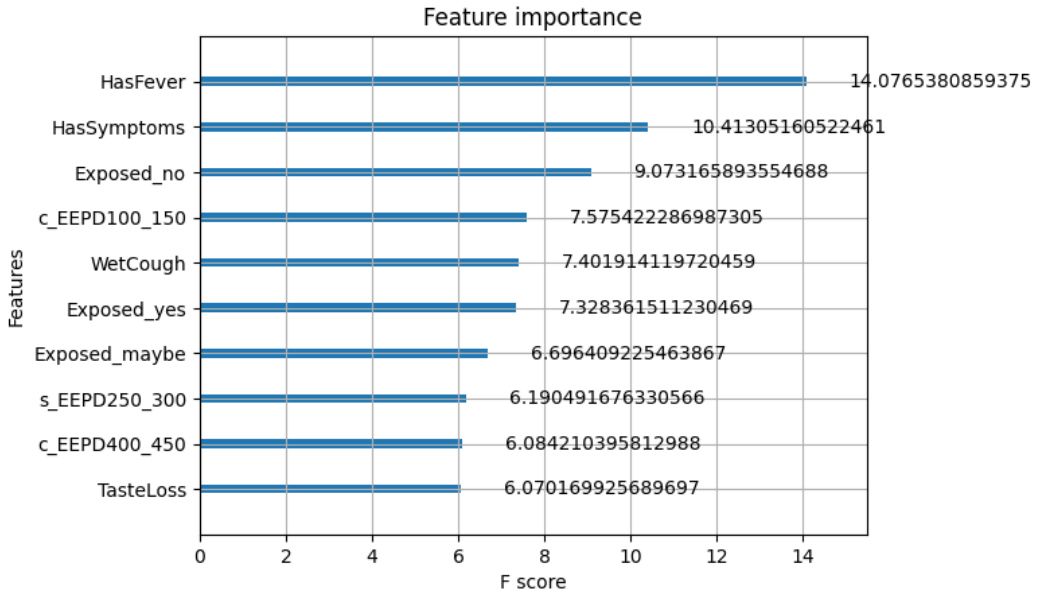


Figure 6.8: Gain feature importance for the XGBoost classifier. The prefixes of the signal processing features $s\_$, $c\_$, $b1\_$, $b2\_$ indicate speech, cough, breathing and deep breathing audio files respectively.

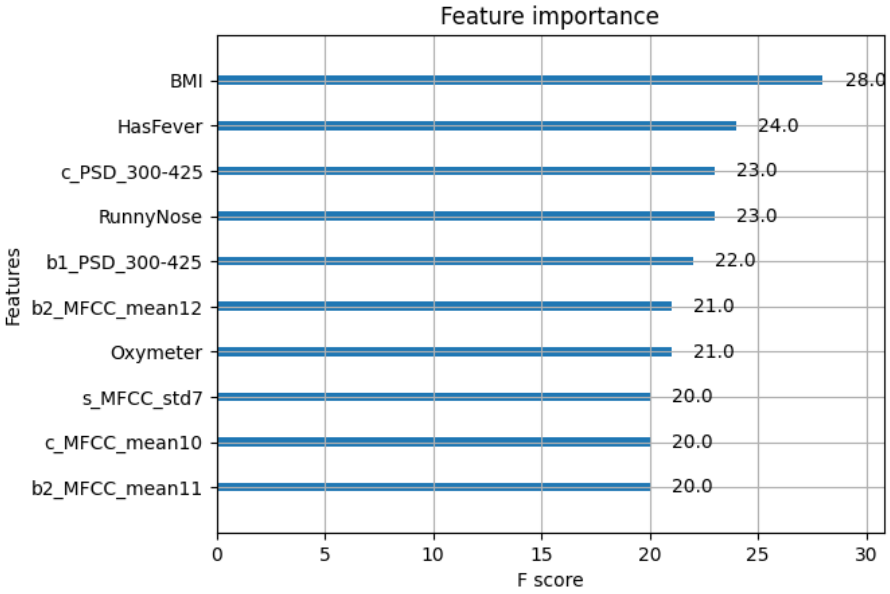


Figure 6.9: Weight feature importance for the XGBoost classifier. The prefixes of the signal processing features $s\_$, $c\_$, $b1\_$, $b2\_$ indicate speech, cough, breathing and deep breathing audio files respectively.

important, and interestingly does not appear in any other feature importance analyses. There are also multiple signal processing features that appear as important, however as mentioned previously, these should better be interpreted by a domain expert.

6.5 Knowledge-based explanations

Using the constructed explanation dataset, provide explanations for the CNN-based classifier from section 6.2, specifically, the winning entry of the IEEE COVID-19 sensor informatics challenge [3]. The input of the classifier is an audio file of a person's coughing, and the output is the probability that the user to has the COVID-19 virus. For this experiment, we further simplified the explanation dataset, keeping only information could potentially be present in the audio file. Thus, we removed concepts such as vaccination status, whether the user has been abroad or if they are anxious about the pandemic.

Results

Since the classifier outputs a continuous value, and we do not know the threshold for classification, we use the source-target significance criterion for counterfactuals, from section 3.5.1. In this case, the importance of a counterfactual for exemplar e is defined $\frac{|P(e)-P(x)|}{\text{edit\_distance}(e,x)}$, where $P(x)$ is the probability of classification assigned by the classifier. By maximizing importance, we are searching for exemplars x who are similar to e (small edit distance), but lead to a large change at the output (large $|P(e) - P(x)|$). A histogram of the predictions of the model on the coughs of smarty4covid in shown in figure 6.10.

An example of a local counterfactual, one with a high importance is from the user with a SubmissionId '435463f8-8e1b-42b3-b467-c7581874700e'. This user is negative to the virus but the classifier has assigned a probability of 0.43 which is on the high end, and with most thresholds would probably be classified as positive. The highest importance counterfactual led to user with submission 'e463ea66-c413-484e-86db-9fb460b1127c' who was also negative to the virus, but the classifier

[3] https://healthcaresummit.ieee.org/data-hackathon/ieee-covid-19-sensor-informatics-challenge/

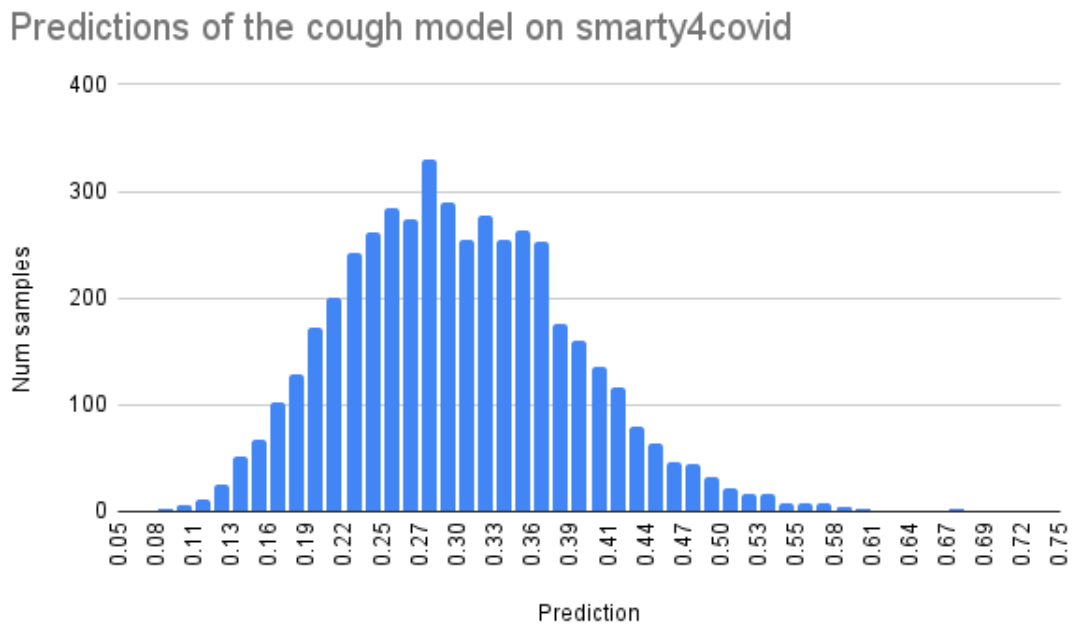


Figure 6.10: Histogram of prediction probabilities of the hackathon cough model, on the coughs of smary4covid.

assigned a probability of 0.14, and would definitely be classified as negative. The counterfactual explanation itself consists of the following edits: Removal of a PneumOther concept, and replacement of UserSixties with UserFourties. This can be interpreted as : If this user did not have a pneumonological preexisting conditions, and were 20 years younger, then the classifier would classify their cough as negative instead of positive. This explanation might indicate confusion of the classifier regarding the pneumonological preexisting condition, as it might have a similar effect on the cough of the subject to the virus itself, and it also indicates a potential bias regarding the age groups. Another example, for submission id '65bb2a6c-4f9c-4594-99b2-1bfbd065172f' who is negative to the virus and is assigned a high probability of 0.38 by the classifier, the most important counterfactual is the exemplar with submission id 'b7a2d599-9a30-43e0-91ef-65ba8e7faa0d' who is actually positive to the virus, but is assigned a probability of 0.20. Thus this counterfactual transforms a false positive into a false negative. The edits that accompany it are: the removals of the Stroke and BreathDiscomfort concepts, and the replacement of UserSixties with UserTwenties, the replacement of PastSmokerInstance with NeverSmokedInstance and the replacement of DiarrheaUpsetStomach symptom with DryThroat symptom. Again, the appearance of the age related concepts might indicate a bias of the classifier, while smoking related concepts might indicate a confounding factor for the classifier (smoker's cough). Finally, the symptom replacement contradicts what we would expect for COVID (more likely that dry throat is a symptom than diarrhea), which could be a result of sparcity of the explanation dataset, or it could indicate a flaw of the classifier.

To further investigate, we can take a look at the global counterfactuals transitioning from "COVID-19 Positive" to "COVID-19 Negative" (Figure 6.11). A first observation is that an important removal is the concept "Symptom", which is the parent of all the symptoms of the knowledge base. However, not every symptom is capable of altering the prediction of the classifier since the concept "Respiratory" which is a child of the concept "Symptom" and the parent of all the symptoms that are related to the respiratory system (e.g., "Dry Cough") is the next most added concept along with its children such as "Sneezing", "Runny Nose", and "Cough". In this experiment, we also uncovered an unwanted bias of the classifier since the most common edit was to change the user's sex from "Female" to "Male". After this peculiar observation, we conducted a search on the training dataset,

and we found out that this bias was inherited from the training set of the classifier. In particular, on the Coswara dataset, 42% of females are COVID-19 positive, while for males the percentage is 27%, which made the classifier erroneously correlate sex to COVID-19 status. Similar observations were made concerning the age groups, specifically the importance of the insertion of the "Age group 30-39" concept and the removal of age groups 40-49 and 50-59. This inherent bias is depicted in Figure 6.12 where we show statistics from the Coswara dataset, on a subset of which the classifier had been trained.

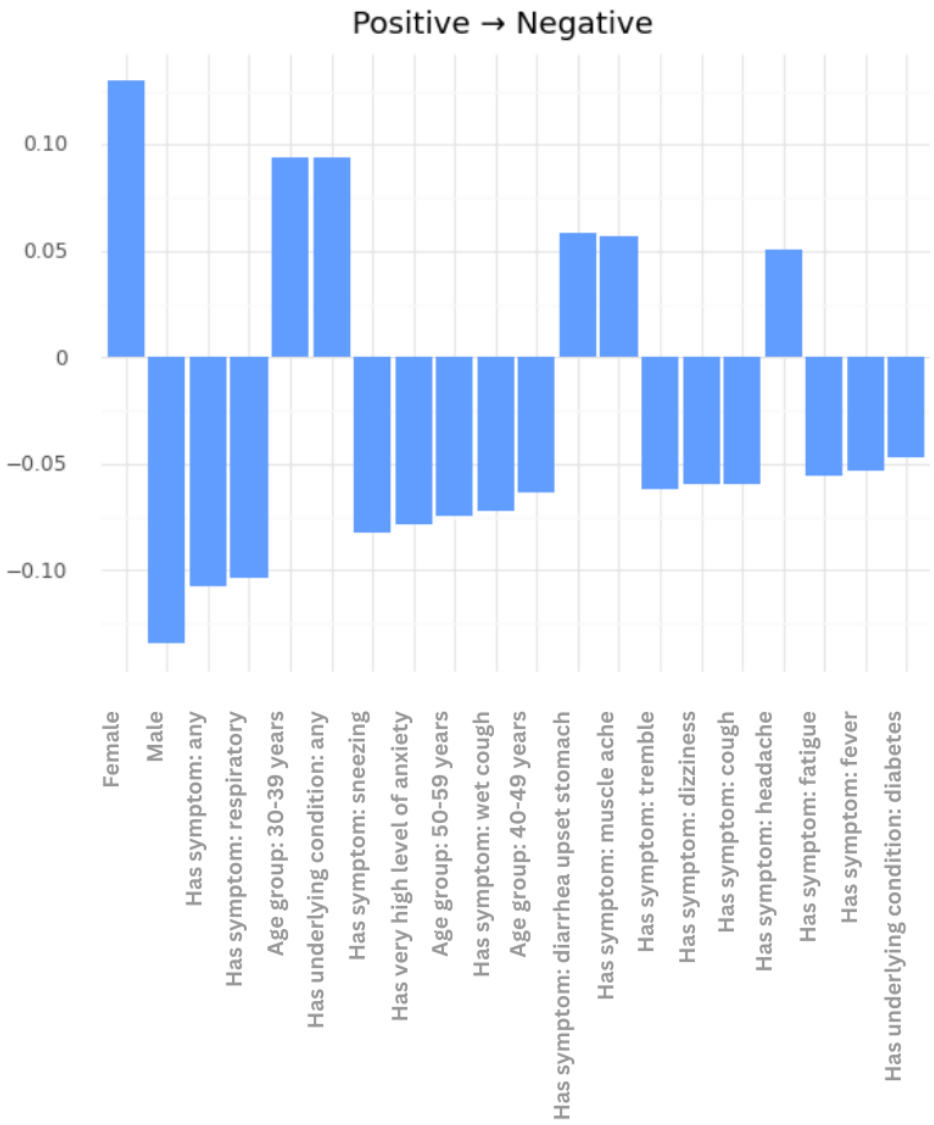


Figure 6.11: Global counterfactual explanations taking into consideration the Coswara dataset [181] as development dataset and the smarty4covid dataset [218] as explanation dataset.

6.6 Conclusion

In this section we have compared three explainability approaches for COVID-19 prediction. Two of them involve opaque models (CNNs), that are explained in a post hoc fashion, first using the features of the classifier (section 6.2.3), and second using the smarty4covid knowledge base within the proposed framework (section 6.5). We also followed a "feature engineering and interpretable classifier approach", which is argued for in [172], as being the way forward for tackling classifier opacity.

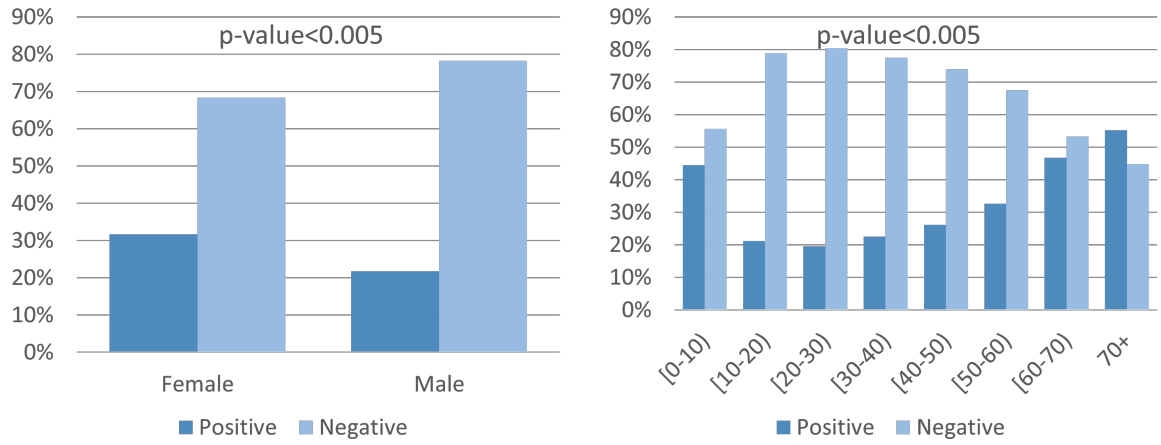


Figure 6.12: Covid-19 prevalence (a) between male and female population and (b) across different age groups in the Coswara dataset [181].

The latter approach we believe highlights exactly the need for knowledge-based semantic explanations that are not necessarily at the same level of abstraction as the features. Specifically, we identify two main issues with this approach. First, in order to explain the predictions of such an interpretable classifier, good knowledge of the classifier itself is required. For example knowing what the coefficients in Logistic Regression mean, or the three different feature importances of XGBoost, how they are effected by class imbalance, scarcity of feature etc. . It is easy to see how this could lead to misleading explanations as an end-user would not necessarily understand what the results mean, but they might believe that they do (Illusion of Explanatory Depth [21]), especially if the explanation indicates causal relationships that the user was expecting. The second issue with the interpretable classifier approach, is its reliance on the understandability of the features. In our experiments this is showcased from the signal processing features, that obviously appear important for the predictions. However, as we lack the domain expertise, we could not really interpret them, and any attempts to do so could be problematic.

Conversely, the post hoc approaches might be able to mitigate some of these issues, but suffer from others. Firstly, as the methods used were model agnostic, the interpretation of the explanations does not require knowledge of the inner workings of classifier under investigation. They would however benefit from knowledge of the explainability algorithm, which we argue would be simpler to explain to a layman end user, rather than the machine learning classifiers, but this would depend on the methods themselves. Being model agnostic also means that we do not have to sacrifice performance for the sake of transparency. Furthermore, the understandability of the terminology used by the explanations is still an issue for the feature-based post hoc approach, as it requires that the semantics of the features are known to the user, however, even in this case, since the output of the explainability algorithm is a visualization, they are more understandable, however they still might be misleading.

Our proposed approach is designed in a way that any potential issues with the resulting explanations are traced back to the explanation dataset, and do not depend on the explainability algorithm, the features of the classifier, or the classifier itself. Upon viewing an explanation, it is immediately understandable, thanks to the terminology that is defined in the knowledge base. An explanation still might be misleading though, if for example the explanation dataset contains unwanted biases. As we have shown however, it is usually easy to check where the biases stem from, especially if we have access to the training set of the classifier, or to an additional explanation dataset.

Chapter 7

Conclusion

In this dissertation, we introduced a theoretical framework that defines explanations based on knowledge graphs, within the formalism of Description Logics. We have presented arguments for this form of explanations, focusing on understandability, thanks to the well-defined terminology and the ability of the framework to be adapted to specific use-cases by appropriately defining an explanation dataset, as we have shown in the multi-domain experiments. We have also argued that this framework provides a solid foundation for enhancing the interpretability of complex AI systems, by way of **grounding** the explanations in a well defined knowledge base. This entails that the quality, understandability, and trustworthiness of the explanations depend solely on the explanation dataset itself, and if for example an explanation were to be misleading, it would be due flaws in the explanation dataset, and not the explainability algorithm. We also explored the utilization of this grounding approach for the evaluation of machine learning systems. Besides the knowledge based heuristics that we developed for evaluating music generation, most of the discussion and analysis of the resulting explanations lead to essentially qualitatively evaluating the model, with respect to very specific information, that which is present in the explanation dataset. For example, uncovering a bias of a model such as the sex bias of the COVID-19 predictor, or the gray color confounding factor of the CLEVR-Hans class 1 images, is arguably concrete evaluation.

The benefits of our proposed approach are accompanied by a unique set of limitations for an explainability method. The main limitation is the reliance of the presented framework and algorithms on the existence of a well-defined explanation dataset. Constructing an explanation dataset from scratch takes significant effort, as we learned during the smarty4covid project in chapter 6, but especially in such use-cases, we argue that the effort can be worth it. Furthermore, we have shown that in some cases it is possible to automatically extract meaningful semantic descriptions that would be useful for explainability, such as the ridge detection and the scene-graph generation approaches from chapter 4, or the extraction of musical intervals from chapter 5. Finally, a limitation for all explainability approaches is the lack of a consistent and reliable evaluation framework.

The lack of a straight-forward, standardized way to evaluate the proposed framework lead us to mainly qualitatively discuss results, and compare with other explanation methods. There are also some quantitative comparisons that seem promising for our framework, such as the human study for CUB in chapter 4. We believe that our qualitative discussion of the experiments may provide unique insights to researchers in XAI, as we have covered and compared numerous existing approaches from different categories, including local, global, counterfactual, rule-based, feature importance, and prototypes. We have shown how these knowledge based explanations can be useful for detecting unwanted biases, and notably, we can guide the algorithms to consider only information that we believe is worth considering for explanations. For example, if we wanted to check a model for gender biases, we would include gender information in the corresponding explanation dataset. This is an important features for many applications.

A domain that is usually mentioned as a motivating example for XAI methods is the domain of medicine. In this case, medical professionals typically look for specific information when given a data sample to analyze (for example given a chest X-ray they might look for specific characteristics, or for a patient's history they might look for specific preexisting conditions given a context). Contrarily, an AI model will have learned to make predictions by ingesting all available information

(for example, the pixels of an X-ray), and we do not have a reliable way to check whether the model looks at the specific characteristics that the medical professional does, which, if it were the case, it would increase the trustworthiness of the model. Our proposed approach, provided we can encode these characteristics in a description logics knowledge base, would be able to check if the model under investigation is checking the characteristics that would be important for the medical expert.

We have also shown how in the context of the proposed framework, external knowledge represented as graphs, and in particular explanation datasets, may be useful for evaluating machine learning systems. The main approach that these models are evaluated with is by measuring performance metrics, such as accuracy or F1 score, on baseline datasets. However, we argue that this approach is somewhat superficial, and even though it provides a solid way to compare models, it does not fully provide an "evaluation" of the model. For example, as we discuss in chapter 5, even though our proposed CNN for MIDI genre recognition surpassed the state-of-the-art with respect to the standardized evaluation metrics, in practice we believe that a musician, or a different end-user, would choose the inherently interpretable, pattern recognition based approach [55]. We showed how external, structured knowledge can be used for evaluation of AI generated music, and discussed how explanation datasets may be used for the explainable evaluation of transformers on the semantic similarity of visual concepts [129]. We believe that evaluation of machine learning systems should include an analysis of explainability, and they should be evaluated across different aspects, and not the single-dimensional performance metrics.

7.1 Future and Ongoing Work

The proposed framework is general enough and the contents of this thesis and our research has been quite horizontal, covering different domains, use-cases, and approaches to explainable machine learning, that there are multiple different branches that may build upon this research.

A first branch that is currently being explored relates to the form of explanations. Within our framework we defined rule-based and counterfactual explanations, and there are arguments supporting both of these as being intuitive and meaningful to humans. Part of our ongoing research involves extending the framework to support counterfactuals of rules and data samples, by combining the two approaches. These might be able to provide both local information about a specific prediction, and global information about the general behaviour of the classifier in a single explanation. Another form of explanations that would be interesting in the context of the proposed framework are prototype explanations. We are exploring the merits of finding these characteristic examples (and criticisms) that may serve as explanations, based on their semantic descriptions. Finally, we are obligated to research the human computer interaction aspect, and draw inspiration from social and cognitive sciences, while experimenting with different forms of explanations.

An important issue with XAI is the issue of trust. On the one hand we need models that we can trust, and explainability serves to increase that trust. However, having misplaced trust can be even more problematic than having none at all. Unfortunately, XAI methods often increase a user's trust by showing inherently incomplete information that the user chooses to interpret in the simplest way, and thus misleadingly increasing trust. For this reason, we believe that XAI methods should attempt to scrutinize models, and prove that they are not trustworthy, instead of trying to find information that supports their trustworthiness. If then a XAI method fails to prove a model's untrustworthiness, then, provided the method's reliability, the model could be considered trustworthy. In our ongoing research, we are developing and formalizing these ideas, and are exploring ways define, and subsequently solve this problem.

An vital aspect for achieving the above goals is the computational one. We defined our framework as rigorously as possible, utilizing the formalism of description logics, having in mind the potential for increased expressivity and more complex knowledge bases in the future. However, even with the simplest explanation datasets, the problem of computing explanations, either rule-based via reverse query answering, or counterfactuals via edit distance computation, is extremely

expensive. Thus, part of our future efforts involves looking into heuristic, and approximate algorithms for the purpose of utilizing larger explanation datasets and more expressive knowledge to compute explanations. For instance, a straight-forward extension would be the utilization of numerical datatype properties, such as representing age as a number, instead of concepts that represent age groups. An important dilemma when developing explainability algorithms is to what extent can we utilize deep learning methodologies in the explanation computation pipeline. For example, the graph similarity computation could be approximated by use of graph neural networks. It is not clear how much the use of opaque systems can impact the reliability and trustworthiness of a XAI pipeline, and since it has the potential to alleviate the huge computational cost, it is worth looking into.

A crucial roadblock for XAI is the lack of a widely accepted evaluation procedure. For further developing our ideas, and measuring information such as reliability and trustworthiness it is imperative that we develop a robust evaluation framework. This is a very active research area, and new ideas appear constantly, either criticising existing evaluation approaches, or proposing new ones. In our view, every use-case is different and would have different priorities regarding the characteristics of explanations that are considered important. For example, for some use-case accuracy might be more important (for example flip-rate for counterfactuals) than understandability (for example minimality for counterfactuals), thus, ideally, an explainability evaluation framework should measure all aspects, and it should be able to be adapted to a specific use-case, by prioritizing information, based on the requirements of the intended user. As first step in this direction we are exploring ways that these requirements can be formalized, and subsequently utilized both for evaluating a XAI pipeline, and for generating more targeted explanations.

A parallel goal, that is necessary both for impooving and building on the framework, and for evaluating it, is the development of more explanation datasets. Throughout the dissertation we constructed explanation datasets via three main routes: a) Using existing knowledge (visual genome, wordnet), b) Using automatic knowledge extraction techniques (scene graph generation, ridge detection), and c) Constructing a dataset from scratch using crowd-sourcing (smarty4covid). We have also identified a fourth route, which we did not follow, which would involve the manual construction and curation of such a dataset by domain experts. Part of our ongoing work involves utilizing our expertise in the domain of symbolic music for developing an explanation dataset via the fourth route. We are also exploring the use of existing resources for developing explanation datasets for other decision critical domains such as law and finance, besides medicine.

A stepping stone for wide applicability of the proposed framework, and a notable omission from this dissertation is the domain of natural language processing (NLP). The reason for the omission is the difficulty of actually defining meaningful semantic descriptions of text, which itself has semantic content. With the popularization of chat bots such as ChatGPT [1], and their increasingly widespread use in most industries, it becomes critical that we are able to explain and scrutinize such models. Thus, significant part of our ongoing work involves developing explanation datasets, and applying our ideas for NLP models. This entails extending the framework to work for generative models besides classification, in addition to coming up with meaningful semantic descriptions of text that can be used for defining explanation datasets in this domain.

[1] https://chat.openai.com

Bibliography

[1] Zeina Abu-Aisheh et al. "An exact graph edit distance algorithm for solving pattern recognition problems." In: *4th International Conference on Pattern Recognition Applications and Methods 2015*. 2015.

[2] Julius Adebayo et al. "Sanity checks for saliency maps." In: *Advances in neural information processing systems* 31 (2018).

[3] José Gómez Aleixandre, Mohamed Elgendi, and Carlo Menon. "The Use of Audio Signals for Detecting COVID-19: A Systematic Review." In: *Sensors* 22.21 (2022), p. 8114.

[4] Marjan Alirezaie et al. "A symbolic approach for explaining errors in image classification tasks." In: *IJCAI Workshop on Learning and Reasoning. Stockholm, Sweden*. 2018.

[5] Julia Angwin et al. "Machine bias: There's software used across the country to predict future criminals." In: *And it's biased against blacks. ProPublica* 23 (2016), pp. 77–91.

[6] Marcelo Arenas, Gonzalo I. Diaz, and Egor V. Kostylev. "Reverse Engineering SPARQL Queries." In: *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*. Ed. by Jacqueline Bourdeau et al. ACM, 2016, pp. 239–249. DOI: 10.1145/2872427.2882989. URL: https://doi.org/10.1145/2872427.2882989.

[7] A. B. Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." In: *Inf. Fusion* 58 (2020), pp. 82–115.

[8] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." In: *Information Fusion* 58 (2020), pp. 82–115.

[9] Franz Baader et al. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[10] Soo Hyun Bae, Inkyu Choi, and Nam Soo Kim. "Acoustic scene classification using parallel combination of LSTM and CNN." In: *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. 2016, pp. 11–15.

[11] Pablo Barceló and Miguel Romero. "The complexity of reverse engineering problems for conjunctive queries." In: (June 2016).

[12] Sören Becker et al. "Interpreting and explaining deep neural networks for classification of audio signals." In: *arXiv preprint arXiv:1807.03418* (2018).

[13] Bernard Bel and Bernard Vecchione. "Computational musicology." In: *Computers and the Humanities* 27.1 (1993), pp. 1–5.

[14] Emmanouil Benetos et al. "Automatic music transcription: An overview." In: *IEEE Signal Processing Magazine* 36.1 (2018), pp. 20–30.

[15] Thierry Bertin-Mahieux et al. "The million song dataset." In: (2011).

[16] Thomas Birtchnell and Anthony Elliott. "Automating the black art: Creative places for artificial intelligence in audio mastering." In: *Geoforum* 96 (2018), pp. 77–86.

[17] Sebastian Böck and Markus Schedl. "Polyphonic piano note transcription with recurrent neural networks." In: *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2012, pp. 121–124.

[18] Dmitry Bogdanov et al. “The MTG-Jamendo dataset for automatic music tagging.” In: (2019).

[19] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. “Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription.” In: *arXiv preprint arXiv:1206.6392* (2012).

[20] Chloë Brown et al. “Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data.” In: *arXiv preprint arXiv:2006.05919* (2020).

[21] Ruth M. J. Byrne. “Good Explanations in Explainable Artificial Intelligence (XAI): Evidence from Human Explanatory Reasoning.” In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 2023, pp. 6536–6544. DOI: 10.24963/ijcai.2023/733. URL: https://doi.org/10.24963/ijcai.2023/733.

[22] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. “A comprehensive survey of graph embedding: Problems, techniques, and applications.” In: *IEEE Transactions on Knowledge and Data Engineering* 30.9 (2018), pp. 1616–1637.

[23] Lei Cai, Jingyang Gao, and Di Zhao. “A review of the application of deep learning in medical image classification and segmentation.” In: *Annals of translational medicine* 8.11 (2020).

[24] Oscar Celma. “Music recommendation.” In: *Music recommendation and discovery*. Springer, 2010, pp. 43–85.

[25] Gunvant Chaudhari et al. “Virufy: Global applicability of crowdsourced and clinical datasets for AI detection of COVID-19 from cough.” In: *arXiv preprint arXiv:2011.13320* (2020).

[26] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system.” In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[27] Keunwoo Choi, George Fazekas, and Mark Sandler. “Text-based LSTM networks for automatic music composition.” In: *arXiv preprint arXiv:1604.05358* (2016).

[28] Alexandros Chortaras, Michalis Giazitzoglou, and Giorgos Stamou. “Inside the Query Space of DL Knowledge Bases.” In: *Description Logics* 2373 (2019).

[29] Madison Cohen-McFarlane, Rafik Goubran, and Frank Knoefel. “Novel coronavirus cough database: Nococoda.” In: *Ieee Access* 8 (2020), pp. 154087–154094.

[30] Richard Cohn. “Introduction to neo-riemannian theory: a survey and a historical perspective.” In: *Journal of Music Theory* (1998), pp. 167–180.

[31] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. “Reltr: Relation transformer for scene graph generation.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[32] Federico Croce et al. “Ontology-based explanation of classifiers.” In: *EDBT/ICDT Workshops*. 2020.

[33] Claudia d’Amato, Nicola Fanizzi, and Floriana Esposito. “A semantic similarity measure for expressive description logics.” In: *arXiv preprint arXiv:0911.5043* (2009).

[34] Roger B Dannenberg, Belinda Thom, and David Watson. “A machine learning approach to musical style recognition.” In: (1997).

[35] W Bas De Haas et al. “HarmTrace: Improving harmonic similarity estimation using functional harmony analysis.” In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*. 2011.

[36] Eoin Delaney, Derek Greene, and Mark T Keane. “Uncertainty Estimation and Out-of-Distribution Detection for Counterfactual Explanations: Pitfalls and Solutions.” In: *arXiv preprint arXiv:2107.09734* (2021).

[37] Jia Deng et al. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[38] Edmund Dervakos, Giorgos Filandrianos, and Giorgos Stamou. "Heuristics for Evaluation of AI Generated Music." In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 9164–9171.

[39] Edmund Dervakos, Natalia Kotsani, and Giorgos Stamou. "Genre recognition from symbolic music with cnns." In: *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer. 2021, pp. 98–114.

[40] Edmund Dervakos, Natalia Kotsani, and Giorgos Stamou. "Genre Recognition from Symbolic Music with CNNs: Performance and Explainability." In: *SN Computer Science* (2023).

[41] Edmund Dervakos et al. "Choose your Data Wisely: A Framework for Semantic Counterfactuals." In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 2023, pp. 382–390. DOI: 10.24963/ijcai.2023/43. URL: https://doi.org/10.24963/ijcai.2023/43.

[42] Edmund Dervakos et al. "Computing Rule-Based Explanations of Machine Learning Classifiers using Knowledge Graphs." In: *arXiv preprint arXiv:2202.03971* (2022).

[43] Edmund Dervakos et al. "Semantic Enrichment of Pretrained Embedding Output for Unsupervised IR." In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*. 2021.

[44] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018).

[45] Prafulla Dhariwal et al. "Jukebox: A generative model for music." In: *arXiv preprint arXiv:2005.00341* (2020).

[46] Gonzalo Diaz, Marcelo Arenas, and Michael Benedikt. "SPARQLByE: Querying RDF Data by Example." In: *Proc. VLDB Endow.* 9.13 (Sept. 2016), pp. 1533–1536. ISSN: 2150-8097. DOI: 10.14778/3007263.3007302. URL: https://doi.org/10.14778/3007263.3007302.

[47] Chris Donahue, Julian McAuley, and Miller Puckette. "Adversarial audio synthesis." In: *arXiv preprint arXiv:1802.04208* (2018).

[48] Hao-Wen Dong et al. "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[49] Yinpeng Dong et al. "Benchmarking adversarial robustness on image classification." In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 321–331.

[50] Kevin Donnelly et al. "SNOMED-CT: The advanced terminology and coding system for eHealth." In: *Studies in health technology and informatics* 121 (2006), p. 279.

[51] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning." In: *arXiv preprint arXiv:1702.08608* (2017).

[52] Sharan Duggirala and Teng-Sheng Moh. "A Novel Approach to Music Genre Classification using Natural Language Processing and Spark." In: *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE. 2020, pp. 1–8.

[53] Jesse Engel et al. "DDSP: Differentiable digital signal processing." In: *arXiv preprint arXiv:2001.04643* (2020).

[54] John Fauvel, Raymond Flood, and Robin J Wilson. *Music and mathematics: From Pythagoras to fractals*. Oxford University Press on Demand, 2006.

[55] Andres Ferraro and Kjell Lemström. "On large-scale genre classification in symbolically encoded music by automatic identification of repeating patterns." In: *Proceedings of the 5th International Conference on Digital Libraries for Musicology*. 2018, pp. 34–37.

[56] Andrés Ferraro and Kjell Lemström. "On large-scale genre classification in symbolically encoded music by automatic identification of repeating patterns." In: *5th International Conference on Digital Libraries for Musicology*. Paris, 2018. ISBN: 978-1-4503-6522-2. DOI: https://doi.org/10.1145/3273024.3273035.

[57] Leonardo Augusto Ferreira, Frederico Gadelha Guimarães, and Rodrigo Silva. "Applying genetic programming to improve interpretability in machine learning models." In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2020, pp. 1–8.

[58] Giorgos Filandrianos et al. "Brainwaves-driven Effects Automation in Musical Performance." In: ().

[59] Giorgos Filandrianos et al. "Conceptual Edits as Counterfactual Explanations." In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering* (2022).

[60] Hironobu Fujiyoshi, Tsubasa Hirakawa, and Takayoshi Yamashita. "Deep learning-based image recognition for autonomous driving." In: *IATSS research* 43.4 (2019), pp. 244–252.

[61] A Gelman et al. "Bayesian data analysis taylor & francis." In: *Boca Raton, FL, USA.[Google Scholar]* (2014).

[62] Amirata Ghorbani et al. "Towards automatic concept-based explanations." In: *Advances in Neural Information Processing Systems* 32 (2019).

[63] Syrine Ghrabli, Mohamed Elgendi, and Carlo Menon. "Challenges and opportunities of deep learning for cough-based COVID-19 diagnosis: A scoping review." In: *Diagnostics* 12.9 (2022), p. 2142.

[64] B. Glimm et al. "Conjunctive Query Answering for the Description Logic SHIQ." In: *IJCAI*. 2007, pp. 399–404.

[65] Birte Glimm, Yevgeny Kazakov, and Trung-Kien Tran. "Ontology Materialization by Abstraction Refinement in Horn SHOIF." In: *AAAI*. AAAI Press, 2017, pp. 1114–1120.

[66] Yuan Gong, Yu-An Chung, and James Glass. "AST: Audio Spectrogram Transformer." In: *arXiv preprint arXiv:2104.01778* (2021).

[67] Michael Good. "MusicXML for notation and analysis." In: *The virtual score: representation, retrieval, restoration* 12.113-124 (2001), p. 160.

[68] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[69] Bryce Goodman and Seth R. Flaxman. "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"." In: *AI Mag.* 38.3 (2017), pp. 50–57.

[70] Masataka Goto and Yoichi Muraoka. "Musical understanding at the beat level: real-time beat tracking for audio signals." In: *Computational auditory scene analysis*. CRC Press, 2021, pp. 157–176.

[71] Georg Gottlob and Christian G. Fermüller. "Removing Redundancy from a Clause." In: *Artif. Intell.* 61.2 (1993), pp. 263–289.

[72] Yash Goyal et al. "Counterfactual visual explanations." In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2376–2384.

[73] Mara Graziani et al. "Concept attribution: Explaining CNN decisions to physicians." In: *Computers in biology and medicine* 123 (2020), p. 103865.

[74] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models." In: *ACM Comput. Surv.* 51.5 (2019), 93:1–93:42.

[75] Riccardo Guidotti et al. “A survey of methods for explaining black box models.” In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[76] Riccardo Guidotti et al. “Local rule-based explanations of black box decision systems.” In: *arXiv preprint arXiv:1805.10820* (2018).

[77] David Gunning et al. *DARPA’s explainable AI (XAI) program: A retrospective*. 2021.

[78] David Gunning et al. “XAI—Explainable artificial intelligence.” In: *Science robotics* 4.37 (2019), eaay7120.

[79] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Leif Sabellek. “Reverse Engineering Queries in Ontology-Enriched Systems: The Case of Expressive Horn Description Logic Ontologies.” In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 1847–1853. DOI: 10.24963/ijcai.2018/255. URL: https://doi.org/10.24963/ijcai.2018/255.

[80] Jing Han et al. “Sounds of COVID-19: exploring realistic performance of audio-based digital testing.” In: *NPJ digital medicine* 5.1 (2022), p. 16.

[81] Tessa Han, Suraj Srinivas, and Himabindu Lakkaraju. “Which Explanation Should I Choose? A Function Approximation Perspective to Characterizing Post hoc Explanations.” In: *arXiv preprint arXiv:2206.01254* (2022).

[82] Boris Hanin. “Which neural net architectures give rise to exploding and vanishing gradients?” In: *Advances in Neural Information Processing Systems*. 2018, pp. 582–591.

[83] Casper Hansen et al. “Contextual and sequential user embeddings for large-scale music recommendation.” In: *Fourteenth ACM conference on recommender systems*. 2020, pp. 53–62.

[84] Norwood Russell Hanson. *Patterns of discovery: An inquiry into the conceptual foundations of science*. CUP Archive, 1965.

[85] Stevan Harnad. “The symbol grounding problem.” In: *Physica D: Nonlinear Phenomena* 42.1-3 (1990), pp. 335–346.

[86] Christopher Harte, Mark Sandler, and Martin Gasser. “Detecting harmonic change in musical audio.” In: *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. 2006, pp. 21–26.

[87] Verena Haunschmid, Ethan Manilow, and Gerhard Widmer. “audiolime: Listenable explanations using source separation.” In: *arXiv preprint arXiv:2008.00582* (2020).

[88] Dan Hendrycks et al. “Natural adversarial examples.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15262–15271.

[89] Romain Hennequin et al. “Spleeter: a fast and efficient music source separation tool with pre-trained models.” In: *Journal of Open Source Software* 5.50 (2020), p. 2154.

[90] Shawn Hershey et al. “CNN architectures for large-scale audio classification.” In: *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE. 2017, pp. 131–135.

[91] Paul Hindemith. *The Craft of Musical Composition: Book 1: Theoretical Part*. Schott Music, 2020.

[92] JA Hirsch et al. “ICD-10: history and context.” In: *American Journal of Neuroradiology* 37.4 (2016), pp. 596–599.

[93] Pascal Hitzler et al. “OWL 2 web ontology language primer.” In: *W3C recommendation* 27.1 (2009), p. 123.

[94] Aidan Hogan et al. “Knowledge Graphs.” In: *CoRR* abs/2003.02320 (2020).

[95] Cheng-Zhi Anna Huang et al. "Music transformer." In: *arXiv preprint arXiv:1809.04281* (2018).

[96] Ali Imran et al. "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app." In: *Informatics in Medicine Unlocked* 20 (2020), p. 100378.

[97] Fanny Jourdan et al. "COCKATIEL: COntinuous Concept ranKed ATtribution with Interpretable ELements for explaining neural net classifiers on NLP tasks." In: *arXiv preprint arXiv:2305.06754* (2023).

[98] Jean Jung et al. "Logical Separability of Incomplete Data under Ontologies." In: July 2020, pp. 517–528. DOI: 10.24963/kr.2020/52.

[99] Spyridon Kantarelis et al. "Functional harmony ontology: Musical harmony analysis with Description Logics." In: *Journal of Web Semantics* (2022), p. 100754.

[100] Richard M Karp. "An algorithm to solve the m× n assignment problem in expected time O (mn log n)." In: *Networks* 10.2 (1980), pp. 143–152.

[101] Ioannis Karydis, Alexandros Nanopoulos, and Yannis Manolopoulos. "Symbolic musical genre classification based on repeating patterns." In: *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*. 2006, pp. 53–58.

[102] Jeremy Kawahara and Ghassan Hamarneh. "Multi-Resolution-Tract CNN with Hybrid Pretrained and Skin-Lesion Trained Layers." In: *Machine Learning in Medical Imaging*. Ed. by Li Wang et al. Cham: Springer International Publishing, 2016, pp. 164–171. ISBN: 978-3-319-47157-0.

[103] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability." In: *Advances in neural information processing systems* 29 (2016).

[104] Been Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677.

[105] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[106] Roman Kontchakov et al. "The Combined Approach to Ontology-Based Data Access." In: *IJCAI*. IJCAI/AAAI, 2011, pp. 2656–2661.

[107] Alexios Kotsifakos et al. "Genre classification of symbolic music with SMBGT." In: *Proceedings of the 6th international conference on PErvasive technologies related to assistive environments*. 2013, pp. 1–7.

[108] John R Koza and John R Koza. *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press, 1992.

[109] R. Krishna et al. "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations." In: *Int. J. Comput. Vis.* 123.1 (2017), pp. 32–73.

[110] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems* 25 (2012).

[111] Jack Lanchantin et al. "Deep motif: Visualizing genomic sequence classifications." In: *arXiv preprint arXiv:1605.01133* (2016).

[112] Freddy Lecue. "On the role of knowledge graphs in explainable AI." In: *Semantic Web* 11 (Dec. 2019), pp. 1–11. DOI: 10.3233/SW-190374.

[113] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *nature* 521.7553 (2015), pp. 436–444.

[114] Yann LeCun et al. “Gradient-based learning applied to document recognition.” In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[115] Jinhyuk Lee et al. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining.” In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[116] Jongpil Lee et al. “Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification.” In: *Applied Sciences* 8.1 (2018), p. 150.

[117] J. Lehmann, S. Bader, and P. Hitzler. “Extracting reduced logic programs from artificial neural networks.” In: *Appl. Intell.* 32.3 (2010), pp. 249–266. DOI: 10.1007/s10489-008-0142-y. URL: https://doi.org/10.1007/s10489-008-0142-y.

[118] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press, 1983.

[119] Feynman Liang. “Bachbot: Automatic composition in the style of bach chorales.” In: *University of Cambridge* 8 (2016), pp. 19–48.

[120] Hongru Liang et al. “PiRhDy: Learning Pitch-, Rhythm-, and Dynamics-aware Embeddings for Symbolic Music.” In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 574–582.

[121] Jason Liartis et al. “Searching for explanations of black-box classifiers in the space of semantic queries.” In: ().

[122] Jason Liartis et al. “Semantic Queries Explaining Opaque Machine Learning Classifiers.” In: (2021).

[123] Rafael Pires de Lima et al. “Deep convolutional neural networks as a geological image classification tool.” In: *The Sedimentary Record* 17.2 (2019), pp. 4–9.

[124] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context.” In: *ECCV (5)*. Vol. 8693. Lecture Notes in Computer Science. Springer, 2014, pp. 740–755.

[125] Tony Lindeberg. “Scale-Space.” In: *Wiley Encyclopedia of Computer Science and Engineering*. American Cancer Society, 2008, pp. 2495–2504. ISBN: 9780470050118. DOI: https://doi.org/10.1002/9780470050118.ecse609.

[126] Zachary C Lipton. “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.” In: *Queue* 16.3 (2018), pp. 31–57.

[127] Caifeng Liu et al. “Bottom-up broadcast neural network for music genre classification.” In: *Multimedia Tools and Applications* 80.5 (2021), pp. 7313–7331.

[128] Omar Lopez-Rincon, Oleg Starostenko, and Gerardo Ayala-San Martín. “Algoritmic music composition based on artificial intelligence: A survey.” In: *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*. IEEE. 2018, pp. 187–193.

[129] Maria Lymperaiou et al. “Towards explainable evaluation of language models on the semantic similarity of visual concepts.” In: *arXiv preprint arXiv:2209.03723* (2022).

[130] Alan Marsden. “Representing Melodic Patterns as Networks of Elaborations.” In: *Computers and the Humanities* 35 (Feb. 2001), pp. 37–54. DOI: 10.1023/A:1002705506386.

[131] Denis Mayr Lima Martins. “Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities.” In: *Information Systems* 83 (2019), pp. 89–100. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2019.03.002. URL: https://www.sciencedirect.com/science/article/pii/S0306437918300978.

[132] Cory McKay and Ichiro Fujinaga. “Musical genre classification: Is it worth pursuing and how can it be improved?” In: *ISMIR*. 2006, pp. 101–106.

[133] Fady Medhat, David Chesmore, and John Robinson. “Masked Conditional Neural Networks for sound classification.” In: *Applied Soft Computing* 90 (2020), p. 106073.

[134] Saroj K Meher and Ganapati Panda. "Deep learning in astronomy: a tutorial perspective." In: *The European Physical Journal Special Topics* 230 (2021), pp. 2285–2317.

[135] Alessandro B Melchiorre et al. "LEMONS: Listenable Explanations for Music recOmmeNder Systems." In: *European Conference on Information Retrieval*. Springer. 2021, pp. 531–536.

[136] David Meredith, Kjell Lemström, and Geraint A Wiggins. "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music." In: *Journal of New Music Research* 31.4 (2002), pp. 321–345.

[137] George A Miller. "WordNet: a lexical database for English." In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[138] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences." In: *Artificial intelligence* 267 (2019), pp. 1–38.

[139] Y. Ming, H. Qu, and E. Bertini. "RuleMatrix: Visualizing and Understanding Classifiers with Rules." In: *IEEE Trans. Vis. Comput. Graph.* 25.1 (2019), pp. 342–352. DOI: 10.1109/TVCG.2018.2864812. URL: https://doi.org/10.1109/TVCG.2018.2864812.

[140] Saumitra Mishra, Bob L Sturm, and Simon Dixon. "Local Interpretable Model-Agnostic Explanations for Music Content Analysis." In: *ISMIR*. 2017, pp. 537–543.

[141] Brent Mittelstadt, Chris Russell, and Sandra Wachter. "Explaining explanations in AI." In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 279–288.

[142] Robert A Moog. "MIDI: musical instrument digital interface." In: *Journal of the Audio Engineering Society* 34.5 (1986), pp. 394–404.

[143] W. James Murdoch et al. "Interpretable machine learning: definitions, methods, and applications." In: *CoRR* abs/1901.04592 (2019).

[144] Daniel Nordgård. "Assessing music streaming and industry disruptions." In: *Policy implications of virtual work*. Springer, 2017, pp. 139–163.

[145] Shu Lih Oh et al. "Classification of heart sound signals using a novel deep WaveNet model." In: *Computer Methods and Programs in Biomedicine* 196 (2020), p. 105604.

[146] Aaron van den Oord et al. "Wavenet: A generative model for raw audio." In: *arXiv preprint arXiv:1609.03499* (2016).

[147] Sergio Oramas et al. "Multimodal deep learning for music genre classification." In: *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.* (2018).

[148] Lara Orlandic, Tomas Teijeiro, and David Atienza. "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms." In: *Scientific Data* 8.1 (2021), pp. 1–10.

[149] Magdalena Ortiz. "Ontology-Mediated Queries from Examples: a Glimpse at the DL-Lite Case." In: *GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence*. Ed. by Diego Calvanese and Luca Iocchi. Vol. 65. EPiC Series in Computing. EasyChair, 2019, pp. 1–14. DOI: 10.29007/jhtz. URL: https://easychair.org/publications/paper/c3CT.

[150] Jose Padial and Ashish Goel. *Music Mood Classification*. 2018.

[151] Johan Pauwels et al. "20 years of automatic chord recognition from audio." In: (2019).

[152] Marcus Pearce, David Meredith, and Geraint Wiggins. "Motivations and methodologies for automation of the compositional process." In: *Musicae Scientiae* 6.2 (2002), pp. 119–147.

[153] Marcus Pearce, Geraint Wiggins, et al. "Towards a framework for the evaluation of machine compositions." In: *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*. Citeseer. 2001, pp. 22–32.

[154] Yifan Peng, Qingyu Chen, and Zhiyong Lu. "An Empirical Study of Multi-Task Learning on BERT for Biomedical Text Mining." In: *arXiv preprint arXiv:2005.02799* (2020).

[155] Alina Petrova et al. "Query-Based Entity Comparison in Knowledge Graphs Revisited." In: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*. Ed. by Chiara Ghidini et al. Vol. 11778. Lecture Notes in Computer Science. Springer, 2019, pp. 558–575. DOI: 10.1007/978-3-030-30793-6_32. URL: https://doi.org/10.1007/978-3-030-30793-6%5C_32.

[156] Jordi Pons et al. "End-to-end learning for music audio tagging at scale." In: *arXiv preprint arXiv:1711.02520* (2017).

[157] Rafael Poyiadzi et al. "FACE: feasible and actionable counterfactual explanations." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 344–350.

[158] Verena Praher et al. "On the Veracity of Local, Model-agnostic Explanations in Audio Classification: Targeted Investigations with Adversarial Examples." In: *arXiv preprint arXiv:2107.09045* (2021).

[159] Colin Raffel. "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching." PhD thesis. Columbia University, 2016.

[160] Sabbir M Rashid, David De Roure, and Deborah L McGuinness. "A music theory ontology." In: *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music*. 2018, pp. 6–14.

[161] Kaivalya Rawal, Ece Kamar, and Himabindu Lakkaraju. "Can i still trust you?: Understanding the impact of distribution shifts on algorithmic recourses." In: *arXiv e-prints* (2020), arXiv–2012.

[162] *Reddit MIDI dataset*. https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/.

[163] Jia-Min Ren, Ming-Ju Wu, and Jyh-Shing Roger Jang. "Automatic music mood classification based on timbre and modulation features." In: *IEEE Transactions on Affective Computing* 6.3 (2015), pp. 236–246.

[164] Yi Ren et al. "Popmag: Pop music accompaniment generation." In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 1198–1206.

[165] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[166] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[167] David Rizo, José Manuel Iñesta, and Francisco Moreno-seco. "Tree-Structured Representation of Musical Information." In: *1ST Iberian Conference on pattern recognition and image analysis. Palma De Mallorca, Spain, Vol. 2652 OF LNCS*. Lecture, 2003, pp. 838–846.

[168] David Rizo and Alan Marsden. "An MEI-based Standard Encoding for Hierarchical Music Analyses." In: *Int. J. Digit. Libr.* 20.1 (Mar. 2019), pp. 93–105. ISSN: 1432-5012. DOI: 10.1007/s00799-018-0262-x.

[169] Perry Roland. "The music encoding initiative (MEI)." In: *Proceedings of the First International Conference on Musical Applications Using XML*. Vol. 1060. 2002, pp. 55–59.

[170] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[171] Aldona Rosner and Bozena Kostek. "Automatic music genre classification based on musical instrument track separation." In: *Journal of Intelligent Information Systems* 50.2 (2018), pp. 363–384.

[172] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215.

[173] Alberto Sanfeliu and King-Sun Fu. "A distance measure between attributed relational graphs for pattern recognition." In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.3 (1983), pp. 353–362. DOI: 10.1109/TSMC.1983.6313167.

[174] Md. K. Sarker et al. "Explaining Trained Neural Networks with Semantic Web Technologies: First Steps." In: *NeSy*. Vol. 2003. CEUR Workshop Proceedings. CEUR-WS.org, 2017.

[175] Heinrich Schenker. *Free Composition: Volume III of new musical theories and fantasies*. Vol. 3. Pendragon Press, 2001.

[176] Alexander Schindler, Rudolf Mayer, and Andreas Rauber. "Facilitating Comprehensive Benchmarking Experiments on the Million Song Dataset." In: *ISMIR*. 2012, pp. 469–474.

[177] Martin G Schultz et al. "Can deep learning beat numerical weather prediction?" In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200097.

[178] A Carlisle Scott et al. "Explanation capabilities of production-based consultation systems." In: *American Journal of Computational Linguistics* (1977), pp. 1–50.

[179] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[180] Christine Senac et al. "Music feature maps with convolutional neural networks for music genre classification." In: *Proceedings of the 15th international workshop on content-based multimedia indexing*. 2017, pp. 1–5.

[181] Neeraj Sharma et al. "Coswara–a database of breathing, cough, and voice sounds for COVID-19 diagnosis." In: *arXiv preprint arXiv:2005.10548* (2020).

[182] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." In: *arXiv preprint arXiv:1312.6034* (2013).

[183] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

[184] Dylan Slack et al. "Counterfactual Explanations Can Be Manipulated." In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 62–75. URL: https://proceedings.neurips.cc/paper/2021/file/009c434cab57de48a31f6b669e7ba266-Paper.pdf.

[185] Leslie N Smith. "Cyclical learning rates for training neural networks." In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2017, pp. 464–472.

[186] Arun Solanki and Sachin Pandey. "Music instrument recognition using deep convolutional neural networks." In: *International Journal of Information Technology* (2019), pp. 1–10.

[187] Robyn Speer, Joshua Chin, and Catherine Havasi. "Conceptnet 5.5: An open multilingual graph of general knowledge." In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[188] W. Stammer, P. Schramowski, and K. Kersting. "Right for the Right Concept: Revising Neuro-Symbolic Concepts by Interacting with their Explanations." In: *arXiv preprint arXiv:2011.12854* (2020).

[189] Lawrence W Stark et al. “Representation of human vision in the brain: How does human perception recognize images?” In: *Journal of Electronic Imaging* 10.1 (2001), pp. 123–151.

[190] Wensi Tang et al. “Rethinking 1d-cnn for time series classification: A stronger baseline.” In: *arXiv preprint arXiv:2002.10061* (2020).

[191] Ilaria Tiddi and Stefan Schlobach. “Knowledge Graphs as tools for Explainable Machine Learning: a survey.” In: *Artificial Intelligence* (2021), p. 103627.

[192] Matt Turek. “Explainable artificial intelligence (XAI).” In: *Defense Advanced Research Projects Agency. http://web. archive. org/web/20190728055815/https://www. darpa. mil/program/explainable-artificial-intelligence* (2018).

[193] Alan M Turing and J Haugeland. “Computing machinery and intelligence.” In: *The Turing Test: Verbal Behavior as the Hallmark of Intelligence* (1950), pp. 29–56.

[194] Dmitri Tymoczko. “The generalized tonnetz.” In: *Journal of Music Theory* (2012), pp. 1–52.

[195] Esko Ukkonen, Kjell Lemström, and Veli Mäkinen. “Sweepline the music.” In: *Computer Science in Perspective*. Springer, 2003, pp. 330–342.

[196] Simon Vandenhende et al. “Making heads or tails: Towards semantically consistent visual counterfactuals.” In: *European Conference on Computer Vision*. Springer. 2022, pp. 261–279.

[197] Sergey Verbitskiy and Viacheslav Vyshegorodtsev. “ERANNs: Efficient Residual Audio Neural Networks for Audio Pattern Recognition.” In: *arXiv preprint arXiv:2106.01621* (2021).

[198] S Vishnupriya and K Meenakshi. “Automatic music genre classification using convolution neural network.” In: *2018 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE. 2018, pp. 1–4.

[199] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. “Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR.” In: *CoRR* abs/1711.00399 (2017).

[200] Catherine Wah et al. “The caltech-ucsd birds-200-2011 dataset.” In: (2011).

[201] Hongwei Wang et al. “DKN: Deep knowledge-aware network for news recommendation.” In: *Proceedings of the 2018 world wide web conference*. 2018, pp. 1835–1844.

[202] Quan Wang et al. “Knowledge graph embedding: A survey of approaches and applications.” In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.

[203] I-Chieh Wei, Chih-Wei Wu, and Li Su. “Improving automatic drum transcription using large-scale audio-to-midi aligned data.” In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 246–250.

[204] Jing Wei, Marimuthu Karuppiah, and A Prathik. “College music education and teaching based on AI techniques.” In: *Computers and Electrical Engineering* 100 (2022), p. 107851.

[205] Gerhard Widmer. “The synergy of music theory and AI: Learning multi-level expressive interpretation.” In: *Proceedings of the National Conference on Artificial Intelligence*. John Wiley & Sons Ltd. 1994, pp. 114–114.

[206] Weibin Wu et al. “Towards global explanations of convolutional neural networks with concept attribution.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8652–8661.

[207] Yiming Wu and Wei Li. “Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model.” In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.2 (2018), pp. 355–366.

[208] Yusong Wu et al. “MIDI-DDSP: Detailed control of musical performance via hierarchical modeling.” In: *arXiv preprint arXiv:2112.09312* (2021).

[209] Tong Xia et al. “COVID-19 Sounds: A Large-Scale Audio Dataset for Digital Respiratory Screening.” In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[210] Di Xie, Jiang Xiong, and Shiliang Pu. “All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6176–6185.

[211] Saining Xie et al. “Aggregated Residual Transformations for Deep Neural Networks.” In: *CoRR* abs/1611.05431 (2016). arXiv: 1611.05431. URL: http://arxiv.org/abs/1611.05431.

[212] Feiyu Xu et al. “Explainable AI: A brief survey on history, research areas, approaches and challenges.” In: *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*. Springer. 2019, pp. 563–574.

[213] Hao Xue, Like Xue, and Feng Su. “Multimodal music mood classification by fusion of audio and lyrics.” In: *International Conference on Multimedia Modeling*. Springer. 2015, pp. 26–37.

[214] H. Yang, C. Rudin, and M. I. Seltzer. “Scalable Bayesian Rule Lists.” In: *ICML*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3921–3930.

[215] Rui Yang et al. “Parallel Recurrent Convolutional Neural Networks-Based Music Genre Classification Method for Mobile Devices.” In: *IEEE Access* 8 (2020), pp. 19629–19637.

[216] Yang Yu et al. “Deep attention based music genre classification.” In: *Neurocomputing* 372 (2020), pp. 84–91.

[217] S. Zagoruyko and N. Komodakis. “Wide Residual Networks.” In: *BMVC*. BMVA Press, 2016.

[218] Konstantia Zarkogianni et al. *Smarty4Covid Dataset*. Zenodo. 2022. DOI: https://doi.org/10.5281/zenodo.8301142.

[219] Konstantia Zarkogianni et al. “The smarty4covid dataset and knowledge base: a framework enabling interpretable analysis of audio signals.” In: *arXiv preprint arXiv:2307.05096* (2023).

[220] Zhiping Zeng et al. “Comparing stars: On approximating graph edit distance.” In: *Proceedings of the VLDB Endowment* 2.1 (2009), pp. 25–36.

[221] Jianfeng Zhao, Xia Mao, and Lijiang Chen. “Speech emotion recognition using deep 1D & 2D CNN LSTM networks.” In: *Biomedical Signal Processing and Control* 47 (2019), pp. 312–323.

[222] Eve Zheng, Melody Moh, and Teng-Sheng Moh. “Music genre classification: A n-gram based musicological approach.” In: *2017 IEEE 7th International Advance Computing Conference (IACC)*. IEEE. 2017, pp. 671–677.

[223] Bolei Zhou et al. “Places: A 10 million image database for scene recognition.” In: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), pp. 1452–1464.

[224] Moshé M. Zloof. “Query-by-example: the invocation and definition of tables and forms.” In: *VLDB ’75*. 1975.